Event Extraction from Biomedical Texts Using Trimmed Dependency Graphs

Dissertation

zur Erlangung des Grades eines

Doktors der Philosophie

der Friedrich-Schiller-Universität Jena
am Institut für Germanistische Sprachwissenschaft
von

Ekaterina Buyko

Jena
2012

Tag der mündlichen Prüfung: 19.07.2012

Dekan: Prof. Dr. Hermann Funk

Gutachter:
Prof. Dr. Udo Hahn, Friedrich-Schiller-Universität Jena
Prof. Dr. Edward John Briscoe, University of Cambridge

Abstract

This thesis explores the automatic extraction of information from bioscientific publications. Such techniques are urgently needed because the biosciences are publishing continually increasing numbers of texts. The focus is on event extraction where an event is a special form of semantic relationship between named entities. Events are currently manually curated from the literature by professionals called biocurators so as to increase research productivity by improving the efficiency and speed with which biologists can discover information. Manual biocuration, however, is time-consuming and costly so automatic methods are needed for information discovery from the literature. My research enables the development of adequately tested, high-performance information extraction solutions.

I examine concepts of *event* from related research in philosophy, and in theoretical and computational linguistics. I apply the outcome to modeling events in information extraction research. In particular, I focus on biomedical event description in the literature and the potential methods for taking account of, and capturing, its intricacies. The results obtained through this investigation can help in modeling and guiding manual event annotation of texts by domain experts. The GeneReg corpus, a result of the annotation campaign performed as a part of this thesis, is presented.

A further considerable part of this thesis is dedicated to modeling, implementing and evaluating an advanced event extraction approach based on the analysis of syntactic dependency graphs. The thesis contains the event extraction approach proposed and its implementation, the JReX (Jena Relation eXtraction) system. This system was used by the Jena University Language & Information Engineering Lab (Julie Lab) team in the "BioNLP 2009 Shared Task on Event Extraction" competition and was ranked second among 24 competing teams. JReX is currently the highest scorer on the worldwide shared U-Compare event extraction server, now outperforming the competing systems from the challenge. This success was made possible, among other things, by my extensive research on event extraction solutions carried out during this thesis, e.g., exploring the effects of syntactic and semantic processing procedures on solving the event extraction task.

The evaluations executed on competition data that is standard and accepted community-wide, were complemented by real-life evaluation of large-scale biomedical database reconstruction. For this evaluation, I selected the highly relevant topic of gene expression regulation. I showed that considerable parts of manually curated databases can be automatically re-created with the help of the event extraction approach developed. Successful re-creation was possible for parts of RegulonDB, the world's largest manually curated reference database for the transcriptional regulation network of *E. coli*, and the *Candida albicans* regulatory network, manually curated from the scientific literature at the Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute in Jena. Furthermore, this thesis demonstrates that reconstructing databases can even complement manual biocuration. In summary, the event extraction approach justified, developed and implemented in this thesis meets the needs of a large human curator community and thus helps in the acquisition of new knowledge in the biosciences.

Acknowledgments

First of all, I thank my supervisor, Prof. Dr. Udo Hahn, for his guidance throughout my PhD. I am grateful for his continual good advice, motivation, immediate readiness for discussion on numerous occasions, and the freedom he generously extended to me during my PhD time. I greatly appreciate the unique way he encouraged me to reach new heights in my research. I am grateful to Prof. Dr. Ted Briscoe for his willingness in examining my thesis and for his good advice during its completion phase.

My thanks go to my colleagues, past and present, at the JULIE Lab in Jena for our exciting discussions on various research topics and for their constant readiness to support my work. My special thanks go to Erik Faessler for his great scientific and technical commitment during our participation in the "BioNLP 2009 Shared Task on Event Extraction". I much appreciated his generously spending with me a number of late evenings and weekends in the lab. My thanks also go to student assistants for their involvement in corpus annotation and software development. I thank especially Tobias Wagner for annotating the GENEREG corpus and his constant help with, and great expertise in, biological issues. My gratitude also goes to my research colleagues, past and present, from other institutions (Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute (HKI) in Jena, European Bioinformatics Institute, Tsujii Lab at the University of Tokyo, Centro de Ciencias Genómicas (CCG) at the Universidad Nacional Autonóma de México). I am grateful to the REGULONDB team from the CCG and the team from the HKI for kindly preparing their document collections for me.

For proof-reading of this thesis, my thanks go to Andrew Davis, Anne Schneider, Erik Faessler, Johannes Hellrich, Konstantin Emich, Michael Poprat, and Michelle Wilbraham.

I thank many institutions for making this research possible – Friedrich-Schiller-Universität Jena, Framework Programmes of the European Commission and the German Ministry of Education and Research.

Personally, I thank my mother for her great support, unique intuition and continual motivation. Thank you for making possible my move to Germany for my studies and thus, finally, enabling my PhD to become a reality.

List of publications included in the thesis

Ekaterina Buyko, Erik Faessler, Joachim Wermter and Udo Hahn. Syntactic Simplification and Semantic Enrichment - Trimming Dependency Graphs for Event Extraction. In *Computational Intelligence*, Vol. 27, Issue 4, pages 610–644, 2011, doi:10.1111/j.1467-8640.2011.00402.x.

Ekaterina Buyko, Jörg Linde, Steffen Priebe, and Udo Hahn. Towards automatic pathway generation from biological full-text publications. In *IDA 2011 – Proceedings of the 10th International Symposium on Intelligent Data Analysis*, Lecture Notes in Computer Science, Springer, Vol. 7014, pages 67–79, 2011.

Ekaterina Buyko and Udo Hahn. Generating Semantics for the Life Sciences via Text Analytics. In *ICSC 2011 – Proceedings of the fifth IEEE International Conference on Semantic Computing*, pages 193–196, Stanford University, Palo Alto, USA, September 2011.

Ekaterina Buyko and Udo Hahn. Evaluating the Impact of Alternative Dependency Graph Encodings on Solving Event Extraction Tasks. In *EMNLP 2010 – Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 982–992, Association for Computational Linguistics, Cambridge MA, USA, October 2010.

Ekaterina Buyko, Elena Beisswanger and Udo Hahn. The GeneReg Corpus for Gene Expression Regulation Events - An Overview of the Corpus and its In-Domain and Out-of-Domain Interoperability. In *LREC 2010 – Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 2662–2666, Malta, May 2010.

Ekaterina Buyko, Erik Faessler, Joachim Wermter and Udo Hahn. Event Extraction from Trimmed Dependency Graphs. In *BioNLP 2009 – Proceedings of Natural Language Processing in Biomedicine NAACL 2009 Workshop, Companion Volume: Shared Task on Event Extraction*, pages 19–27, Association for Computational Linguistics, Boulder, Colorado, USA, June 2009.

Udo Hahn, Katrin Tomanek, Ekaterina Buyko, Jung-jae Kim and Dietrich Rebholz-Schuhmann. How Feasible and Robust is the Automatic Extraction of Gene Regulation Events? A Cross-Method Evaluation under Lab and Real-Life Conditions. In

BioNLP 2009 – Proceedings of Natural Language Processing in Biomedicine NAACL 2009 Workshop, pages 37–45, Association for Computational Linguistics, Boulder, Colorado, USA, June 2009.

Ekaterina Buyko, Elena Beisswanger and Udo Hahn. Testing Different ACE-Style Feature Sets for the Extraction of Gene Regulation Relations from Medline Abstracts. In *SMBM 2008 – Proceedings of the Third International Symposium on Semantic Mining in Biomedicine*, pages 21–28, Turku, Finland, September 2008.

List of publications not included in the thesis

Ekaterina Buyko, Elena Beisswanger and Udo Hahn. Extraction of Pharmacogenetic and Pharmacogenomic Relations - A Case Study Using PharmGKB. In *PSB 2012 – Proceedings of the Pacific Symposium on Biocomputing*, Big Island, Hawaii, USA, January 2012.

Yoshinobu Kano, Jari Bjorne, Filip Ginter, Tapio Salakoski, Ekaterina Buyko, Udo Hahn, K Bretonnel Cohen, Karin Verspoor, Christophe Roeder, Lawrence E Hunter, Halil Kilicoglu, Sabine Bergler, Sofie Van Landeghem, Thomas Van Parys, Yves Van de Peer, Makoto Miwa, Sophia Ananiadou, Mariana Neves, Alberto Pascual-Montano, Arzucan Ozgur, Dragomir R Radev, Sebastian Riedel, Rune Saetre, Hong-Woo Chun, Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta and Jun'ichi Tsujii. U-Compare bio-event meta-service: compatible BioNLP event extraction services. In *BMC Bioinformatics* 2011, 12:481, doi:10.1186/1471-2105-12-481.

Jörg Linde, Ekaterina Buyko, Robert Altwasser, Udo Hahn, Reinhard Guthke. Full-genomic network inference for non-modell organisms: A case study for the fungal pathogen *Candida albicans*. In *Proceedings of the WASET International Conference on Bioinformatics, Computational Biology and Biomedical Engineering*, pages 224–228, Paris, August 2011.

Jörg Linde, Robert Altwasser, Ekaterina Buyko, Udo Hahn, Reinhard Guthke. Genome-Scale Network Inference for the Pathogen *Candida albicans*. In *ICSB 2011 – Proceedings of the 12th International Conference on Systems Biology*, Heidelberg, August-September 2011.

Dietrich Rebholz-Schuhmann, Antonio José Jimeno-Yepes, Erik M. van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Katrin Tomanek, Elena Beisswanger and Udo Hahn. The CALBC Silver Standard Corpus for Biomedical Named Entities – A Study in Harmonizing the Contributions from Four Independent Named Entity Taggers. In *LREC 2010 — Proceedings of the 7th International Conference Language Resources and Evaluation*, pages 568–573, Malta, May 2010.

Dietrich Rebholz-Schumann, Antonio José Jimeno Yepes, Erik. M. Van Mullingen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. The CALBC Silver Standard Corpus - Harmonizing Multiple Semantic Annotations in a Large Biomedical Corpus. In *Journal of Bioinformatics and Computational Biology*, Vol. 8, Issue 1, pages 163–179, 2010, doi: 10.1142/S0219720010004562.

Ekaterina Buyko and Udo Hahn. Comparing the Benefits of WordNet's Semantic Similarity with Simple Morpho-Syntactic features for the Resolution of Noun Phrase Coordination Ambiguity. In *COLING 2008 – Proceedings of the 22nd International Conference on Computational Linguistics*, pages 89–96, Coling 2008 Organizing Committee, Manchester, UK, August 2008.

Udo Hahn, Elena Beisswanger, Ekaterina Buyko, Michael Poprat, Katrin Tomanek, and Joachim Wermter. Semantic annotations for biology - a corpus development initiative at Jena University Language & Information Engineering (Julie) Lab. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2257–2261, Marrakech, Morocco, May 2008.

Udo Hahn, Ekaterina Buyko, Rico Landefeld, Matthias Mühlhausen, Michael Poprat, Katrin Tomanek and Joachim Wermter. The Julie Lab UIMA Component Repository - An Overview and Experience Report. In *Proceedings of the Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP for NLP Workshop at LREC 2008*, pages 1–7, Marrakech, Morocco, May 2008.

Ekaterina Buyko, Christian Chiarcos, Antonio Pareja Lora. Ontology-Based Interface Specifications for an NLP Pipeline Architecture. In *LREC 2008 – Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 847–854, Marrakech, Morocco, May 2008.

Ekaterina Buyko and Udo Hahn. Fully Embedded Type Systems for the Semantic Annotation Layer. In *Proceedings of First International Conference on Global Interoperability for Language Resources (ICGL 2008)*, Hong Kong SAR, January 9–11, 2008.

Ekaterina Buyko, Katrin Tomanek, and Udo Hahn. Resolution of coordination ellipses in biological named entities using conditional random fields. *In PACLING 2007 — Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 163–171, Melbourne, Australia, September 19-21, 2007.

Udo Hahn, Ekaterina Buyko, Katrin Tomanek, Scott Piao, John McNaught, Yoshimasa Tsuruoka, and Sophia Ananiadou. An annotation type system for a data-driven NLP pipeline. In *The LAW at ACL 2007 – Proceedings of the Linguistic*

Annotation Workshop, pages 33–40, Prague, Czech Republic, June 28-29, 2007. Stroudsburg, PA: Association for Computational Linguistics, 2007.

Scott Piao, Ekaterina Buyko, Yoshimasa Tsuruoka, Katrin Tomanek, Jin-Dong Kim, John McNaught, Udo Hahn, Jian Su and Sophia Ananiadou. BootStrep annotation scheme – encoding information for text mining. In *Proceedings of Corpus Linguistics Conference 2007*, Birmingham, UK, 27-30 July 2007.

Ekaterina Buyko, Joachim Wermter, Michael Poprat, and Udo Hahn. Automatically adapting an NLP core engine to the biology domain. In *Proceedings of the Joint BioLINK-Bio-Ontologies Meeting. A Joint Meeting of the ISMB Special Interest Group on Bio-Ontologies and the BioLINK Special Interest Group on Text Data Mining in Association with ISMB*, pages 65–68, Fortaleza, Brazil, August 5, 2006.

Ekaterina Buyko. Numerische Repräsentation von Textcorpora für Wissensextraktion. In *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen, Beiträge zur GLDV-Tagung 2005 in Bonn*, Frankfurt am Main, Berlin, Bern, Bruxelles, New York, Oxford, Wien, 2005.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The holy grail of computational linguistics is to devise automatic solutions for high-performance and large-scale text analytics. As the computational linguistics community will struggle, in the foreseeable future, to produce applications that can completely "understand" the meaning of text, researchers are focusing first on developing programs with shallow text comprehension (cf. Hearst (1999)). These programs extract information with a predefined semantic structure, known as *Information Extraction* solutions. Information extraction is the process of analyzing text so as to find information assessed as being relevant to some interest, including the extraction of named entities and the relations between them. An important aspect of information extraction research is that the information extracted should be of relevance to applications. Another key aspect is that the information extraction solutions should be successful in large-scale real-life applications. Both aspects are investigated thoroughly in this thesis.

The origins of information extraction research date from the late 1980s. From the start, information extraction particularly emphasized *sublanguage* (Harris, 1991) analysis. On the one hand, the language of military messages was analysed in the first Message Understanding conferences (MUCs) series (cf. Grishman and Sundheim (1996)); on the other hand, clinical language was considered in medical informatics applications (Sager et al., 1987). The need for automatically detected relevant information structures from large (sublanguage) text collections is the driving force behind information extraction research. It is therefore not surprising that the language of bioscience came under the spotlight of computational linguistics and became an important information extraction application area in parallel to advances in large-scale genome sequencing in molecular biology (e.g., *Human* genome sequencing by Venter et al. (2001)). The outcomes of complex bioscience experiments are dispersed to the biomedical community using natural language as the principal medium. At the time of writing this thesis, PubMed[1], the major free literature

[1] http://www.ncbi.nlm.nih.gov/pubmed/

database in the biomedical domain, provides over 20 million publication citations and indexes about 700,000 new citations a year. Given such an overwhelming and continuously growing amount of published biomedical knowledge (cf. Baumgartner et al. (2007)), this thesis emphasizes research on dedicated information extraction solutions for the bioscience domain, and in particular for molecular biology as its key field.

The primary focus of molecular biology is on the networks of interactions between bio-molecules within cells considered as research units. Bio-molecules include, in particular, genetic molecules, e.g., DNA sequences of genes that can be transcribed into RNA and finally translated into proteins. This transcription of genetic DNA into RNA and the subsequent translation into proteins is summarized under the term *gene expression*. This gene expression activity in the cell plays a crucial role in forming new molecules in the form of proteins and is therefore responsible for the development of disease in some cases. Gene expression is thus carefully regulated by intracellular mechanisms. This is called gene expression regulation, the process that modulates the frequency, rate or extent of gene expression. The study of gene expression regulation is a core field of current research in molecular biology (cf. the Grand Challenge I-2 described by Collins et al. (2003)). This thesis, therefore, contributes considerably to research on information extraction in this important area of molecular biology. It considers the two aspects of information extraction introduced above, i.e., high relevance of extracted knowledge and large-scale data.

Information extraction solutions for the biosciences have long been synonymous with named entity recognition and normalization. This means finding instances of semantic classes and relating these instances to their conceptual identifiers in biomedical terminologies or ontologies. This task is followed by text analytics dealing with relationship and event extraction, i.e., finding relations that link named entities. Classic examples of such relations in the biosciences are interactions between proteins (PPI) or the association between a gene and a certain disease. Event extraction, which is the automatic identification of dynamic propositional relations (cf. Chapter 3 for the distinction between different relation types), is one special case of relation extraction and possibly the most difficult one. Events usually have a complex internal structure, for example involving temporal relations between several sub-events that make up a complex event. Event identification is necessary for the automatic construction of detailed molecular pathways (Oda et al. (2008)). Thus, event extraction is a current and future issue for the bio information extraction community and for that reason it is the primary focus of this thesis.

I shall first introduce the concepts underlying this thesis and describe the relevant research subjects. This thesis concerns event extraction from biomedical literature, in particular from texts on molecular biology. In this work I extensively examine what has been produced on event extraction in related research fields (e.g., theoretical and computational linguistics on general language), and extend these outcomes to concepts of molecular events in bio information extraction research. The conclusions reached through this investigation will help to model and guide the manual event annotation process of texts performed by domain experts. I also dedicate a substantial part of this thesis to designing, implementing and evaluating an advanced event extraction engine. The evaluation was executed on standard challenge data that is shared community-wide, and was complemented by a large-scale evaluation scenario on biomedical databases.

In summary, my research focused on three major questions. The first question concerns particularities of biomedical event descriptions in literature and potential methods for taking account of and capturing these intricacies within a manual annotation process (cf. Chapter 3). The second question relates to the design of an automatic event extraction engine (cf. Chapter 4), measuring the effects of syntactic pre-processing procedures on achieving this semantic task, and assessing whether the event extraction approach proposed functions as desired (cf. Chapter 5). The third question is whether the implementation of a high-performance event extraction engine, as presented, is able to solve a real-life problem such as the construction of a large-scale biomedical database (cf. Chapter 6).

In order to deal thoroughly with these interconnected research questions, this work uses the following structure. Chapter 2 explains the motivation for this work, acquaints the reader with the publication avalanche in the biosciences, and reports on the biocuration gaps that are widening throughout the community. This chapter stresses the development of automatic methods for information accessibility from the biomedical literature in the form of information extraction solutions. It presents the basic concepts of information extraction (IE) and the origins of IE research from the newspaper domain (such as MUC (Marsh and Perzanowski (1998)) and ACE (Doddington et al. (2004)). It will be shown that events have already been dealt with in event extraction competitions run on newspaper documents, considering topics such as military messages and management succession events or company takeovers. The focus then turns to the particularities of information extraction in the biomedical sublanguage domain. The concepts of named entity recognition, normalization and relationship extraction are explained, and prestigious IE evaluation competitions, including the BioCreAtIve (Hirschman et al., 2005) and the "BioNLP 2009 Shared Task on Event Extraction" (Kim et al., 2009) challenges are described.

Chapter 3 explains that different kinds of semantic relations are often mixed up in evaluation competitions. Therefore, this chapter aims to provide a lucid classification of semantic relations considered in the Natural Language Processing (NLP) domain and to single out semantic relations of interest for this thesis, *viz.* events.

The concept event is then considered from the points of view of diverse but related research areas, such as philosophy, general linguistics, and computational linguistics. In particular, this chapter investigates studies on the temporal and causal structure of events. Furthermore, it provides a comprehensive classification of event concepts from major research projects in computational linguistics, such as FrameNet (Baker et al., 2003) and TimeML (Pustejovsky et al., 2003). After this excursion into general language research, I focus on biomedical events and on how to deal with the intricacies of their descriptions, such as nesting or partial views of molecular processes. I present and discuss various available solutions. In this thesis I opt for an ontology-supported approach that can be applied to large-scale manual annotation projects whose outcomes have a direct connection to the large biomedical community. Manual annotation of corpora is a crucial part of event extraction research and it comes under the spotlight later in this chapter. I present the newly annotated corpus with gene expression regulation events, GeneReg, which was created as a part of the work described in this thesis. Furthermore, this thesis introduces two new concepts for event annotation, *viz.* entity-driven and trigger-driven annotation approaches. The advantages and disadvantages of both approaches are extensively discussed and linked in order to help the reader to make a choice between the two approaches in an annotation case.

Following my theoretical work on what precisely constitutes an event (in Chapter 3), Chapter 4 describes the automatic event extraction task itself. I first present NLP steps required for a high-performance event extraction solution, such as sentence splitting, tokenization, and morphological and syntactic analysis. Here I pay particular attention to the adaptations of available approaches from the newspaper domain to the sublanguage domain of biomedicine. I then describe the methodology for automatically extracting events from literature. This methodology considers dependency graphs to be the central data structure on which various trimming operations are performed, such as, on the one hand, syntactic simplification by pruning informationally irrelevant subgraphs. On the other hand, a further operation is semantic enrichment by conceptual decoration of those lexical nodes which are informationally particularly relevant for event extraction. The trimming methodology is complemented by manually curated dictionaries and machine learning (ML) methodologies (a feature-based and a kernel-based one) to sort out associated event instances and arguments on trimmed dependency graph structures. The event extraction approach developed in this thesis can best be characterized as a combined learning approach for event detection as it does not separate the overall learning task into independent event trigger and event argument learning subtasks.[2] The one-step learning approach to event detection, where event predicate identification

[2] This approach considers all relevant lexical items as potential event predicates which might represent an event.

and argument assignment are combined in a single process seems particularly appropriate for the partially annotated data characteristic of the biomedical domain corpora (cf. Chapter 3). At the end of this chapter, I introduce the implementation of this approach, the JReX (Jena Relation eXtraction) system. The JReX system has been applied by the Julie Lab team in the "BioNLP 2009 Shared Task on Event Extraction" competition and was ranked second among 24 competing teams, with 45.8% precision, 47.5% recall and a 46.7% F-score (Buyko et al., 2009). The evaluation studies of the JReX system are presented and discussed in the following chapter (Chapter 5).

Chapter 5 is dedicated to the extensive evaluation of JReX on the "BioNLP 2009 Shared Task on Event Extraction" data, the widely accepted evaluation standard in the bio information extraction community. The event extraction subtasks, *viz.* event trigger detection and argument extraction, are evaluated on this data. The effects of using various knowledge sources, such as lexical information, shallow syntax, complete dependency graphs, or shortest dependency paths only, are explored for solving the event extraction task. The shortest path hypothesis of Bunescu and Mooney (2007) is confirmed in these studies. After that I explore to what extent the performance of the JReX system depends on the choice of the parser and its output representations. Therefore, in this chapter I investigate an evaluation study of the impact on the event extraction task of various dependency representations and additional trimming procedures. The evaluation results achieved demonstrate that for the JReX machine learning approaches, the trimming operations might account for the shortest CoNLL dependency path only as this syntactic representation constitutes the major relevant knowledge source for the semantic event extraction task. Further detailed evaluations of trimming strategies show that they are beneficial in solving the event extraction task, in particular for complex events. These experiments with diverse dependency representations enabled me to measure their effects on the event extraction task and to increase the overall JReX performance in terms of F-score. After the official competition, the JReX system was updated and achieved 57.6% precision, 45.7% recall and a 51.0% F-score (Buyko et al., 2011a). JReX currently scores best on the U-Compare event extraction server (Kano et al., 2011), now outperforming the best system (Turku) from the "BioNLP 2009 Shared Task on Event Extraction" competition.

However, the next question that arises is whether the supervised event extraction approach developed in this thesis reached the limits of its effectiveness in the experiments. The application of supervised ML techniques (in addition to the engine implemented) is expected to require only sublanguage annotated corpora. This expectation makes this approach more attractive than a hand-crafted rule-based one that would necessitate a system developer acquiring domain expertise for the subsequent rule creation step. However, the performance of a supervised ML system might be constrained by the quality, size and variety of corpora used to train it. The quality of the data should be guaranteed by annotation guideline developers and by annotators, and is to a great extent monitored through inter-annotator agreements. The amount and variety of data created depend on time and cost limitations. Thus, in order to answer the question about the limits of the proposed event extraction approach, I investigate the learning progress of the JReX system on the "BioNLP 2009 Shared Task on Event Extraction" data at the end of Chapter 5. The results of this study demonstrate that there is great potential for the performance of this approach to improve, if more data to learn from is provided.

The evaluations from Chapter 5 are performed *intrinsically*, i.e., under clean experimental lab conditions on publicly available corpora. Although intrinsic evaluations are of high significance for the bio information extraction community, the benefits are unclear for the life science community, which works with large-scale life science data sets with high coverage. Therefore, the next Chapter 6 explores the robustness of the event extraction approach *extrinsically* in a real-life evaluation scenario targeting database curation. For this study, I chose the highly relevant topic of gene expression regulation in two model organisms, *E. coli* bacteria and *Candida albicans* fungus, which are the subject of very active research. The gold standard data for the *E. coli* organism has been gathered from RegulonDB, which is the world's largest manually curated reference database for the transcriptional regulation network of *E. coli* (Gama-Castro et al., 2011). The *Candida albicans* regulatory network, manually curated from the scientific literature at the Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute in Jena, is selected as a gold standard for the second extrinsic evaluation. Having both gold standards in mind, this chapter faces the challenging task of automatically reproducing database contents from the available scientific literature. Both evaluation studies demonstrate that JReX (re-trained on the GeneReg corpus) can automatically re-create considerable parts of manually curated databases and can even complement human curation efforts. The JReX system and corpora developed and implemented in this thesis support real-life needs of curators and bioinformaticians, thus helping to acquire and manage knowledge in the biomedical domain.

In summary, the main contributions of this thesis are situated on three levels. First, this work presents and critically discusses particularities of biomedical event descriptions in text and offers structures for taking account of and capturing biomedical events in manual text annotation process. The second contribution is on the development, extensive evaluation and demonstration of limits of a state-of-the-art supervised event extraction system which relies on dependency graphs as a major knowledge source. In this context, this work extensively exploits different dependency formats and trimming operations on dependency graphs as a general part of a semantic information extraction system. The third main contribution of this thesis is on the evaluation whether a state-of-the-art event extraction engine is robust in terms of performance in the challenging task of automatically reproducing the content of large biomedical databases.

Chapter 2

Motivation and Background

2.1 Avalanche of Published Biomedical Data

The application of next generation sequencing technologies now enables biologists to produce and examine increasing amounts of data in order to comprehend biological processes in organisms. The most ambitious project to date, the 1000 Genomes Project, pursues the study of the human genome by sequencing the coding regions of 1,000 genes of 1,000 people.[1] This project will provide an extraordinary representation of biomedically relevant genetic data, following in the footsteps of the Human Genome Project (International Human Genome Sequencing Consortium, 2001) and the efforts by Venter et al. (2001) in sequencing the human genome. This current work by academics and companies will generate six trillion DNA bases, which is 60 times more sequence data than that which has been published in DNA databases over the past 25 years (cf. The 1000 Genomes Project Consortium (2010)). The 1000 Genomes Project is a perfect showcase for the exponential growth in the amount of biological data. Rapid developments in sequencing and screening technologies are facilitating the production of a wealth of results by a number of large-scale studies around the world. Thus, Howe et al. (2008) consider the results of the 1000 Genomes Project to be only the "tip of the biomedical data iceberg".

High-throughput experimental methods, such as various DNA sequencing methods, DNA microarrays, and pull-down assays are used to generate massive quantities of data and require the application of statistics and computer science to interpret the data, in order to increase understanding of biological processes. Major outcomes from computationally intensive applications include drug discovery, sequence alignment, prediction of gene expression, gene regulation, and protein-protein interaction. Thus, it is not surprising that a dedicated research discipline, Bioinformatics, emerged rapidly by the end of the 20th century and has since established itself as

[1] `http://www.1000genomes.org/`. Please note that all URLs referenced in this thesis were accessed in January, 2012.

a challenging research field for data management, algorithm development and data mining in the biomedical domain.

The essence of experimental results is extracted with help of bioinformatics tools and is usually transferred immediately in its published form. According to this standard workflow, natural language is currently the principal medium for dispersing knowledge produced by genetics, biochemistry, and molecular biology to the worldwide biomedical community. There is a huge volume of work published in the domains of molecular biology and genetics. The main free literature database in the biomedical domain is PubMed[2], which is maintained by the U.S. National Library of Medicine at the National Institutes of Health. A major role of PubMed is to provide access to the Medline[3] database of citations and abstracts from approximately 5,000 life science journals that cover medicine, pharmacy, biology and biochemistry, among other fields, the most authoritative bibliographical database for the life sciences. At the time of writing this thesis, PubMed provided over 20 million Medline citations from 1950 to the present, held over 20 million article abstracts, and 2,2 million full-text articles in the PubMed Central[4] database. PubMed currently indexes about 700,000 new Medline citations a year, and the trend is upwards as illustrated in Baumgartner et al. (2007). It is evident that such a huge amount of literature will overwhelm researchers wanting to track all advances or to obtain efficiently an overview of their research fields or related fields.

This is the mission of specialized databases: to make research more productive in a timely manner. PubMed itself is only part of a cross-database information retrieval system called Entrez[5], which provides a search facility in databases dealing with topics such as gene, protein and transcript sequences, and bioactivity screens of chemical substances. Databases are crucial resources for biomedical data references and hypothesis analysis. Howe et al. (2008) reported from the year 2008 that there were nearly 750,000 visitors (unique IP addresses), who viewed more than 20 million pages in just one month. Access to nine prominent model-organism databases, together with Uniprot[6] and Protein Data Bank[7], are considered in their study. At present we can distinguish between two general types of database – organism-general and organism-specific databases. Organism-general databases contain biological knowledge about diverse species. The Uniprot database, for example, is based on protein sequences and contains information about protein functions from nearly 150,000 organisms including bacteria, eukaryota, and viruses. The second

[2] http://www.ncbi.nlm.nih.gov/pubmed/
[3] http://www.ncbi.nlm.nih.gov/pubmed/
[4] http://www.ncbi.nlm.nih.gov/pmc/
[5] http://www.ncbi.nlm.nih.gov/Entrez/
[6] http://www.uniprot.org/
[7] http://www.pdb.org/pdb/

database type, organism-specific, refers to databases with a view on a single organism, typically a model organism (non-human species). Model organisms, such as *E. coli* bacteria, *Mouse* and *Fly*, are extensively studied for an understanding of biological processes in one living organism, with the expectation of being able to generalize for processes in other living organisms, such as humans. The most prominent model organism databases are Mouse Genome DB[8], FlyDB[9], and RegulonDB.[10] RegulonDB will be a focus of this thesis (cf. Chapter 6).

Professionals, called biocurators, currently create and maintain databases of scientific knowledge from molecular biology, biochemistry, and genetics. Thus, their role is to support researchers in their use of biomedical data. The main tasks of a biocurator are to read published papers, find relevant biomedical knowledge, add this knowledge to a dedicated database, and correct potential inconsistencies and errors in the database. The curation task is known to be an extremely time-consuming and labor-intensive manual process (for example, the EBI sequence group has about 100 curators) (cf. Seringhaus and Gerstein (2007)). In order to illustrate curation challenges, we can imagine the curation of gene regulation interactions in the fungus organism *Candida albicans*. The PubMed search[11] retrieves more than 25,000 documents on the fungus *Candida albicans*, which then have to be read and analyzed by a curator team. Given a team of two curators and 250 working days in the year, one curator needs to read 50 journal articles per day. Given the annual growth rate of 3.1% in new entries in Medline (cf. Hunter and Cohen (2006)), this team will need to read hundreds of articles per year to keep the database status updated. Just the single example of this very dedicated task of curation for just one model organism demonstrates the great cost and effort of curation, particularly at its initial stage. Consequently, this approach cannot keep pace with the ever increasing publication output in the life sciences.

If we consider a more general curation task, such as curation of protein-protein interactions from various organisms, we can see that it is hardly feasible to curate such a database, given the traditional manual curation workflow. The curation team of the database Bind[12] have estimated that nearly 1,900 protein-protein interactions are published each month in 80 selected journals (Alfarano et al., 2005). Ramani et al. (2005) and Mathivanan et al. (2006) evaluated the available databases for coverage of human protein-protein interactions. Their surveys reveal that, although all these

[8] `http://www.informatics.jax.org/`

[9] `http://flybase.org/`

[10] `http://regulondb.ccg.unam.mx/`

[11] "candida albicans"[MeSH Terms] OR ("candida"[All Fields] AND "albicans"[All Fields]) OR "candida albicans"[All Fields]. This search query was applied at the PubMed web site in July 2011.

[12] http://www.bind.ca/

databases derive their knowledge from the same source (Medline), the sets have very few elements in common. Ramani et al. (2005) reported only 0.1% of interactions in common in available data sets. These findings imply that the number of protein interactions presented in the literature is huge and that the curated data sets are biased.

Although the curated databases are important resources for understanding biological processes, they cover only a small fraction of published biomedical knowledge. Howe et al. (2008) warns that the usefulness of curated data can be "seriously compromised" if the gaps between published data and curated data increase further. Generating and testing hypotheses will become ineffective. A number of biomedical researchers, for example Seringhaus and Gerstein (2007) and Howe et al. (2008), have raised the alarm about increasing biocuration gaps and propose urgent action to advance the biocuration process. First, they recommend closer collaboration between databases and journals. All journals should require, in addition to a published article, a machine-readable supplemental document or a structured digital abstract with at least approved gene symbols, model-organism database IDs, and pertinent facts for genes and proteins discussed in the paper. Second, journals should also mandate submission of data into dedicated databases as a part of publication. These proposals still concern only manual work in which human indexers and curators have to invest a lot of effort into searching diverse complex life science terminologies. This workload may overwhelm researchers and reviewers. Another hurdle would be a subjective bias of authors required to encode pertinent facts, which could lead to over- or understatements (cf. Hahn et al. (2007b)). The current curation situation requires revolutionary methods for information accessibility from the biomedical literature. Hahn et al. (2007b) promote applications of automatic text mining procedures that would provide reasonable support, since considerable progress has been made in this field over recent years. The text mining approach would help to avoid a supplementary workload and human subjectivity.

In the biomedical domain, the term *Text Mining* is often used to designate applications that exploit unstructured knowledge from biomedical literature (Cohen and Hunter, 2008). This concept of text mining is coarser than the widely accepted definition by Hearst (1999). In her definition, text mining is a process of detecting previously unknown information within written text resources.[13] Both conceptions agree that text mining can imply different activities that have independently distinct, but aligned, purposes for text mining: information retrieval, information extraction, and integration of textual and optionally non-textual data for hypothesis generation (which is the ultimate aim of text mining according to Hearst (1999)). Thus, text mining starts with the retrieval of relevant documents, continues with the extraction

[13] In this work, I use the term Text Mining in the sense of Hearst (1999).

of facts and, finally, links these pieces of information to generate new hypotheses for people to consider.

The biosciences are considered to be the most promising application area for text mining. The first text mining system in medical sciences was developed by Swanson (1986). The ARROWSMITH system, which is based on classifying co-occurrences of terms in document titles, had great success. It led to some hypotheses, such as an association between magnesium and migraine headaches (Swanson, 1988), which have been confirmed through experiments. The knowledge discovery techniques behind ARROWSMITH are distinct from Natural Language Processing (NLP) techniques because the former are based only on elementary word statistics that might be useful for document retrieval or document classification only. The NLP-based techniques offer more linguistic data for the text mining process itself. In the early 1990s, biomedical text mining systems started using diverse NLP techniques intensively, with a visible impact on molecular biology research. These systems focus primarily on activities such as intelligent information retrieval and extraction of facts from collected literature. Biomedical NLP systems, in the form of dedicated search engines such as IHOP (Hoffmann and Valencia, 2004), and database curation support systems such as PREBIND (Donaldson et al., 2003) or PAPERBROWSER (Karamanis et al., 2008), are emerging and some of them are heavily used. Two very famous systems that are illustrative of classic text mining systems according to Hearst (1999) and search engines for biological facts are CHILIBOT[14] (Chen and Sharp, 2004) and TEXTPRESSO[15] (Müller et al., 2004). CHILIBOT is a prime example of a biomedical text mining system that exploits information retrieval, information extraction and hypothesis generation techniques. CHILIBOT searches PUBMED for interaction relationships between pairs of genes, scans the network of retrieved relationships before proposing hypothetical relationships that are not documented. TEXTPRESSO is a full text search engine for biological entities and facts such as gene-gene interactions. In contrast to CHILIBOT, TEXTPRESSO does not provide hypothesis generation and thus can only be considered as a fact database generation system.

The fact extraction component is at the heart of most biomedical NLP systems. In particular, the focus is on generating interaction networks of biological entities (pathways). The extraction of such a complex network is the primary goal of fact extraction, known as *Information Extraction* in the NLP domain that is the focus of this work.

[14] http://www.chilibot.net/
[15] http://www.textpresso.org/

2.2 Information Extraction

2.2.1 Objective: What is Information Extraction?

Information extraction (IE) applications aim to find information assessed as being important to some interest, including extraction of named entities and relations. The information extraction process turns the unstructured information embedded in texts into structured data, or, more precisely, it fills slots in predefined templates with extracted information and populates the contents of a relational database. The term *unstructured* used as a modifier for *text* is, admittedly, misleading, because a text document can surely be interpreted as a structured object from a linguistic perspective, as it demonstrates, among other things, syntactic, semantic and discourse structures. I use the term *unstructured text* in the sense that the structure of a text is hidden for computers because the running text usually contains no meta-data annotations (no explicit tagging of its semantic structure).

An IE template is considered to be a semantically meaningful group of entities and relations with n slots, where $n > 0$. The IE process thus converts the document content to a set of entities and the relations between them. The term *relation* can designate static relationships or events that involve entities. Relationships usually do not change over time; events are dynamic and have a time stamp associated with them. The relationships can be classified according to relation types, such as `Contained_in`(nucleus, cell) or `Located_in`(Manhattan, New York). Temporal information does not usually play any specific role in these relational statements. An example of a relationship would be the information that a person is working for a company. Using the concept *event* we refer to situations that happen at a point in time or occur for a period of time, for example `Be_Born`, `Dance` and `Wedding`. The following examples contain a relationship (Example 2.1) and an event (Example 2.2):

(2.1) "*Mozart was employed as a court musician by the Prince-Archbishop Hieronymus Colloredo.*"

(2.2) "*Mozart was born on January 27th, 1756 in Salzburg.*"

The blank templates for an `Employment` relationship with two slots to fill and a `Be_Born` event with three open slots look like:

```
Relationship: Employment
Person employed: [ ]
Employer: [ ]
```

```
Event: Be_Born
Person: [ ]
Time: [ ]
Place: [ ]
```

From the first sentence we can fill the `Employment` template as follows:

```
Relationship: Employment
Person employed: [Mozart]
Employer: [Prince-Archbishop Hieronymus Colloredo]
```

The second sentence can be represented according to the `Be_Born` template as follows:

```
Event: Be_Born
Person: [Mozart]
Time: [January 27th, 1756]
Place: [Salzburg]
```

The filling of IE templates can be broken down into subtasks such as recognition of named entities (e.g., persons such as "*Mozart*", places such as "*Salzburg*"), relationships (e.g., `Employment`, `Located_In`) and events (e.g., `Be_Born`, `Marriage`) on textual data and, finally, instantiation of template slots.

The question "How accurate are the IE tools in filling templates from previously unseen data?" is usually answered in public challenges that use identical test data sets and apply a formal evaluation scenario. These challenges typically involve evaluation tasks that share the same design. Task organizers select a well-defined and relevant IE task, such as organization name identification or extraction of employment relationships. The training and test data sets are manually pre-annotated and comprise a gold standard against which the participant can evaluate the IE systems.

In the following I will present the most prominent information extraction challenges, which have influenced information extraction research.

2.2.2 Origins of Information Extraction: MUC and ACE

Message Understanding conferences (MUCs) were organized by the RDT&E Division of the Naval Command, Control and Ocean Surveillance Center, with the support of DARPA, the Defense Advanced Research Projects Agency, between 1987 and 1997. The primary motivation for MUCs was the formal evaluation of information extraction systems. Although they were called "conferences", the basis of MUCs was the challenge in which participants had to submit results from their IE systems in order to participate in the conference. The participants received sample texts (training data) and instructions on the type of information to be detected in order to develop or adapt IE systems. Shortly before the conference, participants had to process test data. The results were evaluated against manually-prepared answers. During the history of the MUCs, different application domains were selected for a challenge: naval operation messages (MUC-1, MUC-2), terrorism in Latin American countries (MUC-3, MUC-4), joint ventures and the microelectronics domain (MUC-5), news articles on management changes (MUC-6), and satellite launch reports (MUC-7).

Grishman and Sundheim (1996) provide an overview and brief history of the MUC series. The first two MUCs provided military messages about navigation and were exploratory for the IE task. MUC-2 worked out details of evaluation measures, namely recall, precision and F-score (see Glossary for definitions). MUC-1 to MUC-5 were organized with the ambitious goal of extracting templates with n slots, where the size of n increased during the first five series, and the application domain changed. These issues required a lot of time for the adaptation of systems by participants. It was noted about MUC that the systems were tending towards relatively shallow understanding techniques, which were based primarily on local pattern matching. Therefore, MUC-6 was organized with the aim of encouraging work to make IE systems more portable and to demonstrate work in "deeper understanding". MUC-6 broke down the IE task into several short-term subtasks, such as named entity recognition, coreference detection, template element extraction, and scenario template extraction. MUC-7 added the template relation task. Here, I will briefly present the MUC-7 IE subtasks and provide the results of the top-scoring systems. The MUC-7 subtasks are:

- Named entity: recognition of named entities of a predefined type (e.g., persons, organizations and locations).
- Coreference: extraction of noun phrases that refer to the same named entity.
- Template element: filling in small-scale templates for specified types of entities, such as persons or organizations. The slots for a person are, for example, title

Table 2.1: Maximum results reported in MUC-7 by task.

MUC-7 Task	Named Entity	Coreference	Template element	Template relation	Scenario template
F-score	94%	62%	87%	76%	51%

and nationality.

- Template relation: identification of facts. Filling in a template that represents binary relations between previously identified slot fillers in the Template element task. `Location_Of`(x,y) and `Product_Of`(x,y) are general examples of relations for this task.
- Scenario template: identification of events. Filling in a scenario template with previously identified entities. Management succession and capitalization of joint ventures are examples of events.

Table 2.1 shows the best F-score results from MUC-7 tasks (cf. Chinchor (1998)). The results demonstrate that named entity recognition could be seen as a solved problem, as the best system achieved 94% F-score performance. The systems achieved good results on the small set template element filling task with the best F-score of 87%. The template relation task demonstrates reasonable system performance with the top-scoring system, which achieved 76% F-score. The most challenging are the scenario template (51% F-score) and coreference tasks (62% F-score), with high error rates.

The MUCs are very remarkable because of the degree to which IE technology development has been encouraged and the degree to which the evaluation program has been defined. The last two MUCs defined the relevant IE subtasks, which continue to influence the focus of IE research.

The MUC series were followed by a new IE evaluation program called Automatic Content Extraction (ACE) (Doddington et al., 2004). The ACE program emphasized the study of a range of elementary named entities, relations, and events expressed in texts. The multilingualism (Arabic and Chinese languages in addition to English) and the high variety of textual media (newswire, broadcast conversation, and weblogs) should stress the portability and generalization characteristics of IE systems (which had been missed in the MUC series). The ACE program started with a pilot study in 1999 and organized evaluation tasks until 2008.

Table 2.2: Maximum results reported in ACE-2007 in mention detection task for English.

ACE Task	Entity Detection	Relation Detection	Event Detection	Temporal expression detection
Performance	82.9%	33.4%	24.1%	61.6%

The ACE program's main tasks are:

- Entity detection and tracking: detecting mentions of named entities of predefined types. Coreference detection is included here.
- Relation detection and characterization: detecting binary relation mentions of five types (ROLE, PART, AT, NEAR, SOCIAL) which are sub-divided, yielding a set of 24 types of relations.
- Event detection and characterization: detecting event mentions with n participating entities, where $n > 0$. Eight event types include LIFE, MOVEMENT, CONTACT, JUSTICE. Sub-divided into subtypes, the set yields 41 types.
- Temporal expression detection: detecting mentions of temporal expressions.

The striking difference between ACE and MUC is in the event/scenario template detection task. The MUC evaluations showed the complexity of the scenario template task. ACE events, therefore, achieved a simpler structure than MUC scenario templates. They contain a limited number of arguments with fixed roles, such as `Agent`, `Object`, `Source`, `Target`.

ACE introduced more complex evaluation values than MUC recall and precision-based evaluation. System performance is scored using a model of the application value of system output. This value (ACE-score) is the sum of the value of each system output unit (entity, time expression, relation or event) accumulated over all system outputs. The value of every single system output unit is computed by comparing its attributes with attributes of the reference (gold) object. Details of the ACE evaluation can be found in Doddington et al. (2004).

The ACE 2007 performance results for the best system in every task are published on the NIST website[16] and are provided here in Table 2.2. Although the entity

[16] `http://www.itl.nist.gov/iad/mig/tests/ace/2007/doc/ace07_eval_official_results_20070402.html/`

detection task achieved good results (82.9% ACE-score), systems still perform very poorly on other tasks. The results of ACE and MUC competitions cannot be easily compared because of different data sets, annotation and evaluation metrics. Still, we see that the named entity task is easier to complete than the relation and event tasks. In particular, the low performance results for the scenario template (MUC, 51% F-score for the best system) / event (ACE, 24.1% F-score for the best system) tasks still bear witness to the complexity of information extraction going beyond named entity (persons, locations) extraction.

MUC and ACE series are formative for the IE domain. While MUC series propagated IE templates and defined general comprehension for the IE domain, ACE challenge refined MUC relationship and concept of event. MUC and ACE series still target the newspaper, weblog or military language domains. In Section 4.2, I will give an overview of the current methods and approaches to solving relationship and event extraction tasks.

The idea of extracting information from texts is relevant for other language domains as well. For example, the idea of extracting structured templates from medical texts was demonstrated at the time of the first MUC conference (Sager et al., 1987). The focus of my work is on IE solutions for the biomedical language domain. Therefore, I will give an introduction to the special requirements for IE in the biosciences below.

2.2.3 Biomedical Information Extraction in a Nutshell

Information extraction from biomedical texts is a crucial application area for various IE techniques. The focus here is on the detection of biological named entities such as genes, proteins, diseases, drugs or organisms, and on the relations between named entities, mostly in the form of events. If we consider a concise world view as seen by biologists we can see why this is the case. Biologists (biomolecular biology) are primarily interested in the interactions between molecules (entities) as a complex network within cells (cf. Cohen (2010)). In texts biologists refer to these processes using entity names such as "*Il-2 protein*", "*T-cell*", and "*mouse*", and usually describe interactions using predicates such as "*mediate*", "*phosphorylate*", and "*regulate*".

The biomedical language (here in particular the language of biomolecular interactions) is a specialized language – *sublanguage* in terms of Harris (1991) – with specific informational content, structures, and regularities. It features a range of expert terms such as "*DNA assay*", "*Lipid metabolism*", and manifests modified meanings of verbs from the general English language ("*express*", "*mutate*") or even new verbs

Table 2.3: Selection of the named entity types from the biomedical domain.

NE Type	**Examples**
Genes/Proteins	*Il-2, BRCA1, breast cancer associated 1*
Diseases	*breast cancer, Alzheimer, osteoporosis*
Drugs	*alendronate, aspirin, CDDP*
Chemicals	*claversal, C5-H-Cl3-N2-O, menaphthon*
Cell types	*T-cell, natural killer cells, NK cells*
Organisms	*human, mouse, Candida albicans*

(e.g., "*phosphorylate*") (cf. Section 3.5.5 for more examples). The application of the sublanguage theory to the biomedical domain has been extensively discussed by Friedman et al. (2002) and Harris (2002), for example. The major idea of this theory is the following. In order to extract the information from a sublanguage, Harris proposes representing the information content and structure of a scientific sublanguage in the form of a *sublanguage grammar* suitable for computation. The IE systems model such a sublanguage grammar by and large in the form of patterns, rules or statistical models (as done in this thesis). Given Harris's theory, the re-usage of newspaper models from MUC and ACE-driven approaches without domain adaptation would be inappropriate. In the following I introduce special features of IE tasks required from this sublanguage domain.

I start with presenting the biomedical Named Entity Recognition (NER). Detecting and characterizing mentions of biological entities is a preliminary step in the detection of relationships between entities using text. There is a much wider range of relevant named entity types in the biomedical domain than the newspaper-style set of MUC types (`Person`, `Organization`, and `Location`). A selection of biomedical entities is presented in Table 2.3. Diseases, drugs, organisms, genes, and proteins are of great interest to researchers. Gene names are in particular focus of IE applications (cf. Section 2.2.4).

However, detection of entity names only is usually not sufficient for information extraction because entity names are highly polysemous and can refer to completely different entities (for example genes from different species). Thus, an important challenge for biomedical NLP is to normalize entity name mentions by mapping entity names to unique identifiers in databases, for example (e.g., EntrezGene[17] IDs).

[17] `http://www.ncbi.nlm.nih.gov/gene/`

Table 2.4: Selection of PPI examples from the AIMED corpus (Bunescu et al., 2005).

PPIs	**Text**
p53 – TAFII40 *p53 – TAFII60*	"*p53 transcriptional activation mediated by coactivators TAFII40 and TAFII60*"
INF-alpha – Il-4	"*Cytokines measurements during IFN-alpha treatment showed a trend to decreasing levels of IL-4 at 4, 12, and 24 weeks.*"
hTAFII250 – TBP	"*Recombinant hTAFII250 binds directly to TBP both in vitro and in yeast.*"
Ras – Raf1 *Ras – Cdc25* *Raf1 – Cdc25*	"*We suggest that activation of the cell cycle by the Ras/Raf1 pathways might be mediated in part by Cdc25.*"
p53 – TFIID *p53 – TAFII60* *p53 – TAFII40* *TFIID – TAFII60* *TAFII40 – TAFII60*	"*Here, a direct interaction between the activation domain of p53 and two subunits of the TFIID complex, TAFII40 and TAFII60, is reported.*"
c-Fos – c-Jun *c-Fos – TBP* *c-Jun – TBP*	"*We propose that c-Fos and c-Jun proteins function as transcriptional activators, in part by recruiting TBP to form complexes to initiate RNA synthesis.*"

Entity normalization is a challenging issue because of the rich variety and considerable ambiguity of entity names (cf. Section 2.2.4 for task evaluation). Cohen (2010) gives some characteristic examples of gene name variety and ambiguity. For example, the *Brac1* gene is referred to by many different names - e.g., "*IRIS*" and "*PSCP*". All *Brac1* gene name synonyms have to be mapped to a unique ENTREZGENE identifier. As for the polysemous gene names, the *TRP1* gene, for example, can refer to five different genes. The full names of these genes are, for example, "*transient receptor potential channel 1*" and "*transfer RNA proline 1*". This means that, in different contexts, the *TRP1* gene has to be mapped to different ENTREZGENE identifiers.

The focus of this work is on higher IE tasks than named entity recognition and normalization, namely on relationship detection, in particular event extraction. Relationship extraction in the biomedical domain is at least as challenging as this problem is in the newspaper domain evaluated in the MUC and ACE competition series. In order to illustrate the intricacies of relationships in the biomedical domain, I provide some examples that contain descriptions of protein-protein interactions (PPIs) (Table 2.4). Although PPIs are only binary relations, their wide description variety in texts makes the PPI extraction task one of the trickiest. In order to demonstrate the inherent complexity of PPIs, let us consider an example from Table 2.4 "*p53 transcriptional activation mediated by coactivators TAFII40 and TAFII60.*" This sentence contains descriptions of two interaction relations, namely `PPI`(p53, TAFII40), and `PPI`(p53, TAFII60). More specifically, *TAFII40* and *TAFII60* "*mediate the activation of the transcription*" of *p53*. Hence, at a deeper level of consideration, the interactions mentioned in the sentence boil down to two specific molecular events[18], namely regulation and transcription, which are more precise than a protein-protein interaction. Obviously, PPI extraction can be broken down further into event extraction of various molecular events and even cascades of events (formally expressed by nested relations), both of which are hard to sort out.

It is evident that the quality control and benchmarking of biomedical IE systems might require complex evaluation scenarios. This will be illustrated in the next section.

2.2.4 Benchmarking of Information Extraction Systems for Biosciences

The expansion of the biomedical NLP as a relatively young and dynamic research branch in the NLP domain has witnessed a range of prestigious information extraction challenges and research projects, including the BioCreAtIve (Critical Assessment of Information Extraction systems in Biology)[19] (Hirschman et al., 2005) and the "BioNLP 2009 Shared Task on Event Extraction" [20] (Kim et al., 2009) challenges that are presented in this section.

Named Entity Recognition and Normalization

Quality control of IE technologies in the biomedical domain is carried out during public evaluation tasks similar in spirit to MUC and ACE competitions in the newspaper domain. Initially, the focus has been in particular on gene mention recognition and gene normalization. The evaluation tasks of JNLPBA (Joint Workshop

[18] Molecular event is here considered as a change in the biological state, properties or the location of a bio-molecule (e.g., proteins, DNA, RNA or cells)(Kim et al., 2009).

[19] `http://biocreative.sourceforge.net/`

[20] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/`

on Natural Language Processing in Biomedicine and its Applications) (Kim et al., 2004) and BioCreAtIve have shown that biomedical NE recognition performance in terms of F-score is lower than its MUC counterpart. It has not yet been possible to achieve the state-of-the-art NER performance of the newspaper-style NER. In the sixth MUC, the highest-performing system achieved 96% recall and 97% precision (matching the human Inter-Annotator Agreement rate) (Sundheim, 1995). The best-performing system in BioCreAtIve II (gene mention recognition task) achieved only 86.0% recall and 88.5% precision (Ando, 2007). Nearly all of the methods used in newspaper-style NER have been also applied to the biomedical NER problem. In general, three basic approaches are used and can be combined to create hybrid approaches: dictionary-based, rule-based and machine learning-based. The dictionary-based approach finds named entities listed in standard nomenclatures such as Uniprot (UniProt Consortium, 2008). The rule-based approach applies manually constructed rules or patterns to find mentions of named entities. The machine learning approach (mostly supervised) employs machine learning techniques such as Hidden Markov Models, Maximum Entropy or Support Vector Machines to build statistical models for NER. Typical features include lexical, morphological and contextual information. NER studies find the biomedical NER task to be very challenging. Yeh et al. (2005) discuss possible reasons for a lower performance, such as differences in the length of gene names and MUC entities, or inconsistencies in training data.

Gene normalization task performance lags behind that of gene mention recognition task. Methods for solving the gene normalization task are usually hybrid approaches integrating a range of background information such as dictionaries, gene definition fields or gene summaries, pattern-based similarity measures (Hakenberg et al., 2008) and machine learning-based similarity scores between pairs of genes (Tsuruoka et al., 2007). The BioCreAtIve II statistics give an impression of the varying complexity of the tasks for different species. For example, yeast has a smaller genome than mouse and human genomes, and yeast gene names are shorter (cf. Morgan et al. (2008)). These factors seem to contribute towards the better performance on yeast gene names (up to 92% F-score). The most challenging gene names to map are gene names of human and mouse genomes. The highest-performing system in BioCreAtIve II (human) achieved 81% F-score, while the top-scoring system in BioCreAtIve I achieved only 79% F-score on the mouse data (cf. Morgan et al. (2008)). The result on the human data has been outperformed in a follow-up study to BioCreAtIve II by Hakenberg et al. (2008) and Wermter et al. (2009), with gene normalization performance peaking at an F-score of 86.4%. The BioCreAtIve II.5 and BioCreAtIve III gene normalization evaluations showed that gene normalization task results are lower due to the complexity of full texts (instead of abstracts, as in BioCreAtIve I and II gene normalization tasks) and requirements for species disambiguation (cf. Leitner et al. (2010); Lu and Wilbur (2010)).

Relation Extraction

The BioCreAtIve challenge further organizes various protein-protein-interaction tasks with the aim of qualifying the systems for automatic database curation for PPIs from full text articles. Three subtasks have been presented to the public to date – the Interaction Pair Subtask (IPS), the Interaction Methods Subtask (IMS), and the Interaction Sentences Subtask (ISS). The IPS requires extraction of pairs of interacting proteins from running full texts. The interaction protein mentions must be detected and normalized. Thus, error propagation from the protein name mention detection and normalization task should be taken into account here if systems performance is examined. The IMS is an extension of the IPS and requires the extraction of experimental methods used for detection of PPIs and their normalization. The ISS is concerned with detection of the most relevant sentences containing salient descriptions for a given protein-protein interaction pair. The main PPI task is the IPS. The best system in the IPS from BioCreAtIve II achieved 28.8% F-score in the official run on the BioCreAtIve test data. In BioCreAtIve II.5 the organizers evaluated systems through exploration of the FEBS Letters experiment, which aims to generate structural digital abstracts for each published full article (Ceol et al., 2008). The three most successful teams from the IPS used quite different approaches, such as the pattern-matching approach, or supervised approaches exporting shallow or deep parsing information (cf. Leitner et al. (2010)). The IPS was substituted in the next BioCreAtIve III by the IMT (Interaction Method Task) task, which required only the extraction and ranking of experimental methods used for PPIs described in articles.

PPI extraction is clearly not a problem that has been solved and, given its inherent complexity (cf. Table 2.4), it may benefit from a methodological approach that deals with the extraction of molecular events in a bottom-up manner, so that the general PPI problem can be broken down into more specific and more feasible subtasks. The main task of the "BioNLP 2009 Shared Task on Event Extraction" competition, "Event Detection and Characterization", required, for a sample of Medline abstracts, that systems determine events mentioned and link them appropriately with protein annotations. The demands placed on text analytics to deal with the complexity of this Shared Task in terms of relation diversity and specificity are unmatched by earlier information extraction challenges. In the main subtask, 42 teams participated and 24 of them submitted final results. The winner system, Turku (Björne et al., 2009), with a 51.95% F-score, achieved the milestone result in that competition, followed by the Julie Lab team (Buyko et al., 2009), which peaked at a 46.7% F-score. The latter work on event extraction will be presented in this thesis (cf. Chapters 4 and 5).

2.2.5 Summary

In summary, this chapter introduced the origins and major ideas of information extraction from ACE and MUC departments, and explained an increasing need for information extraction solutions in the biosciences. It was accentuated that the biomedical language, a sublanguage in terms of Harris (1991), requires information extraction solutions that take into account special features of the biomedical text. The focus of this thesis is on the most challenging information extraction task (as revealed through the ACE, MUC and "BioNLP Shared Task on Event Extraction" (Kim et al. (2009), Kim et al. (2011)) competitions), namely event extraction, a special form of relation extraction. The next chapter will introduce concepts of event from research fields relevant to information extraction such as theoretical linguistics and computational linguistics. After that, it will zoom on intricacies of event descriptions in the biomedical domain with a special attention to molecular biology.

Chapter 3

Concepts of Event in Biomedical NLP

In the information extraction competitions introduced in the previous chapter such as ACE, “BioNLP 2009 Shared Task on Event Extraction” or BioCreAtIvE, different kinds of semantic relations are often mixed up. This chapter aims to generalize about semantic relations considered in NLP in general and to create a comprehensive classification of these relations (Section 3.1). After that, this theoretical work is extended with an elaborate overview about concepts of events in related research fields, e.g., philosophy, theoretical linguistics and computational linguistics (Sections 3.2, 3.3 and 3.4). The outcomes of these explorations serve as a basis for analysis of concepts of events in the biomedical domain in conjunction with special characteristics of molecular events considered in this thesis (Section 3.5).

3.1 Classification of Semantic Relations

Using the categories *relationship* and *event* (cf. Section 2.2.1) we can establish semantic *relations* between objects. Relation extraction can generally be considered as a recognition task where statements of the form $R(e_1, e_2, \ldots, e_t)$ have to be determined, with predicate R denoting a relation type (name), and $e_i, i = 1, \ldots, t$ usually, though not necessarily, denoting instances of named entity types. For proper relation extraction, it is not only necessary to identify the arguments and join them with a suitable relation name but also to determine the order among the arguments (roles played by arguments).

In general, we shall distinguish at least two main categories of semantic relations, i.e., *terminological* and *propositional* relations. Terminological relations characterize intrinsic conceptual relations between semantic entities at the level of concept definitions (such as `Is-a` or `Part-of` relations). Propositional relations deal with non-intrinsic conceptual relations which reflect empirical statements that hold in a particular domain of discourse, *static* ones such as `Has-weight` or `Located-in`, as well as *dynamic* (*eventive*) ones such as `Binds` or `Regulates`. Static propositional relations do not imply any causal relationships between participants or any state change of the participants in a relation. Unlike static propositional relations, dynamic ones describe transitional state changes of the arguments or of their properties. These relations are usually referred to as *events*. Terminological and static propositional relations are not a focus of this thesis. However, I would like to introduce these relations in order to distinguish them from events.

3.1.1 Terminological Relations

Terminological relations characterize intrinsic conceptual relations. The following two examples represent terminological relations:

(3.1) `Hyponym`(rose, flower)

(3.2) `Part-of`(nucleus, cell)

These predicates can be interpreted as follows. A *rose* is known to be a *flower*. A *cell* has a *cell nucleus* as a part.

In NLP, there is a range of research work on automatic extraction of terminological relations from text (e.g., Snow et al. (2005), Snow et al. (2006), Girju et al. (2006)). The *Hearst pattern* approach is the best known (Hearst, 1992). An example of the Hearst pattern is 'NP$_0$ such as NP$_1$, NP$_2$ ... (and | or) NP$_n$', meaning that for all NP$_i$, $1 \leqslant i \leqslant n$, `Hyponym`(NP$_i$, NP$_0$) (cf. Hearst (1992)). This pattern would, for example, give a match in the clause: "*Proteins such as NF-kappaB, and IL-2*" and outputs hyponym relations `Hyponym`(NF-kappaB, Protein) and `Hyponym`(Il-2, Protein). The main purpose of extracting terminological relations from text is to extend existing terminologies or identify new ones.

Furthermore, under terminological relations I subsume the semantic relations inside *intransparent* noun compounds. Noun compounds are combinations of two or more nouns. They are written as separate nouns ("*night frost*"), as words connected by a hyphen ("*father-in-law*"), or even as one word ("*doormat*"). I distinguish two kinds of noun compound structures, *intransparent* and *transparent*. Intransparent noun compounds have highly variable semantic relations between the nouns. They can indicate, e.g., what something is for ("*trash folder*") or what something is made from ("*wood hairbrush*"). Levi (1978) presents general noun compound relations which are produced by nine recoverably deletable predicates such as `Cause`(*flu virus*), `For`(*headache pills*), or `About`(*adventure story*). In contrast, transparent noun compounds can be detected in corpora with the help of paraphrasing procedures. For example, "*Il-2 expression*" can be reformulated as "*expression of Il-2*". Transparent noun compounds can thus easily be paraphrased (e.g., using preposition *of*), occur frequently in paraphrased forms, and usually represent static or eventive propositional relations (Sections 3.1.2 and 3.1.3).

In computational linguistics, the SemEval evaluation of semantic relation extraction within noun compounds was undertaken first in 2007 (Girju et al., 2007) and followed up in 2010 (Hendrickx et al., 2010). The evaluation results indicate that the integration of information extracted from established lexical resources is very useful, and none of the classification methods was particularly better than the others in this evaluation (Hendrickx et al., 2010). In bio-medicine only one work focused specifically on noun compound analysis. Rosario and Hearst (2001) classify noun compounds from medicine using the MeSH[1] hierarchy for normalizing nouns participating in semantic relations. They introduce a set of 38 semantic relations which describe noun compounds (e.g., `Instrument`, `Purpose`, `Measure_Of`). For example, "*vaccine antigen*" contains an `Instrument` relation between "*vaccine*" and "*antigen*".

3.1.2 Static Propositional Relations

Static propositional relations denote properties or stable relations between named entities (e.g., `Located-in`, `Employment`). Static relations have no internal causal and temporal structures and they are always true in a discourse (at a given interval and any sub-interval of a discourse). For example, the text "*Jean Philippe Courtois is director of Microsoft*" describes a static propositional relation `Employment` between "*Jean Philippe Courtois*" and "*Microsoft*", and can be represented as a template:

```
Relationship: Employment
Person employed: [Jean Philippe Courtois]
Employer: [Microsoft]
```

The ACE program (Section 2.2.2) is the main organizer of challenges for extracting static propositional relations from ordinary English language. ACE provides annotations of semantic relations between named entities in newspaper texts (see Section 2.2.2 for ACE relation types). Static propositional relations among ACE annotations are usually expressed inside a noun phrase. The possessives, preposition phrases and noun modifiers are frequently used in descriptions of static relations (Table 3.1). Consequently, an automatic extraction approach for ACE relations might be more effective with morpho-syntactic information than extraction of SemEval terminological relations.

[1] `http://www.nlm.nih.gov/mesh/`

Table 3.1: Syntactic classes presented in ACE relation annotation guidelines. Argument 1 is marked in bold. Argument 2 is underlined.

Syntactic Class	Relation Type	Arguments
Possessive	*Employment*	*Time Warner's* ***director***
Preposition	*Employment*	*The* ***director*** *of Time Warner*
PreMod	*Employment*	*European union* ***employees***
Coordination	*Family*	***He*** *and wife Eve*
Formulaic	*Employment*	*Apple Leader* ***Steve Jobs***
Participial	*Located*	***Apartments*** *located in Boston*
Verbal	*Employment*	***She*** *had worked at Google.*

In biomedicine, Pyysalo et al. (2009) focused on static propositional relations and argued that these semantic relations are highly relevant for current biomedical information extraction. They introduced the annotation of static propositional relations in biomedical texts, e.g., (`Part-Whole` and `Variant`). The `Part-Whole` relations are classified into four sub-types – `Object-Component` (e.g., "*Il-2 promoter*"), `Component-Object` (e.g., "*p50-p65 complex*"), `Member-Collection` (e.g., "*cytokines IL-6 and IL-8*"), and `Place-Area` (e.g., "*beta-globin locus*"). The `Variant` relation is used to annotate variants of genes and proteins such as mutants or isoforms ("*Il-2 mutant*"). In a similar way as for ACE annotations (cf. Table 3.1), the static propositional relations are captured inside noun phrases or even basic noun chunks.

3.1.3 Events

Events occur at some point in time or for a period of time, and usually allocate participants, except when they are zero-argument events such as `Snow` in "*It is snowing*". Unlike static propositional relations, events describe transitional state changes among the arguments involved or their properties, or they describe activities as sequences of changes.

I distinguish in my thesis between three terms, *event*, *predicate-argument relations* and *eventive propositional relations*. All three categories are connected in the *event* concept, and are explained in the following. Usually, events are expressed in natural language with the help of predicates that allocate arguments by assigning semantic roles. These relations between event predicate (e.g., verbs) and arguments are called *predicate-argument relations* and are the focus of *semantic role labeling* (SRL) methods that automatically assign roles to arguments of a predicate. Furthermore, if an event involves more than one argument, semantic relations between arguments involved in the event can be inferred. I call these relations *eventive propositional relations*. Both relation categories are exemplified with the help of the sample sentence "*IclR also represses iclR.*". This sentence contains an `Repression` event represented as a template below:

```
Event: Repression
Agent: [IclR]
Patient: [iclR]
```

This event is expressed using the predicate "*repress*" and two predicate-argument relations `Agent`(repress, IclR) and `Patient`(repress, iclR). The eventive propositional relation `Repression` between the entities "*IclR*" and "*iclR*" can be easily inferred.

Although the concept event is easy to understand, is intuitive and in linguistics can usually be defined in short as "things that happen", there are various concepts of event in different research fields, and in contrast to the concepts *entity* and *property*, this concept is still debated in philosophy and linguistics. This is the temporal dimension that makes the category event hard to determine. The next sections should bring to light the discussions in philosophy in linguistics over the category event.

3.2 Concepts of Event in Philosophy

3.2.1 Event as a Metaphysical Category

In order to define *event*, philosophers use two general approaches. The first of these is to compare events against well-defined philosophical concepts such as entities and properties. The second is to identify conditions under which two events are identical. I start with the introduction of the first approach in this subsection.

At first glance there are a lot of differences between events and entities. Events *happen* while entities *exist* in time, events have distinct temporal boundaries and indistinct spatial boundaries while entities have indistinct temporal boundaries and distinct spatial boundaries, and entities can move while events cannot (Casati and Varzi (2010), p. 3). But the distinction between entities and events disappears if events are considered as four-dimensional entities with a fourth dimension *time* (Quine, 1960). In this sense, (Grenon (2006), p. 156) considered events as entities that "persist in time through the succession of their temporal parts".

In contrast to the event as *entity* approach, other philosophers consider events to be a kind of *property* as "properties of moments or intervals of time" (Montague, 1969) or "particularized property located at some region of space-time" (Bennett, 1996). Concurrently, it is agreed that events and entities are considered as *individuals* with temporal and spatial location (Davidson, 1967) or, sometimes, as exemplifications of properties by objects (Kim (1976) as cited by Casati and Varzi (2010), p. 5). However, there is still debate on whether *entity* or *property* should be decided upon. In my work I prefer to consider the solution made by Davidson (1980) who analyzed events by defining their identity as things in space-time (see below). The consideration of events as things enables quantification over events. This is essential for linguistics and computational linguistics, and thus for this work. I will present it in the next subsection.

3.2.2 Davidsonian Event Concept

Davidson (1967) analyzed events by considering action sentences and speculated about event identity (their non-duplication criteria). He considered two criteria for identifying events, the *causal* and the *spatiotemporal* ones.

- The causal criterion says that events are identical if they have the same causes and effects.
- The spatiotemporal criterion says that events are identical if they occur in the same space at the same time.

Davidson was able to apply these criteria to events only by considering them as *things* or *spatiotemporal individuals*, i.e., particular non-repeatable occurrences. Davidson's idea was that the same occurrence of an event can be described in a number of ways. He therefore used the principle of extensionality in order to show, for example, that an eclipse of the Morning Star is an eclipse of the Evening Star because the Morning Star and Evening Star are identical (Davidson (1980), p. 120). Davidson emphasized that "spatiotemporal areas do not distinguish" events and entities, "but our predicates, our basic grammar, our ways of sorting do." (Davidson (1980), p. 176).

Davidson insisted that we can describe events in a number of ways by using action sentences. In order to ensure the identity of an event, he introduced an additional argument position for events, an event variable which is not realized at the linguistic surface and is existentially bound in clauses as in the logical form of the sentence "*I flew my spaceship to the Morning Star.*" expressed as

(3.3) $(\exists x)$ $(\texttt{Flew}(\text{I, my spaceship}, x), \texttt{To}(\text{the Morning Star}, x))$, where x "consists in the fact that I flew my spaceship to the Morning Star" (see Davidson (1980), p. 117).

Davidsonian consideration of events as things explains the inferential properties of natural language. In particular, the idea of event variables allows quantification over events. As the Example (3.3) contains an event variable x bound to the predicate `Flew`, it can easily be inferred from the Example (3.3) to

(3.4) $(\exists x)$ (`Flew`(I, my spaceship, x)).

The latter inference is possible because Davidson separated arguments bound by prepositions from the basic event structure. Davidson has greatly influenced linguistic research on events. The Davidsonian idea of separating some arguments from the basic syntactic verb structure has culminated in the Neo-Davidsonian program, where event verbs are represented as one-place event predicates (e.g., Higginbotham (1985), Kratzer (1995)). Thus the sentence introduced above would be represented as

(3.5) $(\exists x)$ (`Flew`(x), `Agent`(x, I), `Theme`(my spaceship, x), `To`(the Morning Star, x)),

which allows the inference

(3.6) $(\exists x)$ (`Flew`(x), `Agent`(x, I), `To`(the Morning Star, x)).

Maienborn (2011) reviewed the development of Davidsonian ideas and emphasized two major points in the Neo-Davidsonian program. First, the event is the only argument of a verbal predicate (as presented above in (3.5) and (3.6)). This idea has become a kind of standard in modern event semantics. Second, neo-davidsonians extended the definition of event arguments from action verbs alone (as made by Davidson) to adjectives, nouns and prepositions. However, the status of static verbs is still controversial and open to debate (cf. Section 3.3.4).

In summary, the davidsonian idea of events as *spatiotemporal individuals* that are captured by an extra event argument not visible on the linguistic surface has, over recent decades, influenced linguistics and computational linguistics research on event semantics. I present in the next section the most important concepts of events from the perspective of theoretical linguistics.

3.3 Concepts of Event in Linguistics

While philosophers are preoccupied with defining events as a metaphysical category, linguists cope with details of event semantics encoded in language with the help of cognitive models of temporal and causal information representation and linguistic categories such as aspect or transitivity. Linguists are, in particular, interested in exhaustive classification of events and in a uniform representation of event structure for quantification over events. For the following sections I selected four important research questions on events from theoretical linguistics. These are relevant for a better understanding of the event concept refined for this thesis. My focus is mainly on the introduction to the causal structure of events (Section 3.3.1), complemented with research on roles of arguments involved (Section 3.3.2), on the temporal event structure as a counterpart of its causal structure (Section 3.3.3), and on the controversially discussed distinction between states and events (Section 3.3.4).

3.3.1 Causal Event Structure

The causal event structure of events has been studied with the help of the category of causal event chain and primitive predicates.

Croft (1990) introduces the concept of a causal event chain and proposes three basic event views of a single event, e.g., the *causative*, *inchoative*, and *stative* views. While verbs correspond, in general, to one of these three event view types, subjects and objects in a sentence correspond to the participants in the causal chain. Thus, agent, patient, and force transmitted from agent to patient build a structure which can be represented completely, or only partially, by verbs. The causative event view (complete view of the causal chain) is represented by transitive verbs. The inchoative view (segmented view of the transmitted force and the patient) is represented by intransitive verbs, and the stative view (segmented view of the state of the patient after force transmission) corresponds to stative verbs and adjectives. The major contribution of Croft's classification is its consistent grounding in causation information, i.e., in a causal chain between the participants sharing an event.

In order to represent the causal chain of events in detail, linguists introduced a set of primitives that could capture the general semantic properties of events. Primitive predicates such as BECOME, CAUSE, and BE and logical operators have been used for the representation of event semantics (e.g., Dowty (1979), Jackendoff (1990)). For example, the biomedical verb "*express*" would be represented as follows:

(3.7) $\texttt{BECOME}[\texttt{BE}[\texttt{Available}(x)]]]$

where $x \in Gene$ (set of all genes).

This primitive representation helps to classify verbs into groups because similar verbs seem to have a similar primitive conceptual structure.

An important part of the causal tripartite structure (the causal chain of Croft (1990)) are the participants of an event. Thus, another approach to study event structure is on classification of event participants in accordance with their roles in events. This is explained in the next section.

3.3.2 Semantic Roles

Linguists argue that event argument structures reflect the lexical properties of predicates or our conceptualization of event categories which are universal. For example, Fillmore (1968), who proposes one of the earliest theories about the realization of arguments, argues in his Case Grammar that realized argument roles (`Agent`, `Patient`, `Instrument`, `Goal`) are determined by the lexical properties of a predicate, and calls these roles *theta* roles. Within the framework of the Government and Binding theory, theta roles are considered to be ordered in a thematic hierarchy with the highest role being `Agent` and the lowest `Manner` and `Location` (cf. Jackendoff (1972)). The realization of syntactic arguments depends on the position of corresponding theta roles in this hierarchy. In contrast to the theories focused on lexical properties of verbs in the form of thematic role sets and hierarchy, Dowty (1991) argues for only two universal proto-roles for describing eventive structure, the `Proto-Patient` and `Proto-Agent`. Both proto-roles are characterized by a number of properties such as "volitional involvement in the event or state" (`Proto-Agent`) or "undergoes change of state" (`Proto-Patient`). The latter approach is relevant for this thesis work (cf. Section 3.6).

3.3.3 Temporal Event Structure

In addition to the analysis of causal structure and semantic/thematic roles, linguists have focused on the analysis of the temporal information encoded in events. One of the most prominent works here has based verb classification on four universal *situation* types, which are *states*, *activities*, *accomplishments* and *achievements* (Vendler, 1967), which are defined as follows:

- *States* are durative (extend over time) and do not include any changes or culminations. Verbs such as "*contain*", "*believe*" or "*know*" describe states.

- *Activities* are durative, like states, but they describe sequences of changes in the world without any culmination and without an endpoint. Verbs such as "*swim*" or "*run*" describe activities.
- *Accomplishments* are durative and dynamic, like activities, but they include a clear point of an activity. For example "*climb the mountain*" or "*draw a circle*" are typical accomplishments. Usually this clear endpoint is involved by a patient, as in the previous examples.
- *Achievements* are dynamic and, like accomplishments, include a clear endpoint of an activity and always represent culminations of an event. For example "*reach the top of the mountain*" is an achievement.

Frequently, Vendler's four-category classification is simplified to a three-category system with *accomplishments* and *achievements* combined into one class of *performances*. However, the main distinction made by Vendler and other researchers is the distinction between states and events (e.g., Vendler (1967), Dowty (1991), Jackendoff (1990)).

3.3.4 Events versus States and Facts

The insistence on separating states from events is grounded in different conceptual structures behind states and events. Two major criteria are used for distinguishing between events and states: causal information (Section 3.3.1) and, as its counterpart, temporal information (Section 3.3.3). As for the causal information, eventive verbs have an internal causal structure, while it is absent in stative verbs. There are verbs that express only proper states such as "*contain*" or "*know*". Such verbs do not show causal, and thus temporal, structure in contrast to eventive verbs such as "*open*". This difference seems to be universal: eventive verbs denote a change of participants (change of their states) or transmission of force or sequences of changes, while stative verbs describe only properties or states of participants. Kratzer (1995) insists on a clear distinction between states and events with the help of individual-level predicates and state-level predicates. While Kratzer's individual-level predicates (e.g., states) express permanent properties, the state-level predicates (davidsonian events) represent temporary or accidental properties. The distinction between state-level predicates and individual-level predicates is explained by the presence (in state-level predicates) or absence (in individual-level predicates) of the davidsonian event argument.

As is the case for temporal information, states hold true for an indefinite period of time and, because they have no internal causal structure, they are true at any given interval of a discourse and at its sub-intervals (cf. Maienborn (2011) on the property of *temporal homogeneity*). In contrast to states, events lack the property of temporal homogeneity as changes may happen at the initial, final or any other part of an event depending on its causal chain. For example, if "*John reached the top of the mountain in three hours*", he was not on the top of the mountain before three hours had passed. Representation with the help of primitives and logical operators also reflects the differences between states and events. The representation of states does not involve the primitive predicates `CAUSE` and `BECOME`.

However, we may bear in mind that for every event there exist corresponding states. For our example "*John reached the top of the mountain in three hours*", the initial state was that "John was standing at the foot of the mountain" and the final state was "John was standing on the top of the mountain". That means that states and events are closely linked in our cognitive model. Croft (1990), for example, considers states as a view of the final segment of an event, which means that an event may precede the state described. Thus, states can be considered as components of events and the results of events as presented in the following example for the biomedical verb *"phosphorylate"*:

(3.8) $\texttt{BECOME}[\texttt{BE}[\texttt{Phosphorylated}(x)]]$

where $x \in Protein$ (set of all proteins) and `Phosphorylated` is a state.

Given the close connection between states and events, Bach (1986) introduced the term *eventuality*, which covers these categories.

The next question relevant for my work is about the status of facts. We can abstract an event to a *fact* such as "the reaching of a top by John in three hours" from the event "*John reached the top of the mountain in three hours*". In philosophy, events are distinguished from facts as they are more fine-grained than abstract facts, e.g., events contain temporal information. However, in general for every event there is a companion fact (cf. Bennett (1996)). Facts are considered as states of affairs and correspond to true propositions. Two sentences express the same fact (the same proposition) if and only if they are interderivable (Bennett, 1988). Imperfect nominals (such as "*reaching*" from the previous example) name facts.

This close connection between states, facts and events, captured in the Bach's *eventuality* (states and events) and Bennett's conception of facts, is important for the domain of molecular biology considered in my work and will be demonstrated later (Section 3.5).

3.4 Concepts of Event in Computational Linguistics

Linguistic theories focus in particular on causal and temporal structures of events, on the development of rich or abstract semantic role sets, and on status of state and fact in relation to events. The emphasis of linguistic work is mainly on verb classification according to a research focus and on comprehensive representation of event sentences with the help of primitive predicates. Theoretical linguistics has influenced a range of projects in computational linguistics which deal with the representation of events, their annotation and automatic extraction from text. However, some approaches to event analysis in computational linguistics have another scope and are defined given various applications scenarios such as information retrieval (e.g., Section 3.4.1). In the next sections I will introduce the most prominent projects on events from computational linguistics, and classify them from my point of view. I define six concepts of events in computational linguistics, e.g., "Event as a Document Cluster" (Section 3.4.1), "Event as a Template with Undefined Anchor" (Section 3.4.2), "Event as a Template with Lexical Anchor" (Section 3.4.3), "Event as a Situation Frame" (Section 3.4.4), "Event as a Verbal Predicate-Argument Structure" (Section 3.4.5), and "Event as a Situation Entity" (Section 3.4.6).

3.4.1 Event as a Document Cluster

In the Topic Detection and Tracking (TDT) competition, which represents information retrieval-driven event extraction, an event is defined as "some unique thing that happens at some point in time"(Allan et al., 1998). As an example we can think about the "Oklahoma City Bombing" or "the eruption of Mount Pinatubo on June 15th, 1991" as event occurrences, whereas `City-Bombing` or `Volcanic-Eruption` are considered to be event classes or *topics*. A TDT event is considered to be represented by a set of documents that discuss this event. The TDT challenge initiative investigates an automatic extraction of new events from broadcast news stories. The TDT contains two major tasks. The first task is to identify news stories that are the first to introduce and discuss a new event, and the second is to find the subsequent stories about this event in the broadcast news stream. Consequently, the documents of the TDT corpus are flagged for each of the pre-defined target events with `Yes`, `No` and `Brief` (briefly) tags. From a computational point of view, the event is defined as a cluster or set of broadcast news documents. Thus, predicate-argument structure or any temporal information do not play any role here.

Table 3.2: Template for the `Transfer-Money` event from ACE event annotation guidelines. Entity types are PER (Person), ORG (Organization), GPE (Geopolitical entity), MONEY (Money), TIME (Time), LOC (Location), FAC (Facility).

Argument Role	Argument Types	Role Description
`Giver-Arg`	PER, ORG, GPE	The donating agent.
`Recipient-Arg`	PER, ORG, GPE	The recipient agent.
`Beneficiary-Arg`	PER, ORG, GPE	The agent that benefits from the transfer.
`Money-Arg`	MONEY	The amount given, donated or loaned.
`Time-Arg`	TIME	When the amount is transferred.
`Place-Arg`	GPE, LOC, FAC	Where the transaction takes place.

3.4.2 Event as a Template with Undefined Anchor

Message Understanding Conference (MUC) competitions organized the Scenario Template Task (STT) introduced already in Section 2.2.2. STT introduces an abstraction from concrete events as analyzed in the TDT challenge and provides event class templates (with a fixed arity) that represent the domain of the texts analyzed. From a computational point of view, STT templates are defined in the form of relations between event participants, times and locations (Grishman and Sundheim, 1996). In contrast to the TDT challenge, STT introduces fixed semantic templates for multiple events. The selected event classes are created for application domains such as news articles on management changes (MUC-6), and satellite launch reports (MUC-7). An example of an event type is a `Negotiation` template annotated in MUC-6, which usually contains argument slots for `Party`, `Issue`, `Proposal-Status`, and `Talk-Status`. Such a complex MUC event can be represented over a range of sentences.

3.4.3 Event as a Template with Lexical Anchor

The third approach to handling events in computational linguistics was undertaken under the auspices of the ACE program (cf. Section 2.2.2). An ACE event is defined as “a specific occurrence involving participants, [...] something that happens, [...] can frequently be described as a change of state.” (ACE-Event-Annotation-Guidelines (2005), p. 5). In a way similar to MUC, ACE provides predefined templates for event classes such as `Life`, `Transaction`, `Business`, `Conflict`, `Creation`, `Movement`, and `Contact` with subtypes yielding a total of 33 event types (Doddington et al., 2004).

In contrast to MUC event templates, ACE event classes are not heavily dependent on the sublanguage of texts. Furthermore, ACE attempts to provide more abstraction to the set of semantic roles. Argument roles include `Person-Arg`, `Agent-Arg`, `Victim-Arg`, `Instrument-Arg`, `Vehicle-Arg`, `Destination-Arg`. Table 3.2 shows the template for the `Transfer-Money` event. This event has six slots for semantic arguments such as e.g., `Giver-Arg`, `Money-Arg` and `Place-Arg` slots which can be filled by named entities with allowed argument types (see the second column in Table 3.2).

Another striking difference between ACE and MUC events is that the ACE event extraction task requires annotation of an event mention in text in the form of *event trigger* within an *event extent*. An event extent in ACE is a sentence that contains a taggable event. An event trigger is the word that expresses the event occurrence (cf. ACE-Event-Annotation-Guidelines (2005)). Events can be triggered by verbs, nouns and adjectives. "*Jane Bobert Bond was **born** in England.*", "*He calculated that Jesus' **birth** had occurred 532 years earlier*" and "*[..] a Saudi-**born** dissident Osama bin Laden [...]*" are examples of ACE annotations of `Be-born` event with verbal trigger "*born*" in the first example, noun trigger "*birth*" in the second example, and an adjective trigger "*born*" in the last example.

3.4.4 Event as a Situation Frame

Another initiative for coding and annotating realizations of events is the FrameNet project (e.g., Baker et al. (2003)). FrameNet has a lexicographic character. Its objective is to provide a schematic representation of situations involving participants, which are frame elements. FrameNet allows nouns and adjectives to be lexical units representing situations. The selection of semantic roles in FrameNet is based on a conceptual role set for semantic frames (Fillmore and Atkins, 1992). FrameNet methodology has the following frame creation steps:

1. Select a semantic frame (for example `Commerce`),
2. define conceptual roles of this frame (`Buyer`, `Seller`, `Goods`, `Money`),
3. collect lexical predicates which would refer to the frame (e.g., "*sell*", "*buy*", "*purchase*").

In addition to semantic frame, FrameNet illustrates syntactic realizations with some examples (averaging more than 20 examples per frame, cf. Baker et al. (2003)).

Lexical predicates, which refer to the frame, and arguments, which refer to frame elements, are annotated. FrameNet currently contains 1,020 semantic frames and has a lexical database of 11,830 lexical units (a pair of lemma and a semantic frame).[2]

The FrameNet annotation, for example, of "*buy*" and "*sell*" lexical units in the `Commerce` frame has the following form:

(3.9) [`Buyer` *John*] *bought* [`Goods` *a car*] [`Seller` *from Mary*] [`Payment` *for $5000*].

(3.10) [`Seller` *Mary*] *sold* [`Goods` *a car*] [`Buyer` *to John*] [`Payment` *for $5000*].

Given the frame `Commerce`, FrameNet assigns roles to verbs "*buy*" and "*sell*" according to this semantic frame, e.g., `Seller` and `Buyer`. Thus, semantic roles remain the same in various realization of the same frame, e.g., `Commerce`, that might be useful for semantic applications.

3.4.5 Event as a Verbal Predicate-Argument Structure

While FrameNet has an illustrative character (only 78 full text documents annotated),[3] other projects aim to provide a large set of annotated data useful for statistical tools. The emphasis is to capture many different syntactic realizations of event structures. This is motivated by the fact that on the one hand, natural language can offer different syntactic realizations of the same event structure. For example, an argument "*window*" in different syntactic realizations "*John broke the window.*" (active voice sentence) and "*The window was broken by John.*" (passive voice sentence) takes the same event participant role (`Patient`). On the other hand, humans can use several lexical items to refer to the same event type. Here, "*smash*" could be used instead of "*break,*" lending its own individual semantic nuances. These two characteristics of verbal predicate realization are considered in Levin's verb classification (Levin, 1993).

Levin's verb classification is based on the idea that verbs occur in pairs of syntactic frames that are meaning preserving (Levin's diathesis alternations). Levin's main assumption is that syntactic behavior of a verb in the form of syntactic frames is a direct reflection of underlying semantic frames that control the surface realization of verb arguments. Alternative syntactic realizations of semantic arguments are a frequent phenomenon, affecting most English verbs. The "*break*" examples above represent transitive/intransitive alternation, more precisely "causative/inchoative

[2] This data has been extracted from the FrameNet 1.5 version (December 2011) download data using shell scripts in the *lu* and *frame* directories.

[3] FrameNet 1.5 version (December 2011), the download data.

alternation" (Levin, 1993). "*Break*", "*shatter*" or "*smash*" would be grouped together as they are able to undergo this alternation, and they also share a semantic component of "breaking an object with the resulting change of state of the object as broken in pieces".

Levin's verb classification inspired at least two large projects capturing semantic and syntactic verb classification, VerbNet (Kipper et al., 2000) and PropBank (Palmer et al., 2005). VerbNet was created as a hierarchical lexical verb resource. VerbNet is based on Levin's classification presented above and extends Levin's classes by creating correspondences between syntactic realizations, selectional restrictions, and semantic roles of arguments. Here is the representation of the verb "*break*" in VerbNet:

```
break-45.1
Members: 23, Frames: 10

Members
* Break, Cleave, Crack, [...]

Roles

* Agent [+int_control]
* Patient [+solid]
* Instrument [+solid]

Frames

NP V NP
   example "Tony broke the window."
   syntax  Agent V Patient
   semantics  cause(Agent, E) contact(during(E), ?Instrument, Patient)
   degradation_material_integrity(result(E), Patient)
   physical_form(result(E), Form, Patient)

NP.patient V
   example  "The window broke."
   syntax  Patient V
   semantics  degradation_material_integrity(result(E), Patient)
   physical_form(result(E), Form, Patient)

[...]
```

Table 3.3: PropBank arguments for the verbs "*buy*" and "*sell*", and the corresponding metaframe.

Argument Role	*buy*	*sell*	Exchange of Commodities for Cash
`Arg0`	**buyer**	**seller**	one exchanger
`Arg1`	thing bought	thing sold	commodity
`Arg2`	**seller**	**buyer**	other exchanger
`Arg3`	price paid	price paid	cash, price
`Arg4`		benefactive	benefactive

We see in the *break* entry the representation of semantic roles of `Agent`, `Patient`, and `Instrument`, and syntactic verb properties in the form of frames, such as `NP V NP` or `NP.Patient V`.[4]

The PropBank project, like VerbNet, is inspired by Levin's idea of linking syntactic realizations and semantic roles. PropBank is a 300,000-word corpus based on Penn Treebank (Marcus et al., 1994). PropBank was created by adding a layer of semantic annotation to the Penn Treebank syntactic annotations in the form of predicate-argument relations. PropBank itself does not generalize about verbs in the form of classes (as done in VerbNet), nor does it formalize the semantics of the roles (as done in VerbNet and FrameNet). In contrast to the projects presented above, PropBank prefers atheoretical semantic roles numbered sequentially from `Arg1` to `Arg5`.[5] The objective of PropBank annotation is to provide a large amount of data labeled with predicate-argument structures that could be a basis for learning a statistical model for automatic extraction of predicate-argument structures. The difficulty of defining a universal set of semantic roles for such a large annotation project is the reason for restricting the role set to numbered roles. However, `Arg0` is considered to be a prototypical `Agent`, while `Arg1` is a prototypical `Patient` according to Dowty (1991) (cf. Section 3.3.2). In the PropBank annotation guidelines, different verb senses are represented by different frame sets, i.e., semantic roles (role sets) and their associated syntactic realizations. These frame sets are used by annotators for a more consistent and reliable annotation process. An example of the frame set for the verbs "*buy*" and "*sell*" is presented in Table 3.3. The annotations of these frame sets are presented below:

[4] In addition, VerbNet introduces a representation of an event associated with a verb class. An event is decomposed into a tripartite structure which represents the states of an event, i.e., the preparatory (during(E)), culmination(end(E)) and consequent (result(E)). This representation

(3.11) [`Arg0` *John*] *bought* [`Arg1` *a car*] [`Arg2` *from Mary*] [`Arg3` *for $5000*].

(3.12) [`Arg0` *Mary*] *sold* [`Arg1` *a car*] [`Arg2` *to John*] [`Arg3` *for $5000*].

The argument representation of "*buy*" and "*sell*" verbs demonstrates that, in order to link both activity descriptions, we need, in contrast to the FrameNet approach, additional rules for mapping buyer and seller. This mapping effects in the form of PropBank meta-frames such as "Exchange of Commodities for Cash" (see third column of Table 3.3).

VerbNet and PropBank are good illustrations of how the linguistic theories of event can influence the work on events in computational linguistics in the form of large-scale verb classification and annotation of semantic structures. While the projects such as FrameNet, PropBank and VerbNet are preoccupied with the causal event structure and comprehensive definition and annotation of event arguments and their semantic roles, another projects focus on the temporal event structure and distinction between e.g., events and states. The latter are presented in the following section.

3.4.6 Event as a Situation Entity

There are a number of approaches that adopt and even extend the Vendler's classification of event types, based on the internal temporal structure of events (Section 3.3.3). For example, Siegel and McKeown (1996) annotate and automatically assign Vendler's situation types (*state*, *accomplishment*, *achievement*, and *activity*) to verbs. Other works go beyond Vendler's linguistic conception of four event classes. Palmer et al. (2007) annotate a corpus with nine situation types such as e.g., *event*, *state*, *report*, *fact*, and create statistical models for assigning such situation types. Another interesting and prominent project which deals with an extended Vendler's classification of events is the TimeBank project (Pustejovsky et al., 2003). TimeBank is a corpus annotated with TimeML, an expressive markup language for annotating time and event expressions to capture temporal structures in text. The language TimeML has been developed in order to mark up temporal information in text and be used in particular in the context of temporally sensitive question answering systems. In the TimeBank, events can be expressed not only by finite and infinite verbs, and verb nominalizations but also by nouns, adjectives, and even prepositional phrases. Here are some examples of events from TimeBank:

(3.13) "*John* ***teaches*** *on Monday.*"

is unique in the domain of computational linguistics.

[5] PropBank argument roles numbered higher than `Arg5` are assigned on per-verb basis.

(3.14) "*In July 1994, Ukraine again held free and fair* ***elections***."

(3.15) "*While* ***in office****, Kravchuk was always an advocate for [...] .*"

TimeML adopts and extends Vendler's classification of event types, based on the internal temporal structure of events. It captures seven different types of events: *Reporting*, *Perception*, *Aspectual*, *Intentional_Action*, *Intentional_State*, *State*, and *Occurrence*. TimeML allows *anchoring* of these seven event types in time, and *ordering* of events with respect to one another in time (before, after, during).

The research projects on temporal event structure are focused on anchoring events in time and on establishing temporal relations between events. In contrary to the projects on causal event structure (e.g., Section 3.4.5), this research field does not elaborate on predicate-argument structures and semantic role sets. The annotations are based only on consideration of an event instance as a situation entity which has temporal relations to other situation entities in text.

3.5 Adoption of Concepts of Event in Biomedical NLP

Concepts of events in computational linguistics have matured early before biomedical NLP was established as an important field of computational linguistics. Thus, biomedical NLP can find inspiration and insights in the previous work on events. Actually, Bio-PropBanks (Tsai et al., 2007) and Bio-FrameNets (Dolbey, 2009) appear as counterparts to PropBank and FrameNet from the general language NLP. However, the experience on events gained in the newspaper and ordinary English domain cannot be transferred to the biomedical domain without any adaptation. The intricacies of biomedical language and descriptions of biomedical events should be taken into account if working on the extraction of molecular events from literature. This will be discussed below.

According to Gene Ontology (GO)[6], the major ontology used for molecular biology research, the biological process is defined as "any process specifically pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms. A process is a collection of molecular events with a defined beginning and end." Figure 3.1 illustrates such a biological process. It shows a hypothetical signal transduction pathway inside a cell. The signal is mediated by various proteins to the nucleus of the cell using various events, such as protein-protein interactions and phosphorylation, and initiates transcription of a gene. Further, the process at the bottom of the figure shows how the transcription of a gene results in a protein product through

[6] http://www.geneontology.org/

splicing, translation and synthesis of the gene. This protein product then inhibits receptor signaling and thus regulates its own expression levels.

In molecular biology, a crucial research field in the biosciences, the bio-molecules[7] are the key players in molecular event descriptions spread over the life science literature. Molecular events describe observable changes of bio-molecules, such as binding of proteins or RNA production. The GENIA event annotation project (Kim et al., 2008a), for example, defines a molecular event as "a change of the biological state, properties or location of a bio-molecule".[8] Molecular events can be sub-divided into a set of (nested) events. For example, the regulation of gene expression involves at least two events, i.e., binding of a transcription factor to a promoter and expression of a protein for a corresponding gene or operon. In parallel to molecular formations, these molecular events influence the formation of a phenotype (organism's observable characteristics), which may, itself, be responsible for drug reactions or development of certain diseases.

The goal of biomedical IE is to get detailed views on such a behavior of bio-molecules in the form of their inter-play in molecular events described in text. In texts, biologists describe such molecular processes using predicates such as "*mediate*", "*phosphorylate*", and "*regulate*" (look for predicates in Figure 3.1). Direct extraction from text of such a complex network, as that presented in Figure 3.1 (entities are marked in yellow, green, and blue, pathways are marked with green arrows), is challenging. To get acquainted with the textual appearance of complex molecular events, a student of biology and I manually analyzed 50 sentences randomly extracted from MEDLINE abstracts in order to find evidences of how molecular events are expressed literally in documents. We narrowed our view on gene expression regulation events (`Regulation Of Gene Expression` (ROGE) event). The gene expression regulation can be described as the process that modulates the frequency, rate or extent of gene expression, where gene expression is the process in which the coding sequence of a gene is converted into a mature gene product or products, namely proteins or RNA (taken from the definition of the GO class `Regulation of Gene Expression`, GO:0010468). Transcription factors are proteins that play a central role in the regulation of gene expression, they can bind to the DNA and activate or inhibit a gene expression process. I will first present some examples of how gene expression regulation events are presented in text. In the following list, I rank examples that stand for a certain pattern (cf. Buyko et al. (2008)) by their frequency in the set (in descending order). Event arguments are marked in bold.

[7] Bio-molecules are "molecules that naturally occur in living organisms, e.g., proteins, DNAs, RNAs, cells, etc., or their equivalents which are prepared for experimental purposes, e.g., cultured cells, specially treated proteins, etc." (taken from the GENIA event project (Kim et al., 2008a)).

[8] This definition is widely accepted in the biomedical NLP community.

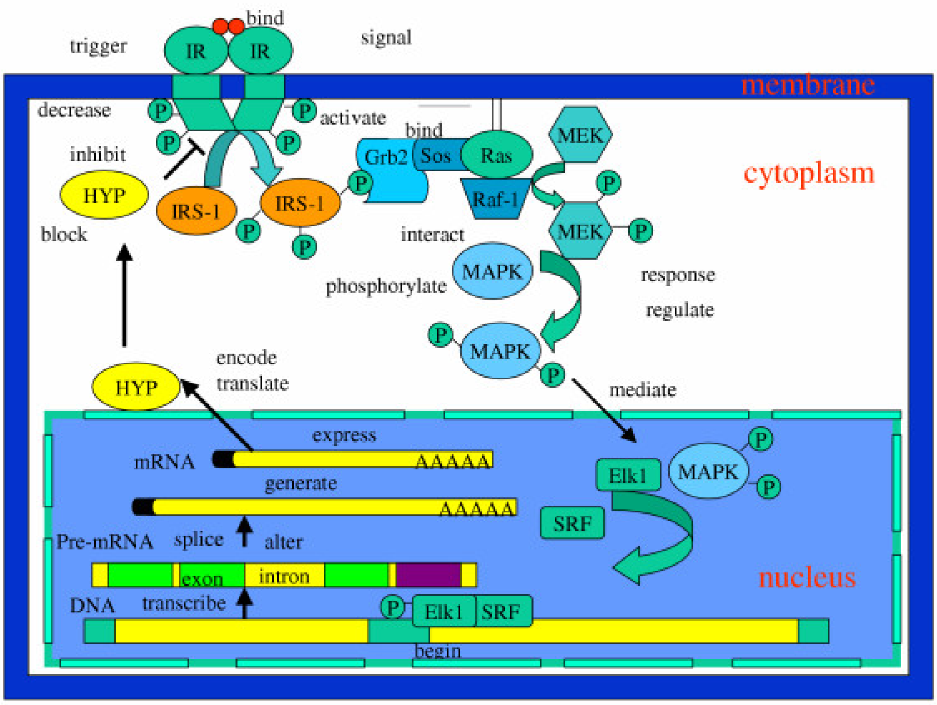


Figure 3.1: A biologist's view of processes in a cell. This figure is taken from Wattarujeekrit et al. (2004).

1. “***IclR*** *also represses the expression of* ***iclR***.”
2. “***yeiL*** *expression is positively activated by* ***Lrp***.”
3. “***SlyA***-*induced* ***proteins***.”
4. “***IclR*** *is a repressor for the* ***Escherichia coli aceBAK operon***.”
5. “*Elevation of* ***ppGpp*** *levels in growing cells ... triggered the induction of all* ***usp genes***.”
6. “***ZntR*** *is a trans-acting repressor protein that binds to the* ***znt*** *promoter region.*”
7. “***rpoS*** *function is essential for* ***bgl*** *silencing.*”
8. “*Transcription repression of* ***the Escherichia coli acetate operon*** *by* ***IclR***.”

9. "*Expression of the* ***tau and ssu genes*** *requires the LysR-type transcriptional regulatory proteins* ***CysB and Cbl***."
10. "*The promoters of* ***the mar/sox/rob regulon of Escherichia coli*** *contain a binding site (marbox) for the homologous transcriptional activators* ***MarA, SoxS and Rob***."
11. "***bgl*** *silencing caused by C-terminally truncated* ***H-NS***."
12. "*Disruption of* ***cueR*** *caused loss of* ***copA*** *expression*."
13. "*Synthesis of* ***Cbl*** *itself is under control of* ***the CysB protein***."

In each of these examples there are different descriptions of ROGE events, such as clear mention (with the help of regulatory verbs) of positive regulation or negative regulation caused by transcription factors, e.g., in Examples (1), (2), and (3); descriptions of the roles of transcription factors in regulating gene expression as in Examples (4) and (7); mention of molecular events which are part of ROGE processes, i.e., a binding event in Example (6); or even descriptions of the properties of the regulated genes as in Example (10). The culmination of the variety is the description of causal relations between the expression events as in Examples (12) and (13). Given the variety discovered by the manual analysis of life science documents, three major characteristics can be identified in the conception and description of molecular events. First, the close connection between events, states and facts, second, descriptions of event parts from which the whole molecular process could (easily) be inferred, and third, the complex descriptions of nested events with usually causal relations. In the following sections I present these main issues (Sections 3.5.1, 3.5.2 and 3.5.3).

3.5.1 Event, States and Facts, or do Biologists Care for this Distinction?

In Section 3.3 the concepts *event*, *state* and *fact* figured out from a linguistic perspective. The main conclusion I could draw from this previous work is that states are integrated in events as proposed, for example, by Croft (1990) for the stative view on events. Another concept that is closely linked to *event* is *fact* so we could conclude that for every event there is a companion fact (cf. Section 3.3.4).

Thus, the consideration of an *event* as particular with causal and temporal structure is a very narrow concept of an event. In analyzing the examples of ROGE molecular events from the linguistic point of view, I was able to find proper events, stative views on events, proper states and facts. All these examples have been classified by a graduate student of biology as *pari passu* ROGE descriptions.[9]

Here are examples of each category:

(3.16) Event proper

> "***IclR** also represses the expression of **iclR**.*"

(3.17) State as a result, stative view on an event

> "***SlyA**-induced **proteins**.*"

(3.18) State proper

> "*The promoters of **the mar/sox/rob regulon of Escherichia coli** contain a binding site (marbox) for the homologous transcriptional activators **MarA, SoxS and Rob**.*"

(3.19) Fact

> "***bgl** silencing caused by C-terminally truncated **H-NS**.*"

In classifying the examples, textual representation can be seen to be distributed into four categories. Event proper descriptions occur six times and concern Examples (1), (2), (5), (8) and (12). State as a result is present in Example (3). State proper fits only Example (10). Facts are represented in Examples (4), (6), (7), (9), (11) and (13). In this work, facts are considered to be expressed not only by imperfect nominals but also as descriptions of roles of participating entities in molecular events. These statements are characterized by features of abstractness. Thus, events, (proper) states and facts identified from events are relevant for capturing the descriptions of gene expression regulation events. However, the proper states are considered in my work outside of the event group. Example (10) provides only a description of properties of a regulated gene. Given these properties the biologist can only conclude that an event may happen.

My first conclusion from this study in cooperation with a graduate student of biology is that biologists do not care for the strong distinction between events, states and facts if they consider the literature for gene expression regulation events. Restricting the consideration of events only to events proper (from the linguistic perspective) would lead to an immense loss of relevant event information from the biologist's field of vision (more than 60% of data in the example set presented). Therefore, I subsume under the term *(molecular) event* in this work the events proper, stative views on events (in the sense of Croft (1990)) and companion facts about events

[9] This conclusion appear from my internal discussion with graduate students of biology.

(Bennett (1996)), and subscribe in this approach to the eventuality idea of Bach (1986), which is extended in this work by the concept *fact*. The eventuality is a more appropriate concept for capturing molecular events in biomedical literature.

3.5.2 Macro and Micro Views on Molecular Events

The views of a biologist on molecular events give a direction for my information extraction work. If the mentioning of molecular events in documents are considered together with biologists, it quickly becomes clear that all biological processes can be sub-divided into a set of molecular processes which are nested and inter-connected. For example, regulation of gene expression involves many sub-processes, e.g., binding a transcription factor to a promoter, activation of a promoter of a corresponding gene or even operon for gene transcription, transcription of DNA snippets into RNA structures, and translation of RNA structures into proteins.

Given this biological picture, it can be seen how complex the only molecular event we consider here “regulation of gene expression” actually is. It is evident that, given its high complexity, descriptions of this event can differ. The descriptions may apply to the whole process “from a bird‘s eye view” (a macro view) or to the sub-processes only (a micro view), and even to the super-processes at the phenotype level which indicate the influence of particular gene expression regulation. The super-processes at the phenotype level such as drug reaction are not considered in this work and remain a challenging issue for the future.

In order to illustrate different views on molecular events, I represent a gene expression regulation event as a bubble that integrates a range of other molecular events such as `Transcription` and `Binding` (see ROGE in Figure 3.2). This figure shows that the description of ROGE might provide only partial views of ROGE which indicate that ROGE events happen as in the following example (the previous Example (6)):

(3.20) “***ZntR** is a trans-acting repressor protein that **binds** to the **znt** promoter region.*”

This sentence contains a description of a binding process of a transcription factor to a promoter, which is a crucial sub-event of a ROGE event. Thus, obtaining a complete view of one molecular event means knowing all the other molecular events which are part of it and all events which might cause the actual event.

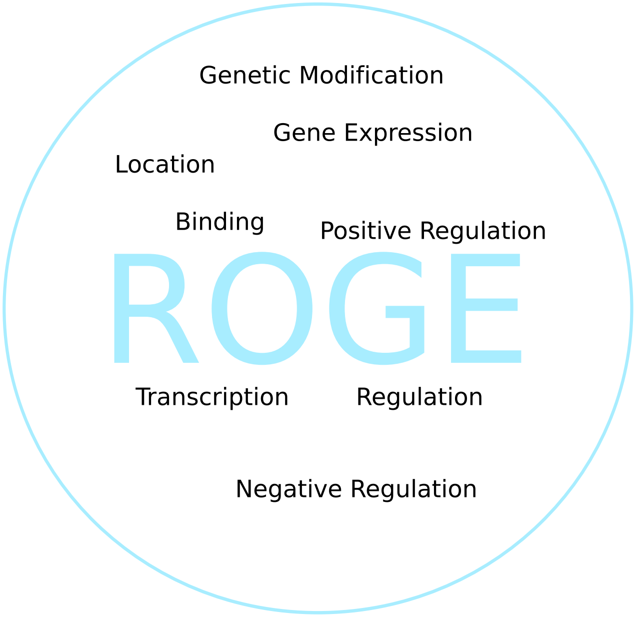


Figure 3.2: Regulation of Gene Expression (ROGE) event as a core event consisting of a range of sub-events.

3.5.3 Nesting of Events

Nesting of molecular events reveals the complexity of molecular interactions in organisms. The occurrences of molecular events are closely inter-connected and can be described in so called pathways. Frequently, molecular events are connected by causal relationships, that means that one event causes another event. This can be illustrated with the help of the following example (previous Example (12)):

(3.21) "***Disruption*** *of cueR* ***caused loss*** *of copA expression.*"

In this example the negative regulation of *cueR* leads to a decreased expression of *copA*, that means that *cueR* plays an important role in the regulation of the expression of the *copA* gene, and normally has a positive regulatory effect on this gene. How can we conclude it? This can be explained in the workflows of wet lab research studies. Experimental environments for molecular event detection often involve modifications of genetic material. By means of these genetic modifications and the expression levels of other genes, researchers explicitly draw conclusions about the role of these genes in the gene regulation processes. The following sentence is a good illustration of such experimental conclusions:

(3.22) "*Transcription of the chromosomal asr was* ***abolished*** *in the presence of a phoB-phoR* ***deletion mutant****.*"

This sentence describes a negative regulation of *asr* that enters into force only if the *phoB-phoR* genes are deleted from the cell and thus are not expressed, that means that an artificially negative regulation of these genes is initiated and causes a negative regulation of the *asr* gene. Finally, reading this sentence, other researchers can conclude that the *phoB-phoR* genes in a normal cell have a positive regulatory effect on the *asr* gene. For non-biologists this knowledge seems to be read between the lines, while for biologists this is a knowledge (easily) inferred from stated experiments.

In summary, the textual appearance of molecular events hides challenges for their modeling and automatic extraction. IE templates for molecular events such as ROGE can be filled with nestings of micro and macro events, states and even facts. Therefore, the question arises "How to deal with the complexity of molecular event descriptions in text?". This will be approached in next sections.

3.5.4 Biomedical Event Predicates in General Language Resources

The first research question for capturing automatically molecular events in text is about the nature of event predicates used. I consider in the following study a collection of event predicates (so called event *triggers*) automatically extracted from two representative molecular event corpora, i.e., the GENIA event corpus (Kim et al., 2008a) and the GENEREG corpus (Buyko et al., 2008) (Sections 3.6.2 and Appendix Section A.1). My aim is to show the distribution of event predicates in available large-scale lexical resources and corpora from the general language domain, e.g., event resources presented in Section 3.4. These include VERBNET, FRAMENET, and PROPBANK (cf. Sections 3.4.4, 3.4.5). Further representative resources I integrated in this study are WORDNET (Fellbaum, 1998) and NOMBANK (Meyers et al., 2004).

The predicates of the following representative molecular event types have been extracted for this study, e.g., `Transcription`, `Gene Expression`, `Regulation`, `Positive Regulation` and `Negative Regulation` events (see Appendix, Section A.2.2 for definitions). The most frequent predicates (cf. Appendix, Section A.2.1) have been manually analysed for POS tag distribution (Table 3.4), and have been manually linked to the lexical resources introduced above (Table 3.5).

The distributions of POS tags differ for various molecular event types. For example, `Transcription` and `Gene Expression` events are frequently expressed with the help of nouns (verb nominalizations) such as "*transcription*" and "*expression*". For the regulatory events such as `Regulation`, the use of verbs prevail the use of nouns. Interestingly, `Negative Regulation` events are expressed almost equally with the help of verbs (15) and nouns (13). For expressing `Positive Regulation` events, biologists apply, in addition to verbs and nouns, adjectives and adverbs. For example an adjective "*inducible*" or "*essential*" may refer to a `Positive Regulation` event.

Table 3.4: Statistics on part-of-speech categories of event triggers for each event category.

Event (lemma)	Verbs	Nouns	Other
`Transcription` (14)	3	10	1
`Gene Expression` (17)	5	11	1
`Regulation` (21)	10	6	5
`Positive Regulation` (43)	19	13	11
`Negative Regulation` (29)	15	13	1
`TOTAL` (124)	52	53	19

Table 3.5: Number of event trigger words matching general language resources for each event category.

Event	NomBank	WordNet	FrameNet	VerbNet	Total
`Transcription`	6 (42%)	7 (50%)	2 (14%)	1 (7%)	7 (50%)
`Gene Expression`	2 (11%)	10 (59%)	6 (35%)	2 (11%)	10 (59%)
`Regulation`	4 (19%)	18 (86%)	13 (62%)	8 (38%)	18 (86%)
`Positive Regulation`	5(11%)	31 (72%)	22 (51%)	12 (28%)	32 (74%)
`Negative Regulation`	5 (17%)	19 (66%)	20 (69%)	9 (31%)	19 (66%)
`TOTAL`	22 (17.7%)	85 (68.5%)	63 (50.1%)	32 (25.8%)	86 (69.3%)

Table 3.5 represents the results of matching event triggers against the resources considered.[10] The resource with the highest number of matches (68.5%) is WordNet where I found between 50% (`Transcription`) to 86% (`Regulation`) of all event

[10] For many triggers, I could not find a corresponding lemma or its sense in the screened resources. Accordingly, in Table 3.5, I only counted the lemmas with correctly traceable and identified senses.

Table 3.6: Frames in FrameNet corresponding to selected event categories from the biomedical domain.

Event	Frames in FrameNet
Transcription	Causation (*induction*), Becoming aware (*detect*)
Gene Expression	Causation (*induction*), Becoming aware (*detect*), Creation (*produce*), Presence (*present*)
Regulation	Objective influence (*effect, affect, influence*), Control (*control*), Participation (*involve, involvement*), Cause change (*change, alter*), Contingency (*dependent*), Response (*response*).
Positive Regulation	Causation *(induce, lead, result, cause*), Cause change position on a scale (*increase, enhance, promoter*), Being necessary (*require, essential, necessary*), Contingency (*dependent*), Cause to start (*stimulate*), Amassing (*accumulation*), Relative time (*after*), Time vector (*through*), Importance (*important*), Extreme value (*high*), Being active (*active*)
Negative Regulation	Hindering (*inhibit*), Cause change position on a scale (*decrease, reduce, reduction, diminish*), Change position on a scale (*decline*), Preventing (*prevent*), Possession (*lack*)

triggers. This outcome supports the view that WordNet is the best-maintained and most commonly used lexical resource in NLP applications. WordNet is followed by FrameNet with 50.1% matches, and VerbNet/PropBank with 25.8% matches. At the bottom of the list appears NomBank with 17.7% matches. The most difficult to link is the Transcription event as it is expressed through compounds such as *"mrna levels", "transcriptional activity", "mrna expression"* which are rare or absent in general English language resources. Regulation and Positive Regulation triggers have the highest coverage in general language lexicon and corpora resources. These events are usually expressed by words that describe general regulation, influence or control.

I took a closer look on the FrameNet data detected for event triggers. Transcription and Gene Expression events share a set of frames, e.g., Causation and Becoming aware, that represent different view points on the production of proteins from a DNA sequence. One of these view points is of regulation by proteins and the other is that of a biologist doing experiments (Table 3.6). The sharing of frames can be explained by the fact that a transcription event is part of a gene expression event. `Regulation` events are expressed by frames such as `Objective influence`, `Causation`, and `Control` (Table 3.6). `Positive Regulation` and `Negative Regulation` correspond to frames that express more emphatic influence such as `Cause change position on a scale`, and `Hindering`. Nevertheless, many predicates could not be connected to FrameNet. The linkage ratio lies between 14% (for `Transcription`) to 69% (for `Negative Regulation`). Very specific biomedical words such as "*down-regulation*" or "*up-regulation*" are not represented at all in any of the lexical resources I explored.

This study provides clear evidence for the modest coverage of general language resources in relation to biomedical triggers that are relevant for the extraction of a range of representative molecular events. This work shows that there is a strong need for extension of general language resources for the domain of molecular biology or even creation of new resources. Some verb lists have already been compiled by individuals (e.g., by Fundel et al. (2007)), while the BioLexicon (Sasaki et al., 2008a) and the Specialist Lexicon[11] currently constitute the most comprehensive repositories of "biological" verbs. Furthermore, we can find in the biomedical NLP domain the counterparts of verb-focused projects from the newspaper domain for PropBank, e.g., PASBio, and BioSmile (Section 3.5.5), and the counterparts of frame-focused projects, e.g., the BioFrameNet project (Section 3.5.6). These projects are introduced in the following sections.

3.5.5 Biomedical PropBanks

The study presented in the previous section revealed that a range of molecular events are described with the help of verbs or their nominalizations. Indeed, verbs are graded in linguistic studies as the major word class for referring to events (Section 3.3). Therefore, they should deserve particular attention. This section introduces the biomedical counterparts of PropBank, i.e., the BioProp and the PASBio projects which focus on biomedical verbs.

In BioProp (Tsai et al., 2007), the biomedical propositional bank, 30 representative (according to their frequency) biomedical verbs, such as "*regulate*" and "*activate*" have been annotated in biomedical texts in a semi-automated way using a semantic role labeler trained on the PropBank. After that the results were corrected by human annotators with reported high Inter-Annotator Agreement rate of 0.95

[11] `http://www.nlm.nih.gov/pubs/factsheets/umlslex.html/`

kappa for semantic role identification and classification. As the predicate-argument structures of BioProp and the annotated corpus are not freely available, it is not possible to analyze this data critically. Therefore, this work focuses only on the PASBio project data.

PASBio (Wattarujeekrit et al., 2004), the first PropBank-oriented project in biomedicine, extends the PropBank frame sets to the domain of molecular biology. PASBio provides predicate argument structures (PAS) for 30 selected verbs (according to their frequency in the biomedical literature) from the year 2004 and is publicly available online.[12] PASBio authors consider that "the predicate-argument structure [...] would be a natural choice for IE, especially event extraction in molecular biology." (Wattarujeekrit et al. (2004), p. 9). PASBio chose predicates that describe molecular processes with gene and gene products as key participants. For example, gene expression or signal transduction are events which describe functions of genes and their products. The working scheme for PASBio is similar to the scheme of PropBank, i.e., select verbs, provide frame sets for verb senses, and annotate example sentences. The PASBio annotation corpus was assembled from Medline abstracts and full text journals. The corpus produced in PASBio is not as large as the PropBank corpus. PASBio is based more on lexical definitions and annotation has only been performed on 300 sentences for 30 predicates, which means ten sentences on average for each predicate. Thus the corpus is very small. PASBio events are mostly described in a sentence with the help of verbal predicates. Nevertheless, PASBio admits that the verb can be realized in its normal verbal form, as a participial modifier, or in its nominal form. For example, the verb describing `Down-regulation` "*down-regulate*" can be realized as participle modifier ("*down-regulating*") or as nominalizations ("*down-regulation*"). All arguments are introduced via semantic roles of a PAS frame of the selected predicate, but no attempt is made to provide consistent semantic roles for arguments numbered higher than `Arg0`.

This thesis considers that an important contribution of PASBio is the categorization of 30 predicates in four groups (cf. Wattarujeekrit et al. (2004)). These groups are defined as follows:

- Group A verbs have the same semantic sense as in the PropBank but require more arguments.
- Group B verbs have the same semantic sense as in the PropBank but require fewer arguments.

[12] `http://sites.google.com/site/nhcollier/projects/pasbio/`

- Group C verbs have the same semantic sense as in the PropBank and PAS frames are identical to the PropBank.
- Group D verbs have a different semantic sense in biomedical documents.

In the following, I present some examples from these four PASBio groups in order to illustrate "biological" verbs in action.

Group A contains nine verbs. For example, the verb "*mutate*" from group A describes changes of an entity. While in PropBank the verb "*mutate*" requires two arguments, `Arg0` (agent) and `Arg1` (entity undergoing mutation), the PASBio frame for "*mutate*" has the following form:

```
<predicate lemma="mutate">
  <roleset id="Mutate.01" name="" wordnet="1">
    <roles>
      <role n="1" descr="physical location where mutation happens"/>
      <role n="2" descr="mutated entity"/>
      <role n="3" descr="changes at molecular level"/>
      <role n="R" descr="changes at phenotype level"/>
    </roles>
  </roleset>
</predicate>
```

PASBio introduces three additional arguments for the verb "*mutate*", for "physical location where mutation happen", "changes at molecular level" and "changes at phenotype level".

An example sentence

(3.23) "*Groucho binding was, however, abolished by* ***mutating*** *a conserved phenylalanine of the eh1/GEH sequence to glutamic acid.*"

contains three arguments: `Arg1` is "*a conserved phenylalanine*", `Arg2` is "*the eh2/GEH sequence,*" and `Arg3` is "*to glutamic acid*". Thus, PASBio does not distinguish between processes and achievements and provides the scheme for eventuality type annotation.

Group B contains five verbs. The group B verb "*block*", for example, offers four semantic roles in PropBank, while in PASBio the verb requires only two core participants, the agent of the blocking process and the entity undergoing blocking as illustrated below:

```
<predicate lemma="block">
  <roleset id="block.01" name="" wordnet="3">
    <roles>
      <role n="0" descr="causer agent"/>
      <role n="1" descr="theme (process or entity being stopped)"/>
    </roles>
  </roleset>
</predicate>
```

An example sentence

(3.24) "*Both RAP1 and 2 are important vaccine candidates because it has been shown that antibodies to RAP1 are able to* ***block*** *merozoite invasion in vitro.*"

illustrates both arguments, `Arg0` "*antibodies to RAP1*" and `Arg1` "*merozoite invasion in vitro*".

Group C contains six verbs. The group C examples are verbs such as "*confer*" or "*lead*".

But the most interesting group is group D, which contains nine verbs. In this group the verbs have a different semantic sense from that in the general language domain, with "*express*", "*transcribe*", "*transform*" as typical examples. The PASBio frame for "*express*" looks like:

```
<predicate lemma="express">
<roleset id="express.01" name="" wordnet="5">
<roles>
  <role n="1" descr="named entity being expressed"/>
  <role n="2" descr="property of the existing named entity"/>
  <role n="3" descr="location referring to organelle, cell or tissue"/>
</roles>
</roleset>
</predicate>
```

The next example illustrates the use of the verb "*express*".

(3.25) "*Two equally abundant mRNAs for il8ra, 2.0 and 2.4 kilobases in length, are* ***expressed*** *in neutrophils and arise from using two alternative polyadenylation signals.*"

Here, three arguments are referred to, `Arg1` "*mRNAs for il8ra*", `Arg2` "*2.0 and 2.4 kilobases in length*" and `Arg3` "*neutrophils*".

The problems of the PASBio approach are of two kinds. First, PASBio argues for molecular event types to be unambiguously assigned to predicates and, second, for invariable expression of the molecular event in text using this predicate.[13] However, events can be expressed by different predicates and predicates may not refer to an event in text if arguments are lacking or "wrong" arguments are addressed. Wattarujeekrit et al. (2004) exemplify both issues in the following sentences describing an `Alternative Splicing` event (multiple transcripts generated from a single gene) (cf. Wattarujeekrit et al. (2004) p. 16-17). The following sentences exemplify these issues.

(3.26) *"Northern blot analysis with mRNA from eight different human tissues demonstrated that [the enzyme]$_{Arg1}$ was **expressed** exclusively [in brain]$_{Arg3}$, [with two mRNA isoforms of 2.4 and 4.0 kb.]$_{Arg2}$."*

(3.27) *"[A complementary DNA clone]$_{Arg1}$ encoding the large subunit of the essential mammalian pre-messenger RNA **splicing** component 2 snRNP auxiliary factor(U2Af65) has been isolated and **expressed** [in vitro]$_{Arg3}$."*

In Example (3.26) the sentence describes an alternative splicing event without referring to it by a predicate "*splice*" but by using the predicate "*express*", which usually refers to a gene expression event. In Example (3.27), although the predicate "*splice*" is used, there is no alternative splicing event description. The sentence in Example (3.27) talks only about expressing a single mRNA splicing factor, and `Arg2` is missing here. Thus, for example, Cohen and Hunter (2006) suggested giving PASBio the more desirable FrameNet-like structure which is not restricted to a single predicate. Furthermore, as PASBio lacks an adequately large corpus, any evaluation of this approach is of a speculative nature, and the number of PASBio verbs within available large corpora, such as Genia Treebank (Ohta et al., 2002) is too small (only 8.5% for PASBio verb tokens and 2.6% for PASBio verb types) (cf. Cohen and Hunter (2006)).

The major contribution of PASBio from my point of view was systematically to show that verb use in biomedical texts often differs from that in general language. PASBio demonstrates that only 23% of representative verbal predicates have an identical predicate-argument structure and semantic sense as in the general language domain. The molecular language contains a lot of domain-specific verbs. Furthermore, the description of molecular events in biology is complex because the argument content can change the event description specified by a predicate (cf. Examples (3.26)

[13] The PASBio team admits these problems (Wattarujeekrit et al., 2004).

and (3.27)). Overcoming these constraints, this thesis stresses similar in spirit to Cohen and Hunter (2006) a more desirable template-like representation for molecular events. I will describe it in the following sections.

3.5.6 Biomedical FrameNet

BioFrameNet (Dolbey, 2009) is conceived as an extension of FrameNet for the domain of molecular biology and currently provides two frames for `Protein-Transport` and `Cause-Protein-Transport`, and a number of annotations for both frames on sentence data to exemplify lexical predicates and their predicate-argument structures.

The frame `Protein-Transport` allocates four participants (Table 3.7).

Table 3.7: `Protein-Transport` frame from BioFrameNet, core frame participants.

`Transported entity`	Protein or protein complex that moves from one location in a cell to another location.
`Transport origin`	The location of the `Transported entity` before the motion event takes place.
`Transport destination`	The location of the `Transported entity` after the motion event takes place.
`Transport location`	The cellular component(s) mentioned in the movement of transported entities in cases where no specific origin or destination is indicated, or the location is both origin and destination in continuous, frequent motion events.

The following example shows an annotation sample of the `Protein-Transport` frame:

(3.28) "*inhibited **translocation** of the enzyme to the membrane*"

In this example, the predicate "*translocation*" invokes the `Protein-Transport` frame with participants "*enzyme*" as a `Transported entity` and "*membrane*" as a `Transport destination`. `Transport origin` and `Transport location` are not mentioned in this text. The set of all lexical units in the `Protein Transport` frame is about 32 items and contains words such as *"delivery", "migrate", "transport", "recycle"*, where 22 items are nominal lexical units while ten items are verbal lexical units. The BioFrameNet was a PhD project and exemplified only the extension of the FrameNet to the subject of protein transport and the linking of two new frames to selected biomedical ontologies such as GO.

The weak points in BioFrameNet and PASBio are e.g., a small size of annotated data and a small number of represented predicates (30 predicates in PASBio) or frames defined (two frames BioFrameNet) (that requires an amount of manual work). Furthermore, both projects do not take into account states and facts, a nesting of events, and potential partial views on molecular events in pathways. These issues, presented as special for the molecular biology in this thesis (cf. Sections 3.5.1, 3.5.2 and 3.5.3), require more comprehensive and more substantial definition and annotation of molecular events which should be applied for large-scale event extraction solutions. The aim of the next section is to present such an approach.

3.5.7 Biomedical Ontology-based Approach

The projects presented in the previous sections (Section 3.5.5 and 3.5.6) follow in the footsteps of the activities from the general language domain of computational linguistics where work on events mostly concerns modeling lexicon-like frames, as in FrameNet, or annotation efforts on argument structures for verbs, as in PropBank. The event annotation projects in the general language domain paid little attention to the interplay between developed lexical resources or annotated corpora and ontologies. This has changed during the last decade (cf. ACE and FrameNet activities). In molecular biology, domain ontologies play a crucial role in all knowledge-based applications from the very start. The ontology is a platform used by biologists to retrieve the knowledge from text and normalize according to ontological representation. In general, the ontological representation should help to abstract and to model the domain knowledge. The linking from text to ontological representation takes place either through the manual analysis by database curators or with the help of information extraction tools. Thus, it is preferable that the modeling and annotation of molecular events is properly linked to domain ontologies to be used by biologists.

In the light of this requirement, the Genia event annotation project and the "BioNLP 2009 Shared Task on Event Extraction" (which is based on the Genia event corpus subset) aim to provide links between the events in text and process classes in ontologies. The "BioNLP 2009 Shared Task on Event Extraction" project addresses nine events which are very representative for the molecular biology. `Localization`, `Binding`, `Regulation`, and its sub-types, e.g., `Positive Regulation` and `Negative Regulation`, are very general events, and, `Transcription`, `Gene Expression`, `Phosphorylation,` and `Protein Catabolism` are specific events describing protein production, modification and destruction. All event types are represented in the

Genia ontology (Figure 3.3), which is collected from GO, the major ontology for the biomedical domain[14] (see Appendix, Table A.7 for definitions).

In this context, the first question arises "How detailed and explicit is the modeling of biological processes in ontologies?". First, the representation of processes in biomedical ontologies usually has a textual definition. Second, the structure is given at the ontological hierarchy level, i.e., by classifying a concept in relation to other concepts in terms of Is-a relations. So, for example, starting from the common node Biological process, the classes Metabolism and Binding extend the branch of physiological processes (Physiological process) whereas the Cell recognition and Cell adhesion classes are defined as cellular processes as they extend the concept Cellular process (see Figure 3.3). The Part-of hierarchy, which is still in an early stage in many ontologies,[15] extends the expressiveness of molecular ontologies and allows representation of sub-processes, e.g., for example that RNA translation can be represented as a part of a gene expression process.

However, besides the Is-a and Part-of hierarchies of molecular processes, a further crucial part of an event description is the representation of its participants in form of argument types and roles as done in frame-oriented approaches (Section 3.5.6). Argument structures of molecular events are not represented in most ontologies, e.g., the Genia ontology.[16] However, ontology languages such as OWL 2 (Krötzsch et al., 2009) offer techniques for representation of such argument structures. Therefore, some ontologies already define biological processes in form of their argument structure, for example, the Gene Regulation Ontology (GRO) (Beisswanger et al., 2008). The latter provides the information that classes have the properties hasParticipant or hasAgent (relations introduced in Smith et al. (2005)). So, for example the class Activation[17] has the following definition in GRO:

```
Definition:
"Any process that activates or increases the frequency, rate or extent
of a biological process, function or phenomenon."

Equivalent classes:
"regulatory process"

Inherited anonymous classes:
```

[14] The following classes: Gene Expression, Regulation, Positive Regulation, and Negative Regulation are re-defined in the Genia ontology.

[15] Genia ontology does not provide a Part-of hierarchy.

[16] The Gene Ontology and the Genia ontology do not dispose about such definitions (at the moment of writing this thesis).

[17] The GRO class Activation corresponds to the Genia class Positive Regulation.

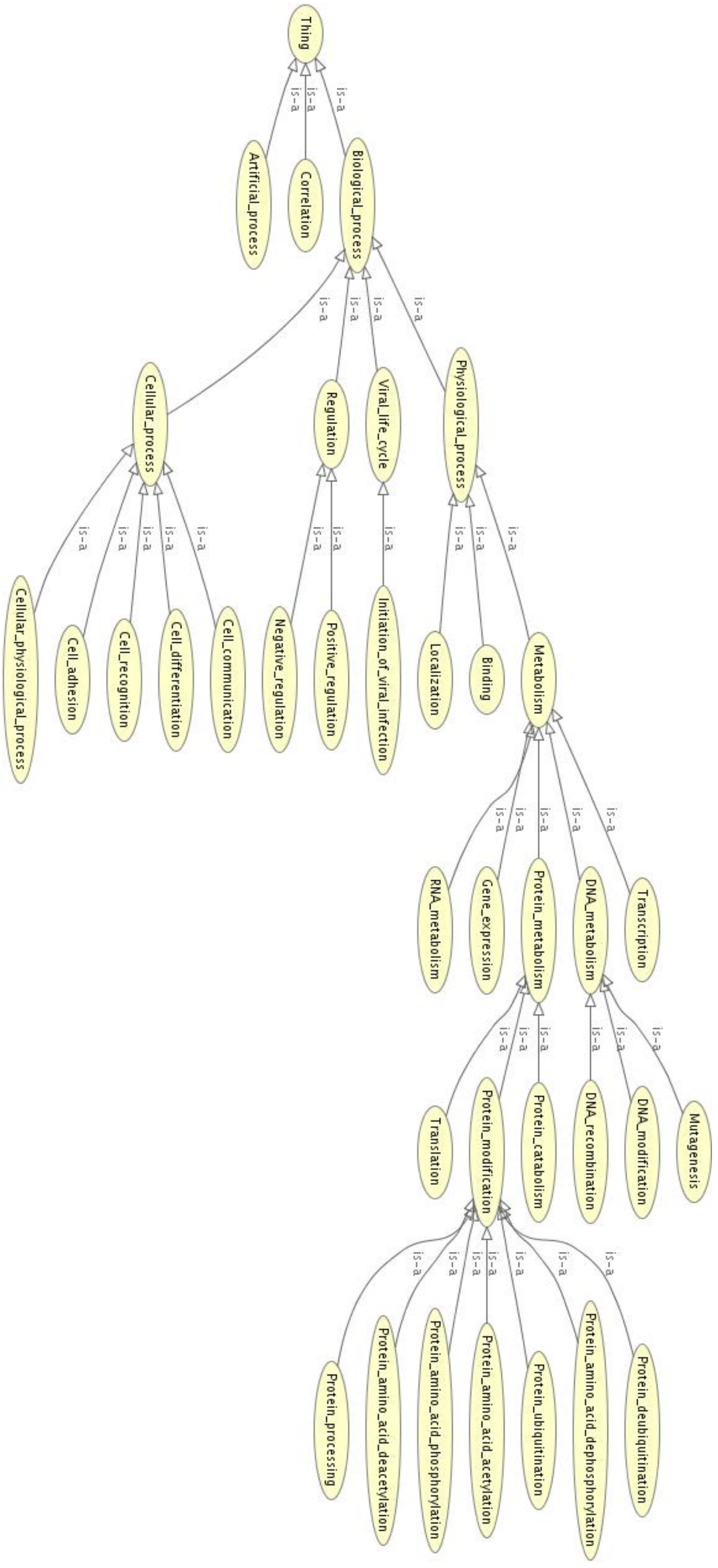


Figure 3.3: GENIA ontology for event annotations.

```
"hasParticipant min 1 Thing"
```

In order to reflect the argument structure, the GRO `Activation` class inherits from the anonymous class "`hasParticipant min 1 Thing`". As the GRO provides very basic definition of molecular processes in form of arguments, the classes need in the future more elaboration.

Thus, ontologies are able to represent molecular events in form of structured ontological representations. In this context, the second question arises "What we gain in using ontologies for anchoring events in text?". First, we get free and direct connection to the large biomedical community.[18] Second, the ontological hierarchies are highly relevant for identifying molecular events which often are described using various event views or complex nesting (Section 3.5.3). The `Is-a` hierarchy might be useful for inferring more events from descriptions of event instances later. The `Part-of` hierarchy allows inferences of events given only sub-event descriptions. Third, arguments of molecular events can be represented in ontologies and classified according to ontological class considered, so for example, the `Gene Expression` event is defined to involve as an argument a `Gene` instance. Still, most of ontologies do not exploit all techniques for an comprehensive and exhaustive representation of molecular events yet. These issues have to be faced in the near future by the biomedical community.

Given the crucial role of ontologies for the biomedical domain and the expressiveness of ontologies in form of instruments for detailed hierarchies plus argument definitions (which should be more elaborated in the future), the ontological concept of molecular events is preferred in this work. However, in order to connect knowledge hidden in text to ontologies, it needs a kind of bridge from text to ontologies in form of textual annotations. The rules and workflows for such annotations are designed in annotation guidelines. In the following I present various ways for large-scale ontology-based annotation approaches of molecular events in the literature.

3.6 Annotations as a Bridge between Text and Ontologies

The role of event annotated corpora is crucial not only for development of event extraction approaches but for equitable standard evaluation of NLP systems in general.

[18] There is a growing number of biomedical ontologies that are accessible via various platforms. For example, GO is a member ontology of an Open Biomedical Ontologies (OBO) Foundry experiment, which aims to provide interoperable reference ontologies (`http://obo.sourceforge.net/`). Biomedical ontologies have also recently been made accessible via BioPortal (Noy et al., 2009), which is an open repository of biomedical ontologies (271 ontologies available in May 2011) and provides access to ontologies via Web services and Web browsers.

In summary, four important contributions of annotated corpora are:

- Representation of language phenomena - Corpora are used for studying language phenomena such as the distribution of language patterns.
- Gold evaluation data - Corpora are considered to be standard data for the evaluation and comparison of NLP tools.
- Training data - Corpora are necessary for the systems based on machine-learning algorithms, in particular for supervised techniques.
- Domain adaptation data - Corpora created for special language domains are used for adaptation of general NLP tools to particular domains.

The last decade has witnessed a proliferation of corpora that have been semantically annotated by domain experts. These annotated corpora contain named entity and relation annotations, particularly in the area of biomedical language processing. These corpora, however, still cover only bits of the vast domain knowledge in the life sciences. Most of these corpora cover PPI and gene regulation annotations, e.g., AIMed (Bunescu et al., 2005), BioInfer (Pyysalo et al., 2007), LLL (Nédellec, 2005). Despite the variety of corpora in the biomedical domain their annotations differ in many respects (Pyysalo et al., 2008), e.g., in their coverage of different, highly specialized knowledge domains, in the different degrees of granularity of the relationships targeted, the specificity of the linguistic grounding of relations and the named entities referred to in the documents. While some corpora provide untyped, undirected annotations (AIMed), others employ annotations based on ontological definitions (BioInfer). Another major difference between these corpora is the amount of detail, i.e., the granularity of annotations. Finally, only a few corpora mark key words that represent the conceptualization of an interaction between named entities (e.g., "BioNLP 2009 Shared Task on Event Extraction" corpus (Kim et al., 2009), Grec (Thompson et al., 2009)) in the text. In this section, I focus on questions of event annotations and consider annotation models applied across different corpora using examples of gene expression regulation events. Furthermore, I introduce the GeneReg corpus developed as a part of this thesis work.

An annotation campaign is an important and challenging part for the development of an event extraction system. There are at least three categories of people involved in the development of event annotated corpora. These people fill the roles of either *information extraction (IE) specialist*, or *annotation guidelines developer* or *annotator*. The IE specialist is usually a computer scientist or computational linguist who initiates an annotation campaign. The annotation guidelines developer or supervisor and the annotators are domain experts and, as such, are annotation responsible people who use their expert knowledge to model or manually annotate information in texts. The IE specialist may also take the role of annotator or guidelines developer in the newswire domain for example. However, in technical domains such as molecular biology or medicine, the annotation responsible role requires a great deal of background knowledge. In these domains, therefore, the annotation process should be undertaken by experts. Annotation guidelines are usually developed by the expert supervising the annotation process. However, this creation process occurs within close cooperation between the IE specialist and the annotation responsible person, where the annotation guidelines developer is a kind of transformer capturing the necessary semantic domain knowledge in the annotation model.

In my annotation work on the GeneReg corpus (Section 3.6.4), I have taken a role of an IE specialist and annotation guidelines developer in cooperation with a graduate student of biology after I have acquired knowledge on gene expression regulation events and organized multiple discussion sessions with graduate biologists (annotators). I introduce, in the following, the main issues for a design of an annotation campaign.

3.6.1 Main Questions for Event Annotation Guidelines

Event annotation guidelines and *schemes* are bridges between ontological concepts of molecular processes and instances of these processes mentioned in documents from the life science domain. The annotation guidelines are the major source of guidance for annotators. These guidelines contain briefings on, and instructions for, an annotation process. The annotation scheme is included in the guidelines and is a form of annotation agreement between the annotation part of the team and the IE scientists applying annotation data for developing and adapting a semantic analysis system.

The event annotation guidelines have, at least, to elaborate mechanisms for dealing with four major annotation categories that help to identify and to annotate an event mention including its arguments:

- *Event extent*,
- *Lexical anchors*,
- *Syntactic anchors*,
- *Background knowledge use.*

Event extent is the portion of text which is eligible for annotations of complete event mentions. Event extent can comprise a phrase, a sentence, extends to a paragraph or even to a complete document. As a phrase is usually too small for expressing complex molecular events, and as annotation beyond sentence borders requires anaphora resolutions, most event annotation guidelines fix the event annotation on a sentence level.[19] In this work, molecular events are considered only within a sentence. Sentences are the most common units used for the extraction of relationships between named entities, because most biomedical relationships mentioned in text are shown to be intra-sentential. The study by Ding et al. (2002), for example, measuring numbers of inter- and intra-sentential interaction relationships in PubMed abstracts, shows that only 15% of relationships would be overlooked in sentence-focused information extraction approaches which do not exploit coreference resolution. Thus, there is clear evidence to support the selection of sentences as natural linguistic units to extract relations between biomedical entities.

Lexical anchors or *triggers* are key words that lead an annotator to a decision about an annotation. In annotation guidelines, annotators are usually provided with an initial list of possible lexical anchors. They then extend this list during the annotation process. How much extension of the list is permitted may be restricted in the guidelines. For example, annotation guidelines may not allow annotations of events including verb nominalizations or adjectives. During event annotation, trigger words are usually explicitly marked by annotators.

Syntactic anchors allow an annotator to decide whether an event is actually described in a particular extent. The direct object and subject connected with the help of an event verb are usually the most certain and frequent indicators for a molecular event. Furthermore, annotation guidelines might restrict annotations on particular syntactic structures, for example, in relative clauses.

Given their rich *background knowledge*, annotators may draw inferences from text and thus provide more information in their annotations than is explicitly stated in the specific text. Annotation guidelines should, therefore, include pointers for the proper use of background knowledge for text interpretation. Thus, for example, "*p50-p65-heterodimer*" is the result of a binding event not explicitly present in that extent. Every biologist knows that the state "heterodimer" precedes a binding process. They are, consequently, seduced into annotating this event in that extent. This can be avoided by appropriate restriction guidelines. Furthermore, in the sentence-wise annotation approach, biologists are not allowed to use knowledge described in other sentences, even if they are in the same paragraph.

Annotation guidelines usually use a descriptive form to provide connections between ontology classes and annotation of their instances in text. The annotation schemes model the event annotation in text. Looking at the variety of relation and event

[19] The "BioNLP 2009 Shared Task on Event Extraction" guidelines allow the annotation across sentences if anaphoric expressions are annotated.

annotated corpora in the biomedical domain referred in the introduction part of Section 3.6, my impression is that there are two basic annotation schemes for molecular process annotations which are in focus of the biomedical NLP community:

- *trigger-driven* – annotate the roles played by entities in relation to the event predicate (*trigger*) in text (e.g., Genia event corpus),
- *entity-driven* – annotate propositional relations between entities in text, mostly in a sentence (e.g., LLL corpus).

In the trigger-driven approach, an annotator aims to classify precisely the roles entities play for concrete event predicates such as the `Binding` trigger. In this approach, we can represent molecular processes with a single participant such as the `Gene Expression` process. On contrary, in the entity-driven approach, in particular in the well known PPI detection task, an annotator considers a range of interaction eventive relations among proteins, usually without detailed classification of any relations detected or with core classification. It is evident that in this approach at least two entities have to participate in a molecular event. Both approaches are presented in the following sections.

3.6.2 Trigger-driven Annotation

In the trigger-driven annotations, the event scheme has the following form for annotation of an event instance:

(3.29) $\texttt{Event}(Event_x) \wedge \texttt{Trigger}(Event_x,\ trigger\_word) \wedge \texttt{Role}(Event_x,\ argument_1) \wedge \ldots \wedge \texttt{Role}(Event_x,\ argument_n)$

where $Event_x$ is the Neo-Davidsonian event variable, $trigger\_word \in Tw$ (Tw is the set of event trigger words), $argument \in Thing$ ($Thing$ comprises named entities and event mentions). $\texttt{Role} \in Role$ ($Role$ is the set of possible roles that arguments can take), and `Trigger` indicates the trigger property of the *trigger_word* to serve as an event trigger for the current event $Event_x$. Through the *trigger_word* the event is explicitly connected to text.

Trigger-driven annotations are grounded in the explicit linguistic expressions used for denoting events (e.g., various interaction types are linked to the key interaction words). The annotation approaches may differ in their event types, in the allowable triggers they list and in their argument types or roles.

The annotation project led by the Tsujii laboratory produced the "BioNLP 2009 Shared Task on Event Extraction" corpus in a trigger-driven annotation approach.[20] This corpus contains a sample of 950 MEDLINE abstracts. The corpus covers nine molecular event types. The given set of molecular events include e.g., `Binding`, `Gene Expression`, `Transcription`, `Negative Regulation`, and (unspecified) `Regulation`. In this corpus, the event predicates are not only verbs and their nominalizations but also adjectives, adverbs, and even phrases. Only two major types of roles (inspired by the work of Dowty (cf. Section 3.3.2)) are considered to be played by participants in all molecular events, i.e., `Cause` (bio-entity which affect the way of occurrence of an event) and `Theme` (bio-entity whose properties are changed by the event). Different types of events have different argument structures. While basic events such as `Protein Catabolism` or `Transcription` involve only one participant, `Regulation` events can have a complex argument structure involving other events as participants.

An example of annotation according to the "BioNLP 2009 Shared Task on Event Extraction" guidelines using the Davidsonian representation form for the example sentence "*NFAT1 appears to be the major NFAT family member* ***responsible*** *for the initial* ***increased expression*** *of IL-4 by primed CD4 T cells.*" (PMID[21] 9916709) looks as follows:

(3.30) $\texttt{Gene\_Expression}(Event_1) \wedge \texttt{Trigger}(Event_1,\ \textit{expression}) \wedge \texttt{Theme}(Event_1,\ \textit{Il-4})$

(3.31) $\texttt{Positive\_Regulation}(Event_2) \wedge \texttt{Trigger}(Event_2,\ \textit{increased}) \wedge \texttt{Theme}(Event_2,\ Event_1)$

(3.32) $\texttt{Regulation}(Event_3) \wedge \texttt{Trigger}(Event_3,\ \textit{responsible}) \wedge \texttt{Theme}(Event_3,\ Event_2) \wedge \texttt{Cause}(Event_3,\ \textit{NFAT1})$

Please note that I use the underscore in event names if they are used in formal expressions. The example sentence contains one individual event, `Gene Expression` and two nested regulatory events, `Positive Regulation` and `Regulation`.

3.6.3 Discussion of Trigger-driven Annotation

In this section I discuss the advantages and disadvantages (Table 3.8) of the trigger-driven event annotation approach.

The trigger-driven annotation allows representation of events with a single participant. For example, gene expression, transcription, and protein catabolism are

[20] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/`

[21] PMID is a PUBMED identifier of the abstract document.

Table 3.8: Advantages and disadvantages of the trigger-driven annotation approach.

	Advantages (Trigger-driven)
1	Annotation of events with n participants where $n \geqslant 0$.
2	Granular representation of complex events.
3	Text anchoring.
4	Semantic role labeler application-ready annotation.
	Disadvantages (Trigger-driven)
1	Lexical restriction of the event predicates allowed (optional).
2	Syntactic restrictions on argument attachment structures (optional).
3	Laborious annotation of complex molecular events.
4	High risk of inconsistency due to the complexity of annotations.
5	High annotation costs due to the variety of annotations.

molecular processes involving only one argument, namely a protein. They require only a single `Theme` argument.

One of the greatest attractions of the trigger-driven approach is its suitability for representing granular structures of molecular events. As events may be part of more complex nested events, the trigger-driven annotation allows for very detailed granular representation of all possible interactions between participants. This granularity is explicitly anchored in a text through event triggers and might be useful for a comprehensive and exhaustive pathway generation. It is evident that this approach is similar to semantic role annotation in the newspaper domain. A corpus created in this way would be application-ready for a semantic role labeler which could be easily retrained on, and applied to, a new application domain (cf. Section 4.2).

However, granularity and complexity of annotations hide high risks of inconsistency and high annotation costs (Table 3.8). For example “BioNLP 2009 Shared Task on Event Extraction” data contains annotations that are described in the annotation guidelines as not annotatable. The “BioNLP 2009 Shared Task on Event Extraction” guidelines do not allow annotation of stative views of events. For biologists as annotators it seems hard to decide on event views. Thus, some annotators provide annotation of static descriptions although they are not allowed to according to the annotation guidelines. For example “*interleukin-12 p40 transcript*” and “*p65-p50 heterodimer*” are annotated with `Transcription` and `Binding` event respectively although the annotation of states is not allowed in the “BioNLP 2009 Shared Task on Event Extraction" corpus. Unfortunately, the Inter-Annotator Agreement figures are not available for the "BioNLP 2009 Shared Task on Event Extraction" data.

The inconsistency in annotations may lead to learning problems for supervised learning approaches and can lead to a noisy gold standard for evaluation. Another disadvantage of very granular annotations are high annotation costs. It is evident that the more complex the guidelines, the longer the annotation takes. The time and person costs of the trigger-driven annotation approach are assumed to be higher than the costs of the entity-driven approach described in the following section.

3.6.4 Entity-driven Annotation

The entity-driven annotation approach focuses on (binary) eventive propositional relations between entities. It functions in the following way:

(3.33) $\texttt{Event}(Event_x) \wedge \texttt{Role}(Event_x,\ entity_1) \wedge ... \wedge \texttt{Role}(Event_x,\ entity_n)$

where $Event_x$ is the neo-davidsonian event variable, $entity \in Entity$ ($Entity$ comprises named entities). $\texttt{Role} \in Role$ ($Role$ is a set of possible roles that arguments can take).

This kind of event annotation has no textual anchoring in text and usually refers to a complete event extent (mostly sentences). All PPI corpora are annotated with this approach, they provide annotation of various molecular eventive propositional relations without precisely classifying them. This kind of annotation is used for potentially complex annotations of interactions in molecular events which are expressed by more than one predicate in text. The gene expression regulation events are of this sort (cf. Section 3.5).

An example of the entity-driven annotation approach is the Gene Regulation Corpus (GeneReg), the result of an annotation campaign led by the Julie Lab. A preliminary version of this corpus has been described in Buyko et al. (2008). GeneReg was designed to provide annotations covering mentions of eventive propositional relations in gene expression regulation events. My contribution in the creation of this corpus was the development of annotation guidelines in cooperation with a graduate student of molecular biology and the supervision of the complete annotation process. The GeneReg corpus created currently consists of 314 Medline abstracts dealing with the regulation of gene expression in the model organism *E. coli.*[22]

[22] A set of 32,155 abstracts was compiled from Medline based on a query including the MeSH terms *Escherichia coli*, *Gene Expression* and *Transcription Factors* (amongst others). From this set I randomly selected a corpus of 314 abstracts for manual annotation.

On this document set, domain experts annotated gene expression regulation events between genes and regulators affecting the expression of the genes. This annotation was based on the GRO (Beisswanger et al., 2008) class `Regulation Of Gene Expression` (ROGE). An event instance contains two arguments, *viz.* `Agent`, the entity that plays the role of modifying gene expression, and `Patient`, the entity whose expression is modified (cf. Dowty's `Proto-Agent` and `Proto-Patient` in Section 3.3.2).

At least the patterns presented in Section 3.5 are available in the GeneReg corpus, (e.g., from the most frequent, containing mention of regulation verbs, adjectives and nominalizations like "*regulator*", to uncertain expressions such as "*be essential*", "*be involved in*", adjectives indicating requirements or dependencies such as "*dependent*", and causal relation constructions between molecular processes in which gene and transcription factors are involved).[23]

Here, I present an example annotation for the sentence "*Regulation of jun and fos gene expression in human monocytes by the macrophage colony-stimulating factor*" from the GeneReg corpus:

(3.34) `ROGE`($Event_1$) ∧ `Patient`($Event_1$, *jun*) ∧ `Agent`($Event_1$, *macrophage colony-stimulating factor*)

(3.35) `ROGE`($Event_2$) ∧ `Patient`($Event_2$, *fos*) ∧ `Agent`($Event_2$, *macrophage colony-stimulating factor*)

This sample sentence contains two gene expression regulation events caused by the transcription factor "*macrophage colony-stimulating factor*". These events are represented in the GeneReg corpus as eventive propositional relations between the transcription factor "*macrophage colony-stimulating factor*" and genes "*jus*" and "*fos*". The GeneReg corpus is presented in detail in Appendix (Section A.1).

3.6.5 Discussion of Entity-driven Annotation

The entity-driven approach can be selected if the annotation process tends to become complex and more than one argument needs to be annotated. The advantages and disadvantages of the entity-driven annotation approach (Table 3.9) are discussed in the following.

The major advantage of this annotation approach is that it allows easy capturing of eventive propositional relations between entities involved in complex molecular processes. The annotator does not need to provide detailed linguistic annotations

[23] States proper are excluded from the GeneReg corpus.

Table 3.9: Advantages and disadvantages of the entity-driven annotation approach.

	Advantages (Entity-driven)
1	Easy annotation of complex eventive propositional relations (even if many events are involved).
2	High speed annotations due to guidelines transparent for biologists.
3	For partial corpus annotations.
4	Usually restrictions for syntactic structures.
5	Usually no restrictions for lexical anchors.
	Disadvantages (Entity-driven)
1	Annotations are not granular.
2	No explicit textual reference.
3	High risk of the annotations being hard to capture by a statistical system.

of nested events in order to represent the desired connections between the entities of interest. Another advantage is the possibility of providing partial annotations of a corpus. For example, in annotating gene expression regulation events, where the entity-driven annotation approach can be applied, annotators only need to annotate event mentions where these events happen. In the trigger-driven approach it would be necessary to annotate all mentions of events that are part of the regulation of gene expression events such as gene expression, transcription, binding and all mentions of regulation processes. This would lead to an immense overhead for the annotation project.

Furthermore, clear annotation guidelines for entity-driven event annotation are usually designed in close cooperation with annotators and do not restrict the annotation process with too many linguistic issues. Thus, annotators can identify and annotate text according to the extent to which they comprehend the text based on biological background knowledge. Avoiding granular annotations allows for a high speed annotation process.

Nevertheless, this flexible and relatively unrestricted annotation approach also has disadvantages (Table 3.9). The major drawback for the annotated corpus is a lack of granular annotations. So the entity-driven approach is unsuitable if very granular information extraction with explicit textual reference is required from the system. Furthermore, such a general annotation approach might hide dangers for an extraction system, because the annotators do not refrain from annotating very complex expressions as no syntactic structure and lexical anchor restrictions are formulated in the annotation guidelines.

3.7 Linking of Trigger-Driven and Entity-Driven Annotations

The question how to decide on an annotation approach can be answered only given an extensive analysis of information to be annotated and the requirements from event extraction tasks. In some cases, both annotation approaches can be combined or one annotation approach can support another. For example, the GeneReg corpus is enriched with the trigger-driven approach annotations. These trigger-driven annotations are given in the form of "trigger-only" event annotation, which treats events as situation entities with no consideration of arguments (cf. Section 3.4.6). This additional annotation level in the style of a trigger-only annotations has a supporting role which requires annotations of lexical anchors indicating gene expression regulation events, parts of these events and causal relations between them.[24] However, the entity-driven annotations remain the primary information encoded in this corpus.

My emphasis in this section is on the compatibility, and thus the linkage between trigger-driven and entity-driven annotations. I selected for this study annotation schemes of two independently developed corpora, e.g., the GeneReg scheme and the "BioNLP 2009 Shared Task on Event Extraction" scheme. The "BioNLP 2009 Shared Task on Event Extraction" scheme does not envisage annotation of gene expression regulation events as such. However, the Genia ontology, which was used in the "BioNLP 2009 Shared Task on Event Extraction" scheme, contains classes for general molecular events that can be used for annotation of gene expression regulation events, e.g., `Regulation` class with its subtypes `Positive Regulation` and `Negative Regulation`, and `Gene Expression`, `Transcription`, `Mutation` and `Localization` event classes. Given these classes, I can re-annotate GeneReg using "BioNLP 2009 Shared Task on Event Extraction" trigger-driven annotation guidelines.

In this study, I explored 150 of the regulation of gene expression (ROGE) events annotated in the GeneReg corpus (about 13%). For 142 ROGE events (approximately 90%), I was able to provide (nested) Genia event class annotations so that

[24] In discussion sessions with annotators it comes to the fore that the "trigger-only" annotations could support the decision process for the annotation of eventive propositional relations.

the corresponding GeneReg ROGE events can automatically be inferred. The arguments of GeneReg ROGE events can be represented in various Genia events, e.g., `Transcription` or `Gene Expression` events.

An example of both annotation approaches for a sentence "***XapR** regulates the expression of xanthosine phosphorylase (**XapA**).*" are presented below.

The GeneReg annotation has the following form:

(3.36) $\texttt{ROGE}(Event_1) \land \texttt{Patient}(Event_1,\ XapA) \land \texttt{Agent}(Event_1,\ XapR)$

This GeneReg ROGE event can be represented by means of two cascaded Genia events, i.e., `Regulation` and `Gene Expression`:

(3.37) $\texttt{Gene\_Expression}(Event_1) \land \texttt{Trigger}(Event_1,\ \textit{expression}) \land \texttt{Theme}(Event_1,\ XapA)$

(3.38) $\texttt{Regulation}(Event_2) \land \texttt{Trigger}(Event_2,\ \textit{regulates}) \land \texttt{Theme}(Event_2,\ Event_1) \land \texttt{Cause}(Event_2,\ XapR)$

For 15 ROGE events (approximately 10%), annotations of a cascade of `Regulation` events are necessary.

For 19 ROGE events (12%), annotations of Genia `Mutation` event are needed. A `Mutation` event denotes the process by which genetic material undergoes a detectable and heritable structural change. Experimental environments for gene regulation detection often involve genetic modifications of genetic material. By means of these genetic modifications and the expression levels of other genes, researchers explicitly draw conclusions about the role of the transcription factor in the gene regulation processes (cf. nesting of events in Section 3.5.3). The sentence "*Transcription of the chromosomal **asr** was abolished in the presence of a **phoB-phoR** (a two-component regulatory system, controlling the pho regulon inducible by phosphate starvation) deletion mutant.*" can be annotated as follows. In the GeneReg annotation approach the annotator needs to annotate two ROGE events in the form of eventive propositional relations between the *phoB*, *phoR* proteins and the *asr* gene. The GeneReg annotations have the following form:

(3.39) $\texttt{ROGE}(Event_1) \land \texttt{Patient}(Event_1,\ asr) \land \texttt{Agent}(Event_1,\ phoB)$

(3.40) $\texttt{ROGE}(Event_2) \land \texttt{Patient}(Event_2,\ asr) \land \texttt{Agent}(Event_2,\ phoR)$

If the trigger-driven approach is pursued, it would produce the seven nested events presented below. This sentence needs concepts describing `Transcription`, `Mutation`, `Localization`, and `Negative Regulation`, the particular mentions are nested.

(3.41) $\texttt{Transcription}(Event_1) \land \texttt{Trigger}(Event_1,\ \textit{transcription}) \land \texttt{Theme}(Event_1,\ asr)$

(3.42) $\texttt{Mutation}(\mathit{Event}_2) \wedge \texttt{Trigger}(\mathit{Event}_2, \textit{deletion mutant}) \wedge \texttt{Theme}(\mathit{Event}_2, \mathit{phoB})$

(3.43) $\texttt{Mutation}(\mathit{Event}_3) \wedge \texttt{Trigger}(\mathit{Event}_3, \textit{deletion mutant}) \wedge \texttt{Theme}(\mathit{Event}_3, \mathit{phoR})$

(3.44) $\texttt{Localization}(\mathit{Event}_4) \wedge \texttt{Trigger}(\mathit{Event}_4, \textit{in the presence of}) \wedge \texttt{Theme}(\mathit{Event}_4, \mathit{Event}_2)$

(3.45) $\texttt{Localization}(\mathit{Event}_5) \wedge \texttt{Trigger}(\mathit{Event}_5, \textit{in the presence of}) \wedge \texttt{Theme}(\mathit{Event}_5, \mathit{Event}_3)$

(3.46) $\texttt{Negative\_Regulation}(\mathit{Event}_6) \wedge \texttt{Trigger}(\mathit{Event}_6, \textit{abolished}) \wedge \texttt{Theme}(\mathit{Event}_6, \mathit{Event}_1) \wedge \texttt{Cause}(\mathit{Event}_6, \mathit{Event}_4)$

(3.47) $\texttt{Negative\_Regulation}(\mathit{Event}_7) \wedge \texttt{Trigger}(\mathit{Event}_7, \textit{abolished}) \wedge \texttt{Theme}(\mathit{Event}_7, \mathit{Event}_1) \wedge \texttt{Cause}(\mathit{Event}_7, \mathit{Event}_5)$

However, in 10% of the annotations I could not provide the Genia class annotations. Of these 15, seven relations (about 5%) are statements and not events and eight ROGE events (again, about 5%) are too complex and cannot be represented as Genia events. For example the sentence "*Primer extension analysis of the **asr** transcript revealed a region similar to the Pho box (the consensus sequence found in promoters transcriptionally activated by the **PhoB** protein) upstream from the determined transcription start.*" contains such a tricky eventive relation between *asr* and *PhoB*, which cannot be annotated according to "BioNLP 2009 Shared Task on Event Extraction" guidelines. In summary, this study reveals that we can connect to a large extent (90% of data considered), various annotation approaches. This knowledge will be used in this work in an event extraction study (see Chapter 6).

3.8 Summary

This chapter provided an overview about various concepts of events in different research fields, e.g., philosophy, theoretical and computational linguistics. The excursion to these related research fields helped to reflect about concepts of events in biomedical NLP given the intricacies of molecular event descriptions such as micro and macro views of events (cf. Section 3.5.2) and complex nesting of events (cf. Section 3.5.3). Finally, I subsumed under the term *(molecular) event* in this work the events proper, stative views on events (in the sense of Croft (1990)) and facts about events (Bennett (1996)), and subscribe that the *eventuality* is an appropriate concept for capturing molecular events in biomedical literature. Given this concept of event, I presented various approaches to model molecular events in biomedical NLP, such as PASBio, BioFrameNet, and an ontology-supported approach to events which was preferred for this thesis. Consequently, this chapter introduced basic concepts for annotation of molecular events in the literature, and discussed two general annotation ways for representing event mentions in texts, *viz.* the trigger- and the entity-driven annotation approaches. Corpora created this way can be exploited in the large-scale advanced event extraction approach, developed in this thesis, which is presented in the following chapter.

Chapter 4

Event Extraction from Biomedical Texts

This chapter presents concepts behind the event extraction approach which constitutes a major part of the work for this thesis.

The workflow of an event extraction system usually has a *pipeline* architecture (Figure 4.1). The pipeline starts with input that consists of scientific texts, usually providing results of laboratory experiments. The running text first has to be preprocessed by a range of Natural Language Processing (NLP) tools for text segmentation, syntactic and semantic analysis before an event extraction engine can start to detect molecular events. The small cog wheels in Figure 4.1 denote a close connection between the services coming from NLP tools and the event extraction engine. Given these requirements of an event extraction task, this chapter first presents the required NLP analysis steps preceding the event extraction step (Section 4.1). Other sections of this chapter focus on the event extraction engine proper. Section 4.2 presents a summary of related work on event and relation extraction research. Section 4.3 presents the event extraction task. The remaining Sections 4.4 to 4.6 focus on the detailed description of the event extraction approach that is the heart of the whole event extraction system. The concluding Section 4.7 gives an overview of the implementation of the Jena Relation eXtraction (JReX) system, which comes with the JReX event extraction component itself (developed in this thesis) and a range of NLP tools (collected and, if necessary, adapted to the biomedical domain) required for high-performance event extraction.

4.1 NLP for Biomedical Event Extraction Solutions

This section presents, in a nutshell, the necessary NLP steps before an event extraction engine can start to work. I particularly want to demonstrate characteristics of the biomedical language as a sublanguage with its specific regularities (cf. Section 2.2.3), which permeate all the levels of NLP taken into consideration in this work. The domain dependence of lexical semantics and syntactic properties of a

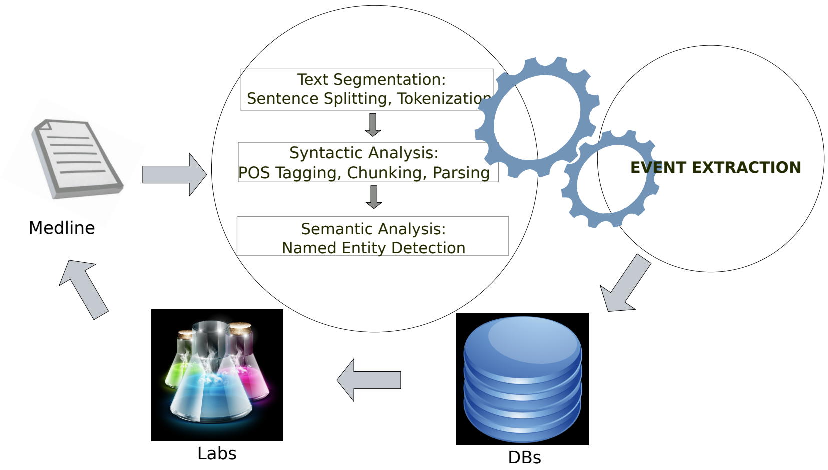


Figure 4.1: Event extraction in a workflow of an IE application for the biosciences.

sublanguage comes to the fore when NLP tools from the newspaper domain are used in the biomedical domain. Therefore, my aim is to present successful adaptations of newswire NLP tools to biomedicine and to report on their state-of-the-art performance in this domain.

4.1.1 Sentence Splitting

The running input text must be split into sentences prior to any NLP processing. In general, *sentence splitting* is the classification/disambiguation task of potential sentence delimiters, such as ".", "!". Sentence splitting in the biomedical domain is a more challenging task than in the newspaper domain, because sequences of periods and capital letters, for example, are less reliable sentence border indicators than in an English newspaper text. In addition to decimal points ("*1.3*") and common abbreviations ("*e.g.,*" "*i.e.,*"), biomedical texts contain a range of organism names ("*E. coli*", "*f. sp. Lycopersicim*"), author name acronyms ("*L. Hoffmann*"), and cited journal acronyms ("*J. Biol. Chem*") which include punctuation markers. This makes sentence splitting more difficult for a scientific text than for a newspaper text. These problems have been tackled in the past. For sentence splitting, rule-based and machine learning-based approaches have been applied (e.g., Xuan et al. (2007), Tomanek et al. (2007a)). Tomanek et al. (2007a) present a Conditional Random Fields (CRF)-based (Lafferty et al., 2001) sentence splitting approach which performs very well (99.6% accuracy). Given the reported 0.4% error rate, this task can be widely considered as having been solved for the biomedical domain.

4.1.2 Tokenization

After the splitting of running text into sentences, the next segmentation step, *tokenization*, takes place before any other linguistic analysis (morphological, syntactic or semantic) can be performed. Tokens are basic linguistic units such as words, numbers, punctuation symbols, acronyms, and abbreviations. Although tokenization can have a strong impact on the performance of semantic and syntactic components, this task is underestimated just as much as the sentence splitting task. The bulk of tokens in running text can be extracted by separating tokens at white spaces and punctuation symbols. In doing this, the naive separation of tokens at potential punctuation symbols such as ".", ",", "-" or "(" hides true tokenization challenges. These characters can be parts of words and should not be considered as symbols separating tokens in running text. Parentheses, hyphens, slashes and symbols frequently appear as characters in biomedical terms (e.g., "*Ca(2+)*", "*Il-2*", "*4-(4-formyl-4'-methyldiphenyl-amino)benzaldehyde*"). Therefore, the tokenization of scientific text, in this case biomedical text, should be treated very carefully. The tokenization performance results reported by Tomanek et al. (2007a) reveal that the tokenization problem has still not been solved (96.7% accuracy).

Furthermore, in considering tokenization as a preparatory step for syntactic and semantic analysis, we should take care that tokens build a comprehensive basis for NER tools and parsers. Here, various tokenization styles can be considered. For example, composed adjectives that have an internal semantic structure ("*CA-dependent*") or entities coordinated by slashes ("*Il-2/4*") have to be tokenized for semantic interpretation, but it is not necessary to perform the tokenization for effective parsing here. Tomanek et al. (2007a) criticize a conservative tokenization by the Genia Treebank corpus (Tateisi et al., 2005), and promote the tokenization style of the PennBioIE Treebank (Bies et al., 2005). But while the Genia Treebank tokenization style seems more suitable for parsing, the PennBioIE corpus tokenization seems more beneficial for the NER task. I applied in this work various tokenization approaches for various tasks, using the PennBioIE style for semantic tasks and the Genia style for parsing.

4.1.3 Morphological Analysis

In English the majority of words (e.g., verbs and nouns) undergo morphological variation, both syntactically-motivated ("*induc-es*", "*induc-ing*") and derivation-motivated (“*induc-tion*”) (Hahn and Wermter, 2006). The inflection process leads to a wide variety of morphological forms for the same lexical element, while derivation processes slightly change core meaning of the base form. The morphological normalization of these forms to their basic canonical form would reduce the comple xity of textual data for many applications (e.g., NER and term recognition). This is the task of stemming and lemmatization algorithms. While the stemming approach looks for a *stem* (here “*induc*”) of lexical elements, the lemmatization approach outputs *lemmas*, canonical lexical elements which are usually listed in lexicons, (here “*induce*” and “*induction*”). In general, Hahn and Wermter (2006) distinguish two methodological approaches, a lexicon-driven approach and a lexicon-free approach. In a lexicon-driven procedure the text is matched against a lexicon which contains morphological forms linked to their canonical form. The lexicon-free techniques use various suffix-stripping algorithms to create the basic word form. The most prominent and successful English stemmer algorithm, the PORTER stemmer, which was first introduced for information retrieval purposes, was presented by Porter (1980). This simple and general algorithm, which strips general English suffixes (e.g., *-ed*, -*ing*, or *-ion*) leaving valid stems, has been widely applied in the biomedical domain. As for the domain lexicon-driven approaches, the SPECIALIST LEXICON has been extensively used for lemmatization purposes (Browne et al., 1998). However, in the latter approach the search in lexicons frequently retrieves multiple basic forms of various part-of-speech categories. For example, the word “*flies*” would be matched to at least two lexical entries, “*fly*” as a verb and “*fly*” as a noun. Therefore, for disambiguation of retrieved lemmas, there is a need for POS tagging, which is the focus of the next subsection.

4.1.4 POS Tagging

Part-of-speech (POS) tagging is a beneficial pre-processing step for NER task and relationship extraction, and a necessary step for parsing. A POS tagger attributes a part-of-speech tag or tags from a predefined final tag set to words (e.g., noun, verb, determiner, adjective, and preposition). The PENN TREEBANK POS tag set is one of the most popular tag sets (Marcus et al., 1994). Frequently, POS taggers are integrated into parsers, as in the well-known CHARNIAK parser (Charniak and Johnson, 2005), for example. But the idea of a pipelined POS tagger is favoured as it simplifies the adaptation of this task to new domains, thus avoiding elaborate and time-consuming adaptation of the complete parsing algorithm. POS tagging approaches use rule-based (Brill, 1995) and ML techniques (Ratnaparkhi, 1996) with high performance results. Brill (1995) and Ratnaparkhi (1996) report 96.5% accuracy on the Wall Street Journal (WSJ) part of PENN TREEBANK and their POS taggers are publicly available.

However, the usage of newspaper POS taggers on biomedical text leads to a severe decline in their performance if they are applied to another domain (cf. e.g., Hahn and Wermter (2004)). Tsuruoka et al. (2005) list typical errors of newspaper-trained POS taggers that are not adapted to the biomedical text. These errors affect, among other things, derivations used as e.g., modifiers in noun phrases ("*after mitogen binding*", "*binding*" tagged JJ but should be NN) and complex noun terms ("*more T-cell determinants*", "*T-cell*", tagged JJ but should be NN). These studies show that domain adaptation is obligatory for achieving reasonable tagging performance. Domain adaptation of POS taggers is effected either by retraining POS taggers on manually annotated biomedical treebank corpora (e.g., Genia Tagger (Tsuruoka et al., 2005), MedPOS Tagger (Smith et al., 2004), or OpenNLP Tagger[1] (Buyko et al., 2006)), or through the creation of an (automatically gathered) domain-specific lexicon with POS information (e.g., (Miller et al., 2007)). The performance of adapted POS taggers matches or even exceeds state-of-the-art figures for POS tagging. Buyko et al. (2006) report 98.9% accuracy on the Genia corpus for the OpenNLP Tagger re-trained on Genia. Tsuruoka et al. (2005) report 98.4% accuracy on the Genia corpus and 97.2% on the WSJ corpus for an ML-based POS tagger re-trained on the union of WSJ, Genia and PennBioIE treebanks.

4.1.5 Chunking

The usage of information about shallow syntactic structures of a sentence has been shown to be beneficial for a range of named entity detection and relationship extraction approaches (e.g., Sun et al. (2007), Jiang and Zhai (2007)). Partial parsing, or chunking, aims to split text into chunks, which are the basic syntactic segments of a sentence. Jurafsky and Martin (2009) define chunks as "flat, non-overlapping segments of a sentence that constitute the basic non-recursive phrases corresponding to the major parts-of-speech found in most wide-coverage grammars" (cf. Jurafsky and Martin (2009), p. 485). Chunks are (in comparison to constituents) non-recursive and non-hierarchical typed base phrases, such as noun chunks, verb chunks, and prepositional chunks.

Training data for chunking can be extracted from constituency-based treebanks using head finding rules. For example, the training data sets for the CoNLL 2000 Shared Task (Tjong Kim Sang and Buchholz, 2000) have been automatically generated from the Penn Treebank corpus using the ChunkLink script[2].

There are various methodological approaches to solve the chunking task, for example rule-based methods (e.g., Abney (1996)), hidden Markov models (e.g., Zhou and Su

[1] http://incubator.apache.org/opennlp/

[2] http://ilk.kub.nl/~sabine/chunklink/

(2000)), and supervised machine learning techniques (e.g., Sha and Pereira (2003)). The CoNLL 2000 Shared Task showed that a large number of participating systems could achieve performance figures above 90% F-score. The best CoNLL 2000 Shared Task system (Kudoh and Matsumoto, 2000) performed with 93.5% F-score.

In the biomedical domain, the situation is similar to the usage of POS taggers developed originally on the newspaper domain. Life science texts "differ from general language in the structure and complexity of noun phrases" (cf. Wermter et al. (2005)). Thus, domain adaptation is necessary to avoid the loss of chunking performance on the new domain. Wermter et al. (2005), Buyko et al. (2006) and Kang et al. (2010) focus on evaluation and adaptation of shallow parsing tools to the biomedical domain.

Buyko et al. (2006) first re-trained the OpenNLP chunker[3] on the data extracted from biomedical treebanks and Genia and PennBioIE Treebank corpora. The adapted system ultimately achieved 93.6% F-score on Genia and 89.5% on PennBioIE Treebank data. At first glance, it seems that the chunker performs markedly better if it is trained on Genia, particularly concerning the important recognition rate for NPs (92.3% on Genia *vs.* 85.1% on PennBioIE). However, these results must be treated with caution because of some inadequacies of the ChunkLink script in converting treebank annotations into chunk notation, especially where PennBioIE is concerned.

The OpenNLP chunker (re-trained on Genia) was judged to be the best in the study by Kang et al. (2010), investigating six commonly used chunkers (Gate chunker[4], Genia Tagger[5], Lingpipe chunker[6], MetaMap[7], OpenNLP , and YamCha[8]). This evaluation study shows that the OpenNLP chunker significantly outperforms all freely available chunkers in the biomedical domain. The overall F-score of 93.6% by the OpenNLP chunker on Genia Treebank (Buyko et al., 2006) is a state-of-the-art figure (in comparison with performance figures from CoNLL 2000). In particular, the reported recognition rates by Buyko et al. (2006) for NPs (92.3% in Genia), PPs (96.9% in Genia) and VPs (95.9% Genia) are essential for deeper linguistic analysis, which is the subject of the following section.

[3] http://incubator.apache.org/opennlp/

[4] http://gate.ac.uk/

[5] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

[6] http://alias-i.com/lingpipe/

[7] http://mmtx.nlm.nih.gov/

[8] http://chasen.org/~taku/software/yamcha/

4.1.6 Syntactic Parsing

As exploitation of syntactic structures will be shown crucial for event extraction in this thesis (cf. Sections 5.2.5 and 5.3), the current section aims to carefully introduce concepts and categories from syntactic parsing and focuses by the end on parser adaptation in the biomedical domain in detail.

Syntactic parsing is defined as "the task of recognizing a sentence and assigning a syntactic structure to it"(Jurafsky and Martin (2009), p. 461). Syntactic parsing results are important for solving a range of information extraction tasks. For example, in order to extract an `Activate` relation between two named entities ("*Il-2 protein*" and "*Inf-alpha protein*") from the sentence "*Il-2 protein activates Inf-alpha protein*", it is crucial to know that "*Il-2 protein*" is the subject of this sentence while "*Inf-alpha protein*" is the object. While the benefits of syntactic parsing for relationship extraction are obvious, there are information extraction applications that exploit syntactic parsing as well. Finkel and Manning (2009) show that parsing is even beneficial for named entity recognition, for example.

The syntactic structure of a sentence is not directly observable (in contrast to the linear order of words) and is represented through ordered relations between words and word groups. In general, syntactic structures are based on two types of relation: part-whole ("Teil-Ganzes" (Brinker (1977), p. 91) and dependency relations. Part-whole relations are the focus of what is called *constituency*-based parsing, while dependency relations form the basis of *dependency*-based parsing (cf. Brinker (1977), p. 91).

Constituency-based syntactic analysis is based on the idea that words in a sentence build groups that behave as single units called *constituents*. This fact can be demonstrated using various *constituency tests* (Grewendorf et al., 1990).[9] In the substitution test, for example, phrases such as "*Il-2 protein*", "*the activator protein IL-2*", "*the human Il-2 protein*", "*it*" can replace each other in the sentence "*Il-2 protein activates Inf-alpha protein*" without violating its grammaticality. Phrase-Structure Grammars (PSGs) are used for dealing with constituency. PSGs ideas can be dated back to the psychologist Wundt (1900) but were first formalized by Chomsky (1956) and reinvented by Backus (1959), and independently by Naur et al. (1960) (cf. Jurafsky and Martin (2009), p. 421-457). The constitutive parts of PSGs are a lexicon of words/symbols, abstract phrase categories (e.g., NP, S), and production/derivation rules that express how the words from a lexicon can be grouped and ordered. *Context-Free Grammars* (CFG) are the most common type of PSGs used for dealing with natural language syntax. CFGs have been shown to not be

[9] Meanwhile it has been demonstrated that *constituency tests* alone are not sufficient for classifying words in constituents (Fanselow and Felix, 1990).

adequate for representing long-distance dependencies and modeling languages with a more flexible linear order than English, such as Czech or Russian. Therefore, dependency grammar offers a more flexible framework.

Dependency grammars can be dated back to the fundamental work of Tesnière (1959), but the notion of syntactic dependencies is older than constituency grammars and has its roots in ancient Greek linguistics. In recent years, dependency grammars have increasingly been recognized as an alternative to long-prevailing constituency-based parsing approaches, particularly in semantically-oriented application scenarios such as information extraction. In dependency grammars, the syntactic structure of a sentence is described by binary dependency relations between words. In a nutshell, in dependency trees of sentences, nodes represent single words and edges account for head-modifier relations between single words. Despite this common understanding, concrete syntactic representations often differ markedly from one dependency theory/parser to the other. Computational implementations of dependency grammars include (for English) Link Grammar (Sleator and Temperley, 1993), Constraint Grammar (Karlsson et al., 1995), and Minipar (Lin, 1998).

Furthermore, dependency graphs can be automatically derived from constituency parse trees using so called *head rules* (Collins, 2003). The head is the word in a phrase that is grammatically the most important. For example, "*protein*" is the head of the phrase "*Il-2 protein*". Heads determine the type of the phrase, for example noun is the head of a noun phrase and verb is the head of a verb phrase. An example of head rules are the rules presented by Xia and Palmer (2001). In general, head rules need two iteration steps (cf. Jurafsky and Martin (2009), p. 450):

1. Mark the head modifiers on each word in a phrase structure, using the head information table.
2. In the dependency structure, make each modifier depend on the corresponding head.

The idea of a lexical head or functor for each phrase dates back to Bloomfield (1914) and is crucial for a range of lexicalized grammars, such as Head-Driven Phrase Structure Grammar or HPSG (Pollard and Sag, 1994) and Combinatory Categorial Grammar (CCG) (Steedman (1987), Steedman (1996)), which thus integrated ideas of constituency and dependency-based grammars.

However, given only the linear word order of a sentence and a syntax grammar (lexicon and rules), there are various possible ways of structuring a sentence and thus various interpretations are possible. This structural ambiguity comes to the fore in the following example:

- "*(Dogs and cats) in the exhibition.*"
- "*Dogs and (cats in the exhibition).*"

The structural ambiguities (in particular coordinations and prepositional attachments) can be resolved using information about *lexemes in structure* probabilities of a particular language domain. For example the selected phrase "*Dogs and cats in the exhibition*" means rather that both "*dogs*" and "*cats*" are in the exhibition and not only "*cats*". To resolve this ambiguity, we might exploit a model with probabilities for lexemes appearing in various syntactic structures. In order to assess empirically the distribution of syntactic structures in languages, the NLP community created a range of what are called *Treebank* corpora that contain sentences annotated with parse trees. The Penn Treebank (PTB) for English is a very prominent treebank, together with the Negra treebank for German (Skut et al., 1997), the Susanne treebank for English (Sampson, 1994), and the Prague Dependency Treebank for Czech (Hajǐc, 1998). Treebanks play a crucial role in parser development. Treebank corpora are heavily used for training probabilistic dependency and constituency parsers based on lexicalized probabilistic context-free grammars (PCFGs), such as modern statistical parsers (e.g., Charniak parser (Charniak and Johnson, 2005), Collins parser (Collins, 2003), and Bikel parser (Bikel, 2004)). As for the performance of constituency-based parsers, Collins (2003) reports labeled recall (LR) and labeled precision (LP) of 89.6% and 86.5% respectively. Statistical dependency parsers have attracted widespread attention, as witnessed by recent activities performed as part of the "CoNLL Shared Tasks on Multilingual Dependency Parsing" (Buchholz and Marsi, 2006). The best performing (in terms of accuracy) parser in the "CoNLL Shared Tasks on Multilingual Dependency Parsing", the MST parser, is a statistical ML-based parser. The labeling accuracy of up to 91.5% was previously achieved by the MST parser on English data (McDonald, 2006).

Due to different evaluation settings and criteria for constituency and dependency-based parsers, the reported performances are not comparable. Therefore, the proper and comparable evaluation of parsers is a highly relevant and challenging subject intensively discussed in the NLP community, as in the recent dedicated workshop on "Cross-Framework and Cross-Domain Parser Evaluation" (Bos et al., 2008). The main questions here are the proper and common evaluation basis (constituents or dependency relations), as discussed by Tam et al. (2008), for example, and the evaluation level, especially the evaluation of unbounded dependency detection (e.g., (Rimell et al., 2009), (Nivre et al., 2010)). Increasing numbers of researchers insist on the formal and common evaluation of parsers on detected dependencies. Even task-oriented evaluation of syntactic parsers has recently attracted attention (e.g., Miyao et al. (2008), Buyko and Hahn (2010)). Some studies have focused on trade-offs between speed and accuracy of parsers, such as Cer et al. (2010).

Table 4.1: Cross-domain F-score performance of the Charniak parser. The table is taken from McClosky et al. (2010).

	Test		
Train	Genia	Brown	WSJ
Genia	**83.6**	64.6	66.6
Brown	71.5	**86.3**	80.6
WSJ	74.9	83.8	**89.0**

As statistical parsers are trained on available treebank corpora, they inevitably learn some domain-specific syntactic properties, in particular at the lexical level and in syntactic parse tree distribution. As a consequence, it was shown that statistical parsers did not perform well across different domains (Sekine, 1997). McClosky et al. (2010) measured differences in parser performance between the target text and source domain. Divergence between training and test domain implies a dramatic loss in parser performance. Table 4.1 shows some figures from the work of McClosky et al. (2010). We clearly see from these figures that domain shift considerably influences parsing accuracy. The corpora set used in this study includes the best-known news articles domain corpus, the Wall Street Journal (WSJ) part of the Penn Treebank, as well as the literature domain corpus Brown (Francis and Kučera, 1979) and the biomedical domain corpus Genia (Tateisi et al., 2005) collected from the Medline database. A Charniak parser trained on WSJ only achieves a 74.9% F-score on Genia and an even lower 71.5% F-score if trained on Brown. An F-score of 83.6% is possible if a Charniak parser is trained and evaluated on Genia. Similarly, considerable performance loss is shown for newspaper and literature domain corpora if a statistical parser is trained on the biomedical domain. The parser evaluation study by Clegg and Shepherd (2005) also confirms the cross-domain performance loss for a range of statistical parsers. Although Clegg and Shepherd (2005) do not provide the figures for re-trained parsers, the evaluation results for WSJ-trained parsers on Genia clearly illustrate the lower F-score results on the biomedical domain treebank and are thus in accordance with the study outcomes of McClosky et al. (2010). For example, Clegg and Shepherd (2005) report that the Charniak-Lease parser achieves only an 80.2% F-score performance on Genia, while its performance on the WSJ corpus is convincing, with an 89.5% F-score (Lease and Charniak, 2005).

Lease and Charniak (2005) and Park (2001) investigate in more detail potential sources of parsing errors by shifting a parser to a new domain. Lease and Charniak (2005) focus on parser adaptation by targeting in particular the *unknown word rate* of a new domain text. They investigate the integration of in-domain knowledge data such as part-of-speech lexica, a dictionary of collocations and a named entity dictionary. This lexically driven adaptation is motivated primarily by investigations of numbers of unknown words from different domains. Lease and Charniak (2005) demonstrate that the unknown word rate on the Genia corpus for a parser trained on the WSJ parser is 25.5%. The unknown word rate increases when moving from general language to increasingly technical domains. By integrating a lexicon and an entity dictionary, they achieve a reduction in the unknown word rate and an increase in parser performance of 3.3 percentage points F-score. The Genia-unadapted Charniak parser performs with a 76.3% F-score, while the Genia-adapted variant achieves a 79.6% F-score. However, integration using specialized lexica, which is the most frequent approach to adaptation of a parser, is not sufficient for customizing a statistical parser to a new sublanguage domain.

Park (2001) classifies CCG parser errors and reports that the three largest error sources are coordinations, appositions, and subject/object relations, while POS errors are negligible. Thus, in addition to the unknown word rate level, the parse tree distribution differs from domain to domain, as it was shown previously by Sekine (1997). The incorrect detection of subject/object relations is mainly due to differences in sub-categorization frames for verbs. The verb "*encode*" in PropBank (Palmer et al., 2005), with the meaning "make into a code, encrypt", requires two arguments, `Arg0` (codemaker) and `Arg1` (message). In PASBio (Wattarujeekrit et al., 2004) the verb means "to specify the genetic code for" and requires two arguments `Arg0` (`gene` or `rna`) and `Arg1` (`gene product`). Here, the meaning of "*encode*" is different. Even more problems occur with the verb "*express*", which in biomedicine usually refers to the expression of genes. Here, the argument structure differs greatly from newspaper or general language use of the verb "*express*" (cf. Section 3.5.5). The outcomes of these studies suggest that statistical parsers have to be adapted to a new domain by re-training them at least on in-domain treebank corpora. This has been done, for the Charniak parser for example, by McClosky and Charniak (2008) exploiting some self-training algorithms in addition to the use of the Genia corpus. The adapted self-trained McClosky-Charniak parser achieves 87.6% on the Genia corpus. A range of other constituency-based full parsers have also been optimized and adapted for parsing biomedical text. In this process, the Genia Treebank plays a central role in adaptation. Parsers such as Enju, with a Genia-trained model (Miyao et al., 2008), and the OpenNLP parser (Ratnaparkni, 1998), re-trained on Genia by Buyko et al. (2006), are publicly available and achieve state-of-the-art results that are comparable to those of WSJ-trained parsers.

Adaptation of dependency parsers to the biomedical domain is similar to the adaptation of phrase-structure parsers. Here, the key role is played either by the lexically driven adaptation with integration of specialized vocabulary or full re-training of statistical dependency parsers using available treebank corpora. Pyysalo et al. (2006b) first addressed lexical adaptation of two dependency parsers, Link Grammar Parser (LGP) (Sleator and Temperley, 1991a) and the Connexor Machinese Syntax (CMX) parser (Tapanainen and Jarvinen, 1997). They targeted in particular the unknown word rate and evaluated three approaches to reducing unknown words. These were automatic lexicon expansion with in-domain terms from the Specialist Lexicon, as previously proposed by Szolovitz (2003), the use of morphological cues for better prediction of POS tags[10], and direct usage of in-domain POS taggers. The best way of targeting the lexical bottleneck was found by Pyysalo et al. (2006b). They proposed applying a high-quality domain POS tagger, such as the Genia Tagger, to perform the named entity pre-processing and to treat named entities, after pre-processing, as single proper nouns (Pyysalo et al., 2006a). They found that the integration of Specialist Lexicon terms caused a negligible and statistically insignificant increase in parser performance, indicating that the extension of a dictionary alone is not sufficient to address the problems of parser adaptation. In its best configuration, the LGP parser was able to recover 73% of dependencies, but it was outperformed by the GMX, which recovered 80% of dependencies. In addition to an increased improvement in the accuracy of lexically adapted parsers, Pyysalo et al. (2006b) also report a dramatic increase in parsing efficiency. For LGP, for example, parsing time decreased by 45% compared to the unadapted LGP version. This first formal evaluation of dependency parsers, using standard evaluation and a manually prepared corpus of biomedical texts, has been followed by only a few studies focused on the evaluation of (dependency) parser adaptation in terms of their positive effect on information extraction performance, as illustrated by Miyao et al. (2008) and Buyko and Hahn (2010). In both works, the dependency parsers had been re-trained on the Genia Treebank converted to dependency representation. In this thesis, I apply various dependency and constituency parsers adapted to the biomedical domain (cf. Section 5.3).

4.1.7 Coreference Resolution

The event extraction approach proposed extracts events within sentences and coreference resolution tools are not applied in this thesis work. However, this subject is relevant for the future work. Therefore, this section introduces coreference resolution shortly.

The language phenomena presented in previous subsections are all situated at the sentence level. One of the supreme disciplines in language analysis mentioned above

[10] Morphological cues for the biomedical domain are e.g., suffix *ase* for nouns (“*kinase*”).

- syntactic parsing - also operates on single sentences. However, as natural language does not function as an arbitrary stringing together of sentences, discourse phenomena deserve closer attention. Let us consider a biomedical text passage with inter-sentential phenomena. "*The D-allose operon of Escherichia coli K-12. Escherichia coli K-12 can utilize D-allose, an all-cis hexose, as a sole carbon source. The operon responsible for D-allose metabolism was localized at 92.8 min of the E. coli linkage map. It consists of six genes.*" (PMID 9401019). In this passage the underlined noun phrases used by the author all mean the same entity, "*D-allose operon*". Thus, the author may denote the same entity within text using various *referring expressions* that all *corefer* to the same entity. As natural language is rife with alternatives for denoting entities (e.g., names, indefinite noun phrases, pronouns or demonstratives), the automatic resolution of referring expressions is a challenging task. The NLP task called *coreference resolution* aims to map all mentions of referring expressions to the referred entity (*referent*).[11] The noun phrase (NP) and, in particular, pronominal reference resolution, has attracted a lot of attention in the NLP community, especially since 1995, when the first coreferentially annotated corpus, MUC-6, was provided.

Only a small number of coreference resolvers have been developed for the biomedical domain. The most widely known of these apply either heuristic-based approaches (e.g., Castaño et al. (2002)) or (semi)-supervised techniques (e.g., Yang et al. (2004), Gasperin (2006)), and since resolution is not performed on standard publicly available data, it is difficult to talk here of state-of-the-art performance data for coreference resolution in the biomedical domain. The reported results on the GENIA-MEDCO corpus (Yang et al., 2004) peak at 73.9% F-score. Huang et al. (2010) reports 70.7% F-score on the full-text corpus annotated by Gasperin (2006).

As far as the part of event extraction proper is concerned, an event extraction component is applied. Before this crucial component will be introduced, I give an overview of related work on relationship and on event extraction in particular in order to juxtapose my approach to solve this task.

4.2 Related Work

While systems for the recognition and interpretation of named entities have reached, by and large, a stable performance plateau at the 80% F-score level, the extraction

[11] Coreference resolution is closely related to anaphora resolution. Anaphora resolution aims to identify an antecedent (entity previously introduced in the discourse) for an anaphoric mention. If an anaphoric mention and its antecedent refer to the same entity in the world, they are called coreferent. Therefore, the resolution task is called coreference resolution.

of relations between these entities lags far behind these figures (cf. Chapter 2). In the newswire domain, the ACE program (Doddington et al., 2004) features the best system with a 24.1% ACE-score for the detection of eventive propositional relations (i.e., ACE events).[12] The performance of the winning system of the BioCreAtIvE II Protein Interaction Subtask (Hirschman et al., 2007) went up to 28.8% F-score (cf. Section 2.2.4). Although for both competition series strict real-world requirements were imposed on the task – the recognition and interpretation of all named entities involved, plus the recognition and interpretation of the associated relation (and, for the biomedical domain, the mapping of entities onto unique database identifiers) – relation extraction remains a challenging research problem under any conceivable conditions. The wide variety of expression patterns for descriptions of relations between entities makes this task computationally hard. A range of sophisticated approaches have been developed to solve this task and this section focuses on presenting these approaches. In the following, I present previous work in newspaper and biomedical domain on relation and event extraction approaches that are considered equally relevant for this thesis, as the argument extraction task in event detection is the relational part of an event extraction system.

As for the event extraction task, the approach depends on the concept of event introduced in this thesis, e.g., event as a document cluster, template, situation frame or even situation entity (see Section 3.4). Three major event extraction streams predominate here. The most representative is the analysis of events in the form of verbal predicate-argument structures. This research branch releases *semantic role labeling* (SRL) systems that automatically assign semantic roles to the arguments of a predicate. The training data is usually extracted from the argument-structure annotated PropBank corpus (Section 3.4.5). As SRL methods are very frequently discussed in the literature, I focus here only on competitions which provide standards for the evaluation of SRL techniques, i.e., the "CoNLL-2004 Shared Task on Semantic Role Labeling" (Carreras and Màrquez, 2004) and the "CoNLL-2005 Shared Task on Semantic Role Labeling" (Carreras and Màrquez, 2005). The best system in the "CoNLL-2004 Shared Task on Semantic Role Labeling" (Hacioglu et al., 2004) achieved a 69.5% F-score. The major idea of the winner system was to solve the SRL task as a chunking problem by labeling syntactic chunks (base phrases) as arguments of a selected predicate. In the "CoNLL-2005 Shared Task on Semantic Role Labeling", this approach inspired the best system (Koomen et al., 2005) to integrate multiple SRLs and, to combine their results within a joint inference process. This system increases the performance in solving SRL up to an F-score of 79.4% on the WSJ part of the PropBank.

The emphasis of the SRL task is only on the detection of semantic roles for se-

[12] Results are published at `http://www.nist.gov/speech/tests/ace/2007/`.

lected predicates. The SRL task is a subtask of a more challenging event extraction approach, the “Frame Semantic Structure Extraction” task in the SemEval competition series (Baker et al., 2007). This is the second stream for event extraction research (cf. Section 3.4.4). The aim of this task is, first, to detect words and phrases invoking frames defined in FrameNet and, second, to extract arguments of the detected frame. The performance results of participating systems are lower than for the SRL task, ranging between F-scores of 35% and 55% on the SemEval competition data. The best performing system in the “Frame Semantic Structure Extraction 2007” (Johansson and Nugues, 2007b) integrated information from dependency parse trees in order to classify arguments of a frame predicate. This work provides one of the high-performance relation extraction systems that integrates dependency parsing for detecting predicate-argument relations. Considering this success in using dependency trees in frame argument extraction and given the previous under-estimation of the application of this parsing approach in the NLP research for semantic extraction tasks, the CoNLL-2008 and CoNLL-2009 workshops organized “Shared task on Joint Parsing of Syntactic and Semantic Dependencies” (cf. Surdeanu et al. (2008); Hajič et al. (2009)). The majority of systems developed apply a pipeline approach, starting with detection of syntactic dependencies and integrating the results in the SRL task afterwards. The winning system achieved a 85.4% F-score for the SRL part of the “Shared task for a Joint Extraction of Syntactic and Semantic Dependencies 2009” (Zhao et al., 2009). The integration of syntactic dependencies in the SRL approach was shown to be beneficial. In the light of this experience and the success in integrating dependency parse results for semantic applications, this thesis considers dependency trees as the major knowledge source for solving the event extraction tasks.

The consideration of events as situation entities (the third major event extraction stream) is the focus of the TempEval task associated with the SemEval challenge series in 2007 and 2010. This task required the detection of events and time expression with a subsequent extraction of temporal relations between events (Verhagen et al., 2007). TempEval-2 in SemEval 2010 (Pustejovsky and Verhagen, 2009) was a follow-up to TempEval-1, which was an initial evaluation exercise based only on temporal relation tasks. The TempEval-2 task required in addition automatic detection of event extents. The winning system for this task achieved a very high F-score of 88.0% (Llorens et al., 2010). As TempEval is based on the TimeBank and considers events as situation entities, the winning system applied a sequence labeling model for labeling event extents.

Previous work on relation and event extraction in the biomedical domain in the form of PPI extraction, however, started with simple methods, such as the detection of bag-of-word-style *co-occurrences* of entities of interest within documents or sentences (e.g., Jenssen et al. (2001)). Co-occurrence-based approaches are characterized by high recall at the cost of very low precision. Furthermore, the type and direction of a relation usually cannot be determined by relying on co-occurrence data alone. Approaches that focus on higher precision but often suffer from weaker recall are based on *manually defined patterns* (e.g., Blaschke et al. (1999)). Some pattern-based approaches make use of morpho-syntactic and syntactic information and are based on *automatically learned patterns* from large corpora (e.g., Huang et al. (2004), Hakenberg et al. (2005)). These methods provide higher recall than those based on manually defined patterns. *Rule-based approaches* typically exploit full parse data of sentences and additional semantic information (e.g., Yakushiji et al. (2001), Leroy et al. (2003), Fundel et al. (2007)).

Going far beyond co-occurrence statistics, patterns and rules, a range of current state-of-the-art relation extraction systems apply machine learning methods. IE systems have been applying machine learning techniques since the 1990s, in particular for solving named entity recognition problems. However, there have been few studies applying ML techniques to relation extraction. The main reason for the restricted development of ML approaches applied to IE relational problems can be traced to the lack of large relationally annotated corpora. Indeed, during the last decade we have witnessed a proliferation of relationally annotated corpora and, in parallel, a rapid development of ML techniques for relation extraction. In the newswire domain, the long history of research on supervised approaches is partly due to the availability of large annotated corpora such as ACE or PropBank, which can be used for training ML models. The representative works for ML-based relation extraction in the newspaper domain are e.g., Zelenko et al. (2003), Bunescu and Mooney (2005), Kambhatla (2004), and Zhou and Zhang (2007). Newswire relation extraction research has turned in recent times to exploring semi-supervised and unsupervised techniques (e.g., Blanco and Moldovan (2011), Fürstenau and Lapata (2009)). In contrast, in the biomedical domain, supervised learning approaches still dominate the relation extraction domain, partly due to the late availability of large, annotated corpora such as AIMed[13] (Bunescu et al., 2005), which can be used for training statistical models. The systems developed either exploit kernel methods especially designed for the comparison of syntactic trees (e.g., Bunescu et al. (2005), Sætre et al. (2007), Airola et al. (2008a)), or they incorporate a variety of lexical, morpho-syntactic and syntactic features (e.g., Katrenko and Adriaans (2006), Kim et al. (2008b)) using various learning algorithms. Unfortunately, many systems are evaluated using different evaluation settings, even when using the same corpus, and the experimental settings are often unclear, even for evaluations that have been carried out. This makes any comparison of systems difficult. In consequence, the full potential of relation extraction has not been achieved. Therefore, Airola et al. (2008a) suggested indicating the evaluation settings of the instance-,

[13] ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/

sentence- or document-wise evaluation for relation detection. They proposed to use the document-wise evaluation as the default evaluation setting for PPI extraction and extracting a default test data set from publicly available corpora.[14]

PPI extraction is still the major application area for relation learning from the biomedical domain (cf. BioCreAtIvE III challenge)[15]. PPI extraction is clearly not a problem that has been solved. Given its inherent complexity, it may benefit from a methodological approach that deals with the extraction of molecular events in a bottom-up manner or restriction of interactions to specific molecular events such as those dealing with gene expression regulation. The Genia event corpus (Kim et al. (2008a)), and the GeneReg corpus, both introduced in the previous chapter, contain fairly detailed (Genia) or more precise (GeneReg) annotations of PPIs (among other things) and are first steps towards the extraction of specific pathways with faithful information about the molecular events involved (Oda et al. (2008)). Biological encodings of such kinds of annotations are, in the trigger-driven annotation approach (in Genia), also backed up by a more adequate linguistic view on events. Accordingly, biological events are considered as predicate-argument relations similar to the definition of the "CoNLL Shared Task on Semantic Role Labeling" (Carreras and Màrquez (2004)) and the ACE challenge (Doddington et al. (2004)), which, however, were both run in the newspaper domain. The biomedical natural language processing (BioNLP) community has already tried to work on detailed molecular events for selected predicates using semantic role labeling approaches which have been shown to be useful in the newspaper domain. Bethard et al. (2008), for example, analyze the event structure underlying protein transport in cells, while Yakushiji et al. (2001) extract events that are expressed by a number of biomedical verbs (e.g., "*bind*" or "*activate*").

Standard evaluation settings for event extraction were introduced for the first time by the "BioNLP 2009 Shared Task on Event Extraction" (Kim et al., 2009). This task was a first step towards extracting detailed information about a range of molecular events, which could be evaluated on unseen common test data. In the main task "Task 1", 42 teams participated and 24 of them submitted final results. This task, "Event detection and characterization", required, for a sample of 260 Medline abstracts, that all mentioned events are determined. These events were to be chosen from a given set of molecular event types. The complexity of Task 1 is increased by the requirement that arguments are not only allowed to be proteins but also events, which quite naturally leads to the nesting of events. The winning system, Turku University (Björne et al., 2009), with a 51.95% F-score, achieved the milestone result in that competition, followed by the Julie Lab team (JReX system) (Buyko

[14] This setting is strictly used in this thesis for the event extraction task (cf. Section 5 and 6).

[15] `http://www.biocreative.org/`

et al., 2009), which peaked at a 46.7% F-score and the CONCORDIA UNIVERSITY system (Kilicoglu and Bergler, 2009), which ranked third in this challenge with a 44.6% F-score. TURKU UNIVERSITY used a supervised machine learning approach incorporating rich features, mostly extracting information about dependency tree positions of event triggers and arguments, and intermediate dependency paths. An extension of the TURKU system, the TOKYO system, has recently been developed by Miwa et al. (2010). TOKYO system's event extraction capabilities are based on the TURKU system, yet TURKU's manually crafted rule system for post-processing and the combination of extracted trigger-argument relations is replaced by a machine learning approach in which rich features collected from classification steps for triggers and arguments are re-combined. TOKYO achieves an overall F-score of 53.29% on the test data, thus outperforming TURKU by 1.34 percentage points. The JULIE Lab team applied the event extraction approach proposed in this thesis, which is described in the following Section 4.3. The CONCORDIA UNIVERSITY system applied event trigger dictionaries and manually defined rules, operating on dependency trees along the paths connecting event trigger and potential arguments, where special attention was given to appositions and coordinations. In the "BioNLP 2011 Shared Task on Event Extraction" (Kim et al., 2011), the new top performance in the same Task 1 was achieved by the FAUST system (Riedel et al., 2011), which peaked at a 56.0% F-score exploring a combination of several models. A major contribution came from the UMASS dual decomposition (Riedel and McCallum, 2011), with a 55.2% F-score as a stand-alone version, and also from Stanford event parsing (McClosky et al., 2011), with a 50.0% F-score as a stand-alone version.

I present below the event extraction approach developed in this thesis. A general description of event extraction task in terms of its subtasks is provided in Section 4.3, while the methodologies intended to be used to solve each subtask are discussed in the subsequent sections (Section 4.4 – 4.6). The pipeline of the implemented JREX system, reflecting the text pre-processing and the event extraction task proper is described in Section 4.7.

4.3 Event Extraction Problem

This section describes the steps necessary for solving the event extraction task that is to find event instances in text.

Event extraction is a complex IE task that can be decomposed into a number of subtasks depending on whether the focus is on the event itself or on the arguments involved. The event extraction subtasks can be categorized as follows:

Table 4.2: Event extraction subtasks in the entity-driven and trigger-driven event approaches.

Event Extraction Subtask	Entity-driven	Trigger-driven
Event trigger identification	optional	required
Event trigger typing	optional	required
Argument identification	required	required
Argument ordering	required	required

- **Event trigger identification** deals with the large variety of alternative verbalizations of the same event type, e.g., whether the event is expressed in a verbal or in a nominalized form (e.g., "*A is activated*" and "*the activation of A*" both refer to the same event, *viz.* `Activation`(A)).
- **Event trigger typing** deals with the semantic classification of an event trigger (at the lexical level) and the assignment to an event type (at the conceptual level). Since the same trigger may stand for more than one event type, event trigger ambiguity has to be resolved as well (for an example of trigger ambiguity, cf. Section 4.4.2).
- **Argument identification** is concerned with finding all necessary participants in and conditions of an event, i.e., the arguments of the event.
- **Argument ordering** assigns to each participant that has been identified its functional role within the event, mostly `Agent` and `Theme` (often also designated as `Patient`).

The event extraction approach proposed in this thesis can produce both event representation levels, i.e., the trigger-driven and the entity-driven one. While the previous chapter (cf. Section 3.6) presented the manual event annotation schemata, this section introduces the corresponding extraction procedures to construct both event representation levels automatically. Table 4.2 provides a summary of required and optional tasks for both event extraction approaches. The main difference between the two approaches is in the argument identification step (see **Argument identification** above) and is the type of relations considered. While the trigger-driven approach considers predicate-argument relations between the event trigger and potential arguments, the entity-driven approach focuses on extraction of eventive propositional relations between potential event arguments (cf. Section 3.1.3).

Furthermore, while all subtasks are necessary for the trigger-driven extraction approach, the event trigger identification and event trigger typing subtask can be omitted in the entity-driven approach.

For the sentence

(4.1) "***Regulation*** *of* ***jun*** *and* ***fos gene expression*** *in human monocytes by the* ***macrophage colony-stimulating factor****.*"

we can extract molecular events as follows.

The trigger-driven approach demands, in the first step, the identification and typing of the triggers "*regulation*" and "*gene expression*", and in the second step the identification of arguments, i.e., "*jun*", "*fos*" and "*macrophage colony-stimulating factor*" typed as proteins. In the next step the roles of arguments are assigned in corresponding events. In this case the nesting of events is required, as the `Regulation` event has as `Theme` argument the `Gene Expression` event. The trigger-driven approach outputs the representation according to the scheme from Section 3.6.2.

The entity-driven annotation approach focuses on (binary) eventive propositional relations between "*jun*", "*fos*" and "*macrophage colony-stimulating factor*" entities and outputs the representation according to the scheme from Section 3.6.4 for two molecular gene expression regulation events with "*macrophage colony-stimulating factor*" as `Agent` and "*jun*", "*fos*" gene as `Theme` (`Patient`) arguments.

In the following sections, I will introduce the event extraction approach proposed to solve the event extraction task.

The event extraction approach in this thesis is best categorized as a combined learning approach to event detection, as it does not separate the overall machine learning task into independent event trigger and event argument learning subtasks. This choice is motivated by the characteristics of biomedical text data, which is usually only partially annotated. For example, in the "BioNLP 2009 Shared Task on Event Extraction" data molecular events are restricted to protein-type arguments. For example, "*activation of NF-kappaB*" is not annotated as a `Positive Regulation` event because "*NF-kappaB*" is a protein complex. Obviously, the effective learning of event triggers is thus constrained by the partial annotation. Given the strong correlation between trigger annotations and the occurrence of proteins as arguments, my approach is designed to learn events in one step (Section 4.6).

The event extraction approach proposed consists of three major steps – first, the detection and disambiguation of lexicalized *putative* event triggers (Section 4.4), second, the trimming of dependency graphs, which involves eliminating informationally irrelevant lexical material and enriching lexical material that is informationally relevant (Section 4.5), and third, the identification and ordering of arguments for the event under scrutiny (Section 4.6). The outcome of the event trigger detection subtask is interlinked with the outcome of the argument identification subtask in the final event creation step (Section 4.6). In the next section, I present the three major steps of my event extraction methodologies in more detail.

4.4 Event Trigger Identification

Considering the wide variety of potential lexicalized triggers for an event, their lack of discriminative power relative to individual event types and their inherent potential for ambiguity, I decided on a dictionary-based approach whose curation principles are described in Section 4.4.1. The disambiguation policy for the ambiguous lexicalized event triggers assembled in this suite of dictionaries, one per considered event type, is discussed in Section 4.4.2.

4.4.1 Manual Curation of Event Dictionaries

I began by collecting event trigger dictionaries from the original Genia event corpus (Kim et al. (2008a)), the most important event annotated corpus in the biomedical domain. I extracted triggers of nine representative molecular events considered in the "BioNLP 2009 Shared Task on Event Extraction" (referred to as Shared Task) challenge from this corpus. The event triggers extracted were then automatically lemmatized using the Specialist NLP Tools for lexical variant and derivation generation.[16] I selected the Specialist Lexicon as it provides good coverage for syntactic (part-of-speech, subcategorization frames), morphological (base form) and orthographic information (spelling variants) for biomedical vocabulary.

In the next step, the Specialist Lexicon lemmas were ranked by two students of biology according to their predictive power to act as a trigger for a particular event type. The putative triggers were presented to the students as lists without any context, but they were allowed and encouraged to consult the Genia event corpus for clarification. This expert assessment led us to four trigger groups (these groups were determined separately for each event type):

(1) Triggers are *important* and *discriminative* for a specific event type. This group contains event triggers such as "*up-regulate*" for the event type `Positive Regulation`, "*DNA-binding*" for `Binding`, and "*phosphorylate*" for `Phosphorylation`.

[16] Specialist NLP Tools (2008 release) accessible via `http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html/`.

(2) Triggers are *important,* though *not fully discriminative,* for a particular event type. However, this deficiency can be overcome by other lexical cues within the context of the same sentence. This group with in-context disambiguators contains lexical items such as "*proteolyse*" for the event type `Protein Catabolism`.

(3) Triggers are *non-discriminative* for an event type and cannot be disambiguated even by linguistic cues within the context of the same sentence. This group contains lexical items such as "*presence*" for the event types `Localization` and `Gene Expression`.

(4) Triggers are absolutely *non-discriminative* for an event (either due to annotation errors or the lack of an appropriate event trigger). This group contains general lexical triggers such as "*observe*", "*demonstrate*" or "*function*".

The final dictionaries used for the detection of event triggers combine the first two groups. They were further extended by biologists with additional lexical material from the first group. The dictionaries thus became event type-specific. The current dictionaries for each relevant event type contain all morphological forms of Specialist Lexicon lemmas.[17]

Table 4.3 represents the size of dictionaries given lemmas or morphological forms for each event trigger dictionary. The size of the dictionaries collected for molecular events ranges between 3,732 triggers (morphological forms) for `Positive Regulation` and 75 triggers for the `Protein Catabolism` event. Regarding the dictionary sizes, the smallest are created for the `Protein Catabolism` and `Phosphorylation` events. The largest are the dictionaries for the `Positive Regulation` and `Negative Regulation` events. Based on this, I can conclude that molecular events concerning only proteins (e.g., `Protein Catabolism`, `Phosphorylation`, `Gene Expression`, `Transcription`) are expressed by a small number of trigger words. In contrast, more general molecular events such as `Binding` and `Regulation` with its subtypes can be denoted by a range of various trigger words.

4.4.2 Event Trigger Disambiguation

My preparatory experiments on Shared Task data conducted prior to solve the event extraction task indicated that the disambiguation of event triggers is beneficial for the overall event extraction results since events tend to be expressed via highly ambiguous triggers. Most of the triggers are not specific, neither for molecular event descriptions in general nor for a special event type in particular. "*Induction*", for example, occurs 417 times in the Shared Task training data as a putative trigger.

[17] These morphological forms can be automatically generated using the Specialist NLP Tools.

Table 4.3: Size of event trigger dictionaries collected for the Shared Task events, e.g., dictionaries for morphological forms and lemmas.

Event trigger type	Size (morph. forms)	Size (lemmas)
`Localization`	282	108
`Protein Catabolism`	75	37
`Phosphorylation`	83	53
`Gene Expression`	135	61
`Transcription`	177	142
`Binding`	578	281
`Positive Regulation`	3,732	1,477
`Negative Regulation`	2,070	951
`Regulation`	766	347

I talk about putative triggers because all relevant lexical items are considered as potential event triggers for an event. Only those event triggers that can eventually be connected to arguments indicate a true event. In 162 of these 417 cases "*Induction*" acts as a trigger for the event type `Positive Regulation`, 6 times as a trigger for `Transcription`, 8 times as a trigger for `Gene Expression`, while 241 occurrences do not trigger an event at all. Therefore, the developed event extraction approach included a disambiguation step preceding the extraction of any argument structures. Three word sense disambiguation heuristics considered as standard approaches were tested on words $w \in W$ which are putative event triggers of event types $e \in E$, where W represents the union of all event dictionaries and E the set of considered event types in the training data.

- **Frequency-**based heuristic:

$$\textit{trigger\_freq}(w) = \underset{e \in E}{\operatorname{argmax}}(\operatorname{freq}(w, e)) \tag{4.2}$$

  For the trigger word w the function *trigger_freq* yields the event type e which occurs, relative to w, with maximum frequency (freq(w,e)) in the training corpus.

- **TF-IDF (Term Frequency, Inverse Document Frequency) score** based heuristic:

$$\textit{trigger\_tf-idf}(w) = \underset{e \in E}{\operatorname{argmax}}(\text{tf-idf}(w, e)) \tag{4.3}$$

For the trigger word w the function *trigger_tf-idf* yields the event type e which occurs, relative to w, with maximum *tf-idf* score (*Lucene*-computed, sentence-wise)[18] in the training corpus.

- **Importance-**based heuristic:

$$\mathit{trigger\_imp}(w) = \operatorname*{argmax}_{e \in E} \left(\frac{\mathrm{freq}(w, e)}{\sum\limits_{\tilde{w} \in W} \mathrm{freq}(\tilde{w}, e)} \right) \tag{4.4}$$

For the trigger word w the function *trigger_imp* yields the event type e which occurs, relative to w, with the maximum importance in the training corpus. This importance is defined by the frequency of occurrence of w relative to all the other trigger words $\tilde{w}$ (including w) for e. For example, the importance of the trigger word "*depend*" amounts to 0.013 for the event type `Positive Regulation`, while for the event type `Regulation` it yields 0.036 (calculated on the Shared Task training data set). The trigger word "*depend*" is thus selected only for the event type `Regulation`.

4.5 Trimming Dependency Graphs

Most of the event extraction approaches presented in related work (Section 4.2) use full parsing results from a wide range of constituency and dependency parsers without any form of filtering. Given event (relation) extraction as a semantic interpretation task, plain dependency graphs as they result from deep syntactic parsing might not be fully appropriate for directly extracting semantic information. There are two main reasons for this: they contain a lot of apparently irrelevant lexical nodes (from the semantic perspective of event extraction, at least) and they also contain lexical nodes that are far too specific and that deserve some form of semantic overhauling. Trimming dependency graphs for the purposes of event extraction, therefore, amounts to eliminating informationally irrelevant lexical nodes and enriching informationally relevant ones with concept overlays. I thus deliberately rearrange the final representations for the event learners in order to avoid noise in the statistical models with overly specific syntactic and lexical data. I stipulate that this reduction of structural information is achieved in a linguistically motivated way.

[18] I employ the indexing and retrieval facilities of the Apache LUCENE search engine (`http://lucene.apache.org`).

The notion of trimming is related to sentence simplification and sentence compression. Most work on sentence compression with minimal information loss has focused on word deletion from constituency-based parsing results (e.g., Knight and Marcu (2002), Turner and Charniak (2005), McDonald (2006), Galley and McKeown (2007)). Cohn and Lapata (2008) were the first to present work on sentence compression over constituents beyond mere word deletions. They used additional operations such as substitutions, reordering or insertion of words using a tree transduction framework based on the synchronous tree substitution grammar formalism (Cohn and Lapata (2009)). Sentence compression has already been widely used for text summarization tasks, particularly extractive summaries (e.g., Vanderwende et al. (2007), Yousfi-Monod and Prince (2008), Martins and Smith (2009)). Other tasks, such as semantic role labeling and relation extraction, have also been shown to benefit from sentence simplification (Vickrey and Koller (2008), Qian et al. (2008)).

Sentence simplification in the biomedical domain has focused on splitting long sentences into shorter ones ("*that*" clauses, coordinated verb structures; cf. Huang et al. (2005)). Jonnalagadda et al. (2009) use sentence simplification techniques to transform frequently occurring compact linguistic structures in the biomedical domain into simpler ones, such as the de-elliptification of coordinations ("*alpha- and beta-catenin*" is split into "*alpha-catenin*" and "*beta-catenin*"). They also propose replacing multi-word terms by conceptually more general placeholders (e.g., "*human CREB binding protein*" replaced by `GENE`).

The sentence compression approaches in the newspaper and biomedical domains usually use chunking or constituency-based parsing. To the best of my knowledge, there are only two studies on dependency parsing-based sentence compression. One is by Yamagata et al. (2006), who prune dependency structures based on the dependency path length, and the other is by Filippova and Strube (2008), who achieve dependency tree compression by pruning subtrees dependent on a score that accounts for the probability of dependencies and the importance of dependent words.

In my trimming approach, I adapt syntactic trimming in terms of the elimination or modification of grammatical relations of de Marneffe et al. (2006) and the idea of lexical normalization mentioned in various works (e.g., Vanderwende et al. (2007)) to dependency graph structures for the extraction of argument relations. I enhance the purely eliminative operation of trimming by enriching relevant lexical carriers of event information with semantic meta data at different levels of conceptual granularity, thus elaborating on the work of Jonnalagadda et al. (2009). The proposed mechanisms aim at achieving a higher level of descriptive abstraction from which classifiers might benefit in order to avoid over-fitting to excessively specialized and overly diverse dependency structures. This will be presented in the following section.

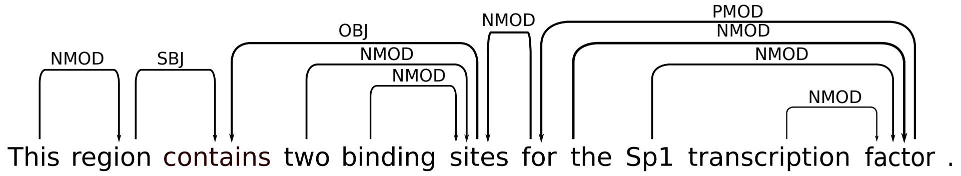

NMOD
SBJ
OBJ
NMOD
NMOD
NMOD
PMOD
NMOD
NMOD
NMOD
This region contains two binding sites for the Sp1 transcription factor .

Figure 4.2: An example of a syntactic dependency tree for the sentence "*This region contains two binding sites for the Sp1 transcription factor.*". The CoNLL 2007 dependency label set is applied in this example. "*Region*" and "*sites*" nodes are connected to the node "*contains*" using dependency relations SBJ and OBJ respectively, for example.

4.5.1 Dependency Graph Concepts

In recent years dependency parsing has increasingly been recognized as an alternative to long-prevailing constituency-based parsing approaches, particularly in semantically-oriented application scenarios such as information extraction (cf. Surdeanu et al. (2008); Hajič et al. (2009)). Yet even on purely methodological grounds it has attracted widespread attention, as witnessed by recent activities carried out as part of the "CoNLL Shared Tasks on Multilingual Dependency Parsing" (Buchholz and Marsi (2006)). For the fundamental notions underlying dependency parsing see Section 4.1.6. Briefly, in dependency graphs of sentences, nodes represent single words and edges account for directed head-modifier relations between single words. An example of a dependency graph for the sentence "*This region contains two binding sites for the Sp1 transcription factor.*" is given in Figure 4.2. Each word from this sentence is a vertex in a graph (usually called dependency node) connected to other vertices (nodes) through directed labeled edges. Every label comes from a finite set of dependency relation labels. I formalize the dependency graph representation in the following paragraph.

Let D denote a dependency graph which can be represented as follows:

$$D = (V, E) \tag{4.5}$$

where V is a set of dependency nodes or vertices and E is a set of directed labeled dependency edges which are 2-element subsets of V.

Each edge can be represented as an ordered pair of dependency vertices. These pairs build so called *incidence list.* A dependency edge $e = (x, y)$ (where $e \in E$) is considered to be directed from x to y where x is called the modifier and y is called the head (where $x \in V$, $y \in V$ and $x \neq y$). Two edges are called *incident* if they share a common vertex, these edges are *adjacent* to this vertex. Similarly, two vertices are called *adjacent* or *incident* if they share a common edge. The dependency graph from Figure 4.2 can be represented as follows: $V = \{$*this, region, contains, two, binding, sites, for, the, Sp1, transcription, factor, .*$\}$, $E = \{$*(this,region), (region,contains), (sites,contains), (binding,sites), (two,sites), (for,sites), (the,factor), (Sp1,factor), (transcription,factor), (factor,for), (contains,ROOT)*$\}$. The labels of edges can be represented as, e.g., $SBJ = \{e \in E : label(e) = SUBJECT\}$, where e is an edge in a graph D and $label(e)$ is a labeling function on edges.

Various graph operations can be performed on dependency graphs, e.g., *shortest path extraction* and *vertex contraction*. Both operations are relevant for the event extraction approach in this work. Detection of the shortest path connecting two specific vertices x and y is crucial for argument extraction in this thesis (cf. Section 4.6). The shortest path problem in a dependency graph is finding a path between two vertices in a graph such that the distance between these vertices is minimized. This distance is the sum of weights of its constituent edges (weight is usually 1). Common algorithms for solving the shortest path problem include the Bellman-Ford algorithm (Bellman, 1958) and Dijkstra's algorithm (Dijkstra, 1959). Dijkstra's algorithm works for arbitrary directed graphs with nonnegative weights. Therefore dependency graphs are suitable for an analysis with the Dijkstra's algorithm.

Another important graph operation is vertex contraction. The vertex contraction is performed on a subset of vertices in a graph. In a dependency graph D, contraction of two vertices x and y is the replacement of x and y by a single vertex z such that z is adjacent to all of the vertices to which x or y were originally adjacent. If x and y were connected by an edge, this edge is simply removed. This operation is called *edge contraction*. The vertex contraction and its special case, i.e., edge contraction is relevant for trimming procedures as presented in the next section.

4.5.2 Syntactic and Semantic Modifications

One of the main methodological contributions in this work centers on transforming the result of syntactic parsing (dependency graphs). While there was already evidence that dependency structures are much more appropriate for (biomedical) relation extraction than constituent structures (cf. Hajič et al. (2009)), I elaborated on the idea of *trimming* (and thus "normalizing") the original dependency graph on two axes. First, I eliminated informationally irrelevant lexical material from the dependency graphs through subgraph pruning and adapting dependency relations. Second, I enriched these leaner dependency graphs by enriching informationally relevant lexical nodes with additional conceptual information at different levels of semantic specificity. Whether a node is relevant or irrelevant is based on its impact on the final event representation. Thus, by "informationally (ir)relevant" I refer to lexical and dependency label material that carries crucial (or negligible) information from the information extraction perspective. Typical examples of elimination candidates are auxiliary or modal verbs. In the following two trimming operations, syntactic simplification and semantic enrichment will be introduced.

Syntactic Simplification

Despite the common understanding of dependency graphs presented in the previous section, concrete syntactic representations often differ markedly from one dependency theory/parser to the other. The differences fall into two main categories: dependency pairing or structuring (which pairs of words join in a dependency relation?) and dependency typing (how are dependency relations for a particular pair labeled?). The CoNLL dependencies, for example, are defined by 54 relation types,[19] while the Stanford scheme (de Marneffe et al. (2006)) incorporates 48 types (so-called grammatical relations or Stanford dependencies). The Link Grammar Parser (Sleator and Temperley (1991b)) employs a particularly fine-grained repertoire of dependency relations adding up to 106 types, whereas the well-known MiniPar parser (Lin (1998)) relies on 59 types. Differences in dependency structure are at least as common as differences in dependency relation typing.

In the following I focus on currently popular representations of dependency structures, i.e., CoNLL and Stanford dependency graph representations generated by various syntactic parsers.[20] In general, dependency graphs can be generated by syntactic parsers in two ways. First, native dependency parsers output CoNLL or Stanford dependencies according to which representation format they have been trained on. Second, the output of constituency-based parsers in the form of phrase structures can subsequently be converted either into CoNLL or Stanford dependencies using corresponding head rules (see Section 4.1.6). In Figures 4.4 and 4.5 I provide examples of dependency structures for CoNLL and Stanford dependency trees for the sentence "*AcnB expression is activated by CRP and repressed by ArcA, FruR and Fis from PacnB.*" extracted from a constituency tree (see Figure 4.3) using head rules.

CoNLL and Stanford dependency sets are in the NLP community currently the

[19] Computed by using the conversion script on WSJ data (accessible via `http://nlp.cs.lth.se/pennconverter/`; see also Johansson and Nugues (2007a) for additional information). From the Genia corpus, using this script I only could extract 29 unique CoNLL dependency relations.

[20] I disregard in this thesis other dependency representations such as MiniPar and Link Grammar representations. I apply Stanford and CoNLL dependency representation, because they are popular representation formats of dependency graphs and can be used by native dependency parsers for training. Furthermore, dependency-annotated data can easily be extracted from available established treebanks such as Penn Treebank or Genia.

dominant dependency representations that originate from the conversion of Penn Treebank (PTB) phrase-structure trees to the dependency format using adapted head rules (cf. Johansson and Nugues (2007a) and de Marneffe et al. (2006) for details).

I describe below both established dependency graph representations:

- **CoNLL dependencies (CD).** The dependency tree format used in the CoNLL Shared Tasks on multilingual dependency parsing (Figure 4.4). This format is used by most native dependency parsers and was originally obtained from PTB trees using constituent-to-dependency conversion (Johansson and Nugues (2007a)).
- **Stanford dependencies (SD).** This format was proposed by de Marneffe et al. (2006) for semantics-sensitive applications using dependency representations, and can be obtained using Stanford tools[21] from PTB trees. The Stanford format is widely used in the biomedical domain (e.g., by Miyao et al. (2008) or Clegg and Shepherd (2005)). SD can be produced by the Stanford parser in three modi, *basic*, *collapsed* and *ccprocessed.* The basic mode is the standard Stanford dependency representation. The *collapsed* and *ccprocessed* modi elaborate a *basic* output. The *collapsed* mode outputs prepositions and referential structures in a *collapsed* representation. An example of collapsing is the conversion of "*expression* $\xrightarrow{\text{nmod}}$ *in* $\xrightarrow{\text{pmod}}$ *cells*" to "*expression* $\xrightarrow{\text{prep\_in}}$ *cells*". The *ccprocessed* mode is an extension of the *collapsed* mode where the dependency relations of conjuncts are shared (cf. de Marneffe et al. (2006)).

There are systematic differences between CoNLL and Stanford dependencies, e.g., as regards the representation of passive constructions, the position of auxiliary and modal verbs, or coordination representation. For example, in the SD representation scheme, rather than taking auxiliaries to be the heads in passive or tense constructions (as in standard dependency theory used, e.g., in the CoNLL scheme), main verbs are assigned this grammatical function. Thus, from the perspective of semantic relation extraction, the Stanford scheme is certainly closer to the desired predicate-argument structure representations than CoNLL's scheme.

Linguistic intuition suggests that the closer a dependency representation is to the format of the targeted semantic representation, the likelier it is that it will support the semantic application. I subscribe to this idea for the task of event extraction as well, and thus narrow the distance between dependency representations and predicate-argument representations as much as possible. Hence, CD representations have to be *simplified* (refined according to the Stanford scheme).

[21] Available from `nlp.stanford.edu/software/lex-parser.shtml/`.

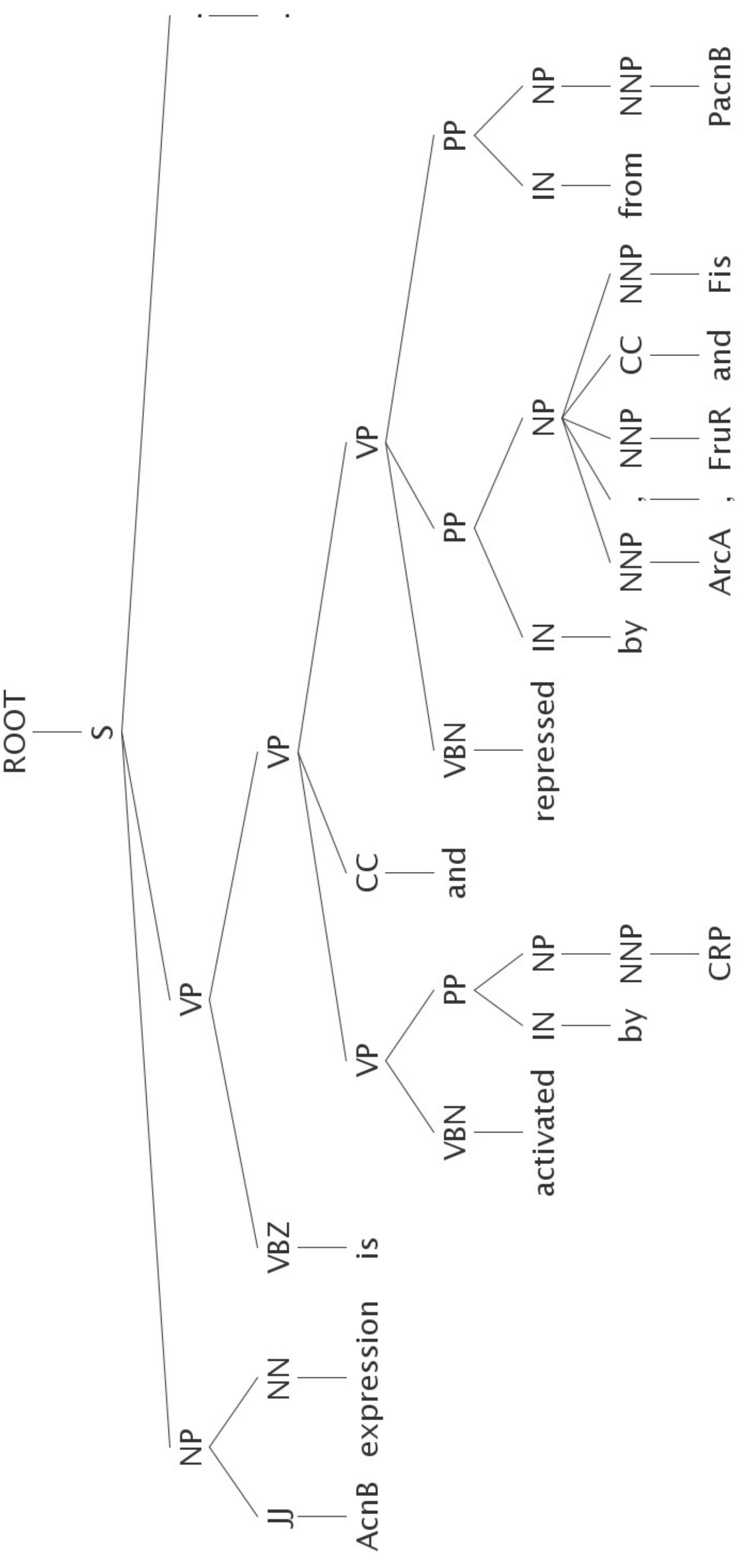


Figure 4.3: Phrase structure of a sentence “*AcnB expression is activated by CRP and repressed by ArcA, FruR and Fis from PacnB.”*, parsed by STANFORD Parser (de Marneffe et al., 2006).

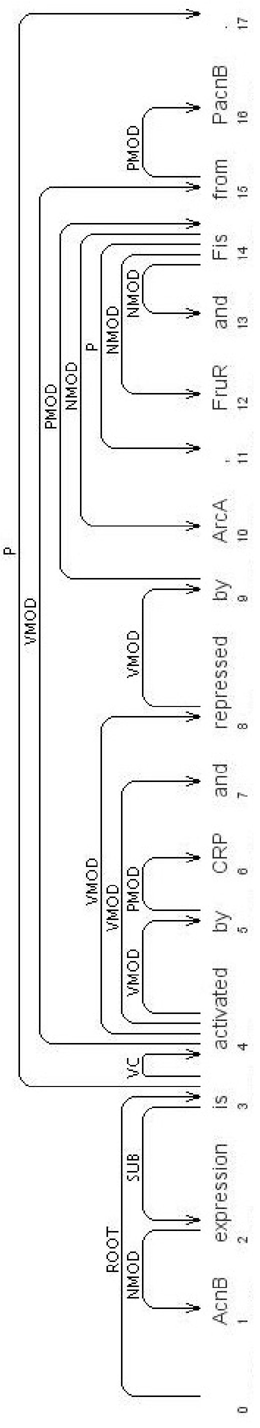


Figure 4.4: Example of CoNLL-style dependencies for the sentence "*AcnB expression is activated by CRP and repressed by ArcA, FruR and Fis from PacnB.*", as used in the GDEP parser.

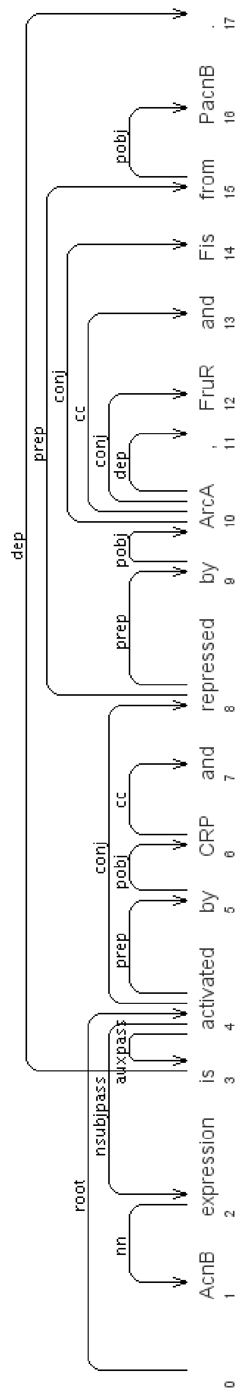


Figure 4.5: Stanford dependencies, *basic* conversion from a PTB tree for the sentence "*AcnB expression is activated by CRP and repressed by ArcA, FruR and Fis from PacnB.*".

The basic idea is derived from the SD framework for extracting grammatical relations from the PTB and collapsing the dependency tree obtained. This idea is directly reflected in the SDs, which narrow the distance between nodes in the dependency graph through collapsing procedures.

In accordance with the Stanford scheme, I propose collapsing scenarios on CD graphs. The idea to treat CD graphs was born given a range of available and high-performance native dependency parsers which rely on CD representation (e.g., MST parser (McDonald, 2006), GDEP parser (Sagae and Tsujii, 2007)). I propose so-called *trimming* procedures to treat four syntactic phenomena, *viz.* auxiliaries/modals (*aux*), prepositions (*preps*), coordinations (*coords*), and noun phrases containing action adjectives (*np action*). I now describe syntactic trimming procedures in detail. Variable v will denote a vertex and variable e will denote an edge. In the figures illustrating trimming procedures, edges pruned from dependency graphs will be marked in grey while added edges will be marked in red.

Pruning of auxiliary and modal structures in the *aux* procedure: Syntactic pruning targets auxiliary and modal verbs that govern the main verb in syntactic structures such as passives, past or future tense. For these auxiliaries/modals, the corresponding dependency nodes are pruned as governors from the CD dependency graph and the dependency relations of these nodes are propagated to the main verbs. Adhering to the dependency tree format and labeling conventions set up for the "CoNLL Shared Tasks on Multilingual Dependency Parsing" (Buchholz and Marsi (2006)), main verbs are usually connected with auxiliary verbs by the `VC` dependency relation. Thus, the heads of `VC` relations are pruned and their modifiers are promoted to the nodes of removed heads. Accordingly, in the example "*NF-kappaB may activate*" (Figure 4.6), the verb "*activate*" is promoted to the ROOT of the dependency graph and governs all nodes that were originally governed by the modal "*may*" (Figure 4.7). This procedure is a vertex contraction of vertices connected by the `VC` edge and can be represented with pseudocode as follows:

```
if label(e)==VC
do {
 head=getHead(e)
 modifier=getModifier(e)
 contractVertices(head,modifier,modifier)
}
```

In this procedure, the `VC` edge (e) is first detected in the graph and the head (auxiliary or modal verb) plus the modifier (main verb) of this edge are extracted. Afterwards, vertex contraction is performed for the head and the modifier (see *contractVertices(head,modifier,modifier)*). The place of a new node is taken by the

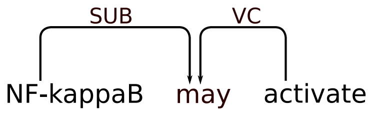


Figure 4.6: Auxiliaries representation in CoNLL dependency trees.

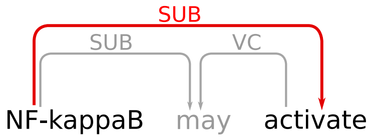


Figure 4.7: Trimming procedure for *auxiliaries and modals* on CoNLL dependency trees.

modifier node and all edges (except the connected edge `VC` which is pruned) are connected to the new node (Figure 4.7).

The incidence list for the dependency graph of the example in Figure 4.6, before the pruning of auxiliaries, is defined as $E\{$*(NF-kappaB,may),(activate,may)*$\}$. The incidence list of the dependency graph after the pruning procedure (Figure 4.7) looks like $E\{$*(NF-kappaB,activate)*$\}$.

Collapsing of prepositions in the *preps* procedure: For prepositions, the trimming procedure *preps* collapses pairs of typed dependency edges, which are adjacent to a preposition node, into a single typed dependency. An example for simplification of prepositions is the conversion of "*activation* $\xrightarrow{\text{nmod}}$ *in* $\xrightarrow{\text{pmod}}$ *cells*" to "*activation* $\xrightarrow{\text{prep\_in}}$ *cells*", as illustrated in Figures 4.8 and 4.9. The simplification of prepositions can be represented with pseudocode as follows:

```
if v.getPOS()==IN
do {
 e[]=getEdges(v)
 contractEdges(e[])
}
```

Traversing a dependency graph, the nodes (v) which are assigned the POS tag `IN` (see Penn Treebank tag set) are extracted for this procedure. All edges adjacent to this node (e[ ]) are contracted, as illustrated in Figure 4.9 where the original edges `NMOD` and `PMOD` are replaced by a new edge `PREP_IN`. The label of a new edge is a preposition word such as "*in*" plus the text "`PREP`". The incidence list of the dependency graph from the example in Figure 4.8 before the collapsing of prepositions is $E\{$*(in,activation),(cells,in)*$\}$, and after the pruning procedure (Figure 4.9) the incidence list is $E\{$*(cells,activation)*$\}$.

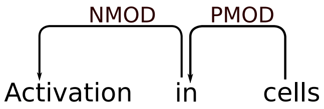


Figure 4.8: Preposition structure representation in CoNLL dependency trees.

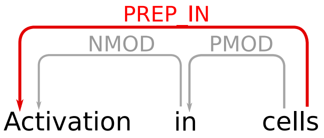


Figure 4.9: Trimming procedure for *prepositions* on CoNLL dependency trees.

Trimming of coordinations in the *coords* procedure: The lack of a discriminative coordination label in early CoNLL-like dependency sets, as used, for example, by the GDEP parser, leads to a lot of ambiguities when we extract the shortest path between coordinated elements. Some head-modifier structures are collapsed, thus making, e.g., noun compounds indistinguishable from noun coordinations at the representational level. For instance, both "*expression activation*" and "*expression and activation*" end up being represented through the shortest path "*expression* $\xleftarrow{nmod}$ *activation*". In my approach to coordination trimming, I applied first the coordination detection approach developed by Buyko et al. (2007) in order to detect conjuncts in a noun-phrase coordination. Buyko et al. (2007) classifies tokens and thus dependency nodes as conjuncts inside a noun phrase. Subsequently, rules for re-coding dependency structures are applied, e.g., propagating the head of the coordinated structure to its elements similarly to the approach of de Marneffe et al. (2006). First, the "*Il-2*" and "*Il-4*" nodes are connected with an egde `COORD`, as illustrated in Figure 4.11, if this edge is not produced by parsers.[22] In the next step the conjuncts share the same head, as illustrated for the phrase "*Il-2 and Il-4 activation*", so the elements "*Il-2*" and "*Il-4*" will share the head "*activation*". The propagation of the dependency relation from the first conjunct to all the other conjuncts and changes of dependency labels between conjuncts within the coordination can be represented with pseudocode as follows:

```
if v1.isConjunct() & !v1.headConjunct()
do{
  headConjunct v2 = v1.getHead()
  edge2HeadConjunct e1 = getEdge2Head(v1, v2)
  e1.setLabel(COORD)
```

[22] Normally, the dependency parsers trained on corpora annotated with the CoNLL 2007 and 2008 dependency sets, produce the `COORD` edge between conjuncts.

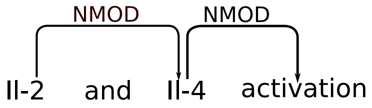


Figure 4.10: Coordination structure representation in CoNLL dependency trees.

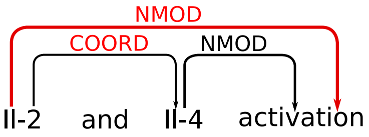


Figure 4.11: Trimming procedure for *coordinations* on CoNLL dependency trees.

```
  headOfHeadConjunct v3 = v2.getHead()
  edgeOfHeadConjunct e3 = getEdge(v2, v3)
  string labelOfEdge =  e3.getLabel()
  setEdgeBetween(v1, v3, labelOfEdge)
}
```

First, in a coordination structure, the dependency edge (*e1*) between the first conjunct (*v1*) and the last conjunct (head conjunct (*v2*)) is extracted and changed to the `COORD` label. In the second step, the head node (*v3*) of the head conjunct (*v2*) is propagated to be the head node the first conjunct (*v1*) by setting a new edge between the *v1* and *v3*. This procedure is repeated for all other conjuncts (except the head conjunct). The incidence list before restructuring of coordinations (Figure 4.10) is *E*{*(Il-2,Il-4),(Il-4,activation)*}, and the incidence list after the pruning procedure (Figure 4.11) is *E*{*(Il-2,Il-4),(Il-4,activation), (Il-2,activation)*}.

Restructuring of noun phrases in the *np action* procedure: Finally, I introduce the restructuring of noun phrases that contain action adjectives. The original dependency representation of the noun phrase selects the noun furthest to the right as the head of the NP and thus all remaining elements are its dependents. For the noun phrases containing action adjectives (mostly verb derivations), this representation does not reflect the true semantic relations between the elements. For example, in "*IL-2 specific activation*" it is "*IL-2*" that is responsible for the activation. Therefore, I restructure the dependency graph by changing the head of "*specific*" from "*activation*" to "*Il-2*". The re-coding rules first select all the noun phrases containing action adjectives ending in "*-ed*", "*-ing*", "*-ible*" suffixes and with words such as "*dependent*", "*specific*", "*like*", etc. In the second step, the noun phrase is restructured by encoding the adjective as the head of all the nouns preceding this adjective in the noun phrase under scrutiny (Figures 4.12 and 4.13). The restructuring of noun phrases can be represented with pseudocode as follows:

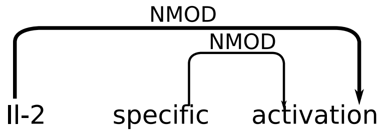


Figure 4.12: Noun phrase representation in CoNLL dependency trees.

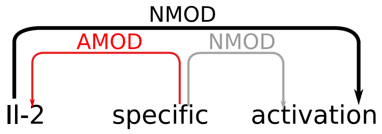


Figure 4.13: Trimming procedure for *noun phrases* on CoNLL dependency trees.

```
if (np contains actionAdjective)
do{
  token npHead = getPhraseHead(np)
  node v1 = getNode(token)
  e [] edges = v1.getEdges
  for (int i = 0; i < edges.length(); i++)
    if (e.getModifier() contains actionAdjective){
      v2 = e.getModifier()
      deleteEdge(e)
     }
    if(firstModifier(e.getModifier)){
      v3 = e.getModifier();
      newEdge(v2, v3, "AMOD")
      break
    }
}
```

This code can be interpreted by considering dependency-based and constituency-based parse trees. If a noun phrase contains action adjectives (see above), the head of a noun phrase (*v1*) is extracted (the head of a noun phrase is the token that has a dependency head outside the noun phrase under consideration). The dependency edge between the head (*v1*) and the token in focus (action adjective (*v2*)) is then deleted (see *deleteEdge(e)*) and a new dependency edge with label `AMOD` is created between this token (*v2*) and the noun in the noun phrase occurring in the first position (*v3*), as illustrated in Figure 4.13. The incidence list before restructuring the noun phrase (Figure 4.12) is *E*{*(Il-2,activation),(specific,activation)*}, and the incidence list after the pruning procedure (Figure 4.13) is *E*{*(Il-2,activation), (specific,Il-2)*}.

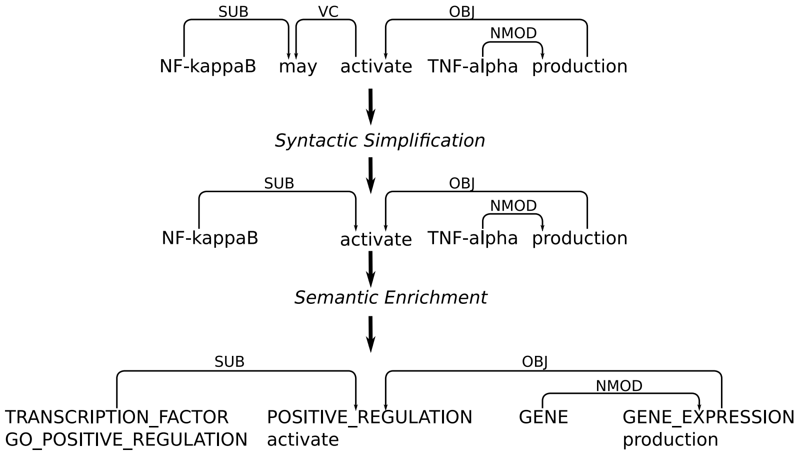


Figure 4.14: Trimming of dependency graphs.

In summary, eliminating syntactic dependency information should be seen as pruning apparently noisy information from the original dependency graph and zooming in on really relevant dependency information. Reformatting dependency graphs in this way avoids exposing the classifiers in their learning phase to distracting structural noise, such as the limited or even total lack of accessibility in the dependency graph, or the interference of spurious dependency nodes or dependency relations. For an evaluation of syntactic simplification see Section 5.3. The four procedures presented are applied in my event extraction approach for all CD dependency graphs. Figure 4.14 shows an example of syntactic trimming required for the sentence "*NF-kappaB may activate TNF-alpha production.*". The modal verb "*may*" is pruned from the dependency tree and the `SBJ` relation to the node "*NF-kappaB*" is propagated to the main verb "*activate*". Whereas the syntactic trimming procedures presented operate on dependency edges, semantic enrichment considers words in dependency nodes and will be presented in the following section.

Semantic Enrichment

Lexical nodes in the dependency graphs which have not been pruned and which are deemed to be important (see below) for argument extraction are then enriched or even substituted with semantic class annotations, instead of keeping only the original lexical representation (Figure 4.14). The rationale behind this decision is to generate more discriminative kernel- and feature-based representations (Sections 4.6.2 and 4.6.3).

The conceptual enrichment process is based on a five-tier task-specific semantic hierarchy of named entity classes which are defined as crucial for the molecular event extraction task in this work.

The first layer (*Entity Layer*) consists of ontological concepts from the Gene Regulation Ontology (GRO) (cf. Section 3.5.7), e.g., for genes, by the concept class `Gene,` and by the equivalent classes `Transcription Factor`, `Binding Site`, and `Promoter` for these entities so providing highly specific information about genes and proteins. The ontological definitions of these concepts can be found in GRO. Whenever a lexical item is categorized into one of these entities, the associated node in the dependency graph is overlaid with that category information, applying the ranking in cases of conflicts. The concepts such as `Transcription factor`, `Binding site`, and `Promoter` have a higher priority than the concept `Gene`. Thus, if a token is annotated as both, *viz.* transcription factor and gene, the annotation as a transcription factor is preferred. The transcription factors, binding sites and promoters can be detected using NER taggers developed in JCore (Hahn et al., 2008), for example. Gene annotation can be provided by the named entity tagger/gene name normalizer GeNo (Wermter et al. (2009)) with a direct mapping to the Uniprot database.[23] This database is itself used for enriching informations from wet lab experiments in the form of Gene Ontology Annotations (GOA).[24] For this purpose, I first categorized Gene Ontology (cf. Section 3.5.7) terms[25] from both the 'molecular function' and from the 'biological process' branch with respect to their matching molecular event type, e.g., `Phosphorylation` or `Positive Regulation`. Then I mapped all gene name mentions which occurred in the text to their UniProt identifiers (UniProt Consortium (2008)) using GeNo. UniProt identifiers link a gene with a set of (curated) GOA annotations. The GOA annotations constitute the second layer (*Gene Ontology Layer*) of semantic enrichment.

The third layer (*Event Trigger Layer*) contains annotations of event trigger types of all events considered, and is based on the collected dictionaries presented in Section 4.4.1.

The fourth layer (*MeSH Layer*) assembles Medical Subject Headings (MeSH) terms. MeSH[26] is one of the most important biomedical terminologies, maintained by the NLM (National Library of Medicine), which is used for the curation and indexing of, for instance, the Medline literature database. MeSH is hierarchically organized and consists of 26,000 inter-linked descriptors with 177,000 entry terms (named

[23] `http://www.ebi.uniprot.org/index.shtml/`

[24] `http://www.ebi.ac.uk/GOA`

[25] `http://www.geneontology.org/`

[26] `http://www.nlm.nih.gov/mesh/`

synonyms).[27] As MeSH aims to collect the names of entities used in the biomedical domain, it is important for extraction of molecular events.

Finally, the fifth layer (*Methods Layer*) assembles names for experimental methods. The corresponding dictionary is collected from the Genia event corpus. One student of biology sorted the experimental methods in relation to the event categories under scrutiny. For example,"*affinity chromatography*" was assigned both to the `Gene Expression` and to the `Binding` category. The experimental methods dictionary currently contains more than 500 entries. I only included those Genia annotations and experimental methods which matched the event types to be identified in a sentence. I added to the dependency nodes semantic information in terms of the experimental method category and a corresponding Genia event annotation concept.

After conceptual overlaying, the original dependency graph is transformed into a semantically decorated one (Figure 4.14). In this figure, the lexeme "*NF-kappaB*" is replaced by `Transcription Factor` and the GO annotation for `Positive Regulation`, while "*TNF-alpha*" is only substituted by `Gene` since no additional information is available for this gene. Finally, the event triggers "*activate*" and "*production*" are enriched by their associated event types, `Positive Regulation` and `Gene Expression`, respectively.

In summary, various sources of semantic knowledge are applied for semantic enrichment. These are the Gene Ontology as the most important ontology for molecular biology, the MeSH terminology as the largest biomedical terminology source, the Uniprot and GOA databases, and also dictionaries collected for this thesis for experimental methods and event triggers assembled from the Genia event corpus, the key corpus for molecular events.

After the completion of the event trigger detection and disambiguation step and the trimming of dependency graphs, the argument identification component can start its procedures.

4.6 Argument Identification and Ordering

Argument identification is the main subtask in the event detection process. In this subtask the event arguments and their roles are identified. The distinction between the trigger-driven and entity-driven event extraction comes to the fore in this task. In the trigger-driven event extraction, sentence-wise pairs of putative triggers and their putative argument(s) are built, the latter involving ontological information about

[27] MeSH Fact Sheet 2011 (`http://www.nlm.nih.gov/pubs/factsheets/mesh.html/`).

the event type (e.g., `Protein`). In the entity-driven approach, sentence-wise pairs of entities are built and considered to be arguments in putative eventive relations.

4.6.1 Supervised Argument Detection

Argument identification can be considered as a classification problem in which argument $x \in \mathcal{X}$ is mapped onto a relation label $y \in \mathcal{Y}$, where $\mathcal{X}$ is a set of input examples and $\mathcal{Y}$ is a finite label set. A large set of (x_i, y_i) pairs for all i is called training data. The training data can be used to learn mappings from $\mathcal{X}$ to $\mathcal{Y}$ or, in other words, to create a function that takes an input example x and generates an output y. The goal is to learn to output a correct y for unseen x examples, which means that the model generalizes over training data. This learning approach is called a *supervised* learning approach. As the supervised classification learning is common for tasks where classification labels and feature types can be determined for the classification data, for example in diverse information extraction scenarios for named entity and relation extraction tasks, it is used in this thesis.

In a supervised learning approach, original examples are usually pre-processed and represented by a set of features in the form of feature vectors. The transformation from raw input data to feature vector representation is called *feature extraction.* Feature types are first determined by a system developer and can later be ranked by a model at a feature selection stage.

Supervised learning models usually require the specification of feature functions $f_i : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, with

$$f_i(x, y) = \begin{cases} 1, & \text{if } x = x_i \text{ and } y = y_i \\ 0, & \text{else} \end{cases} \tag{4.6}$$

where $i = 1, ..., n$ and n is a size of training data with training instances x_i and corresponding label values y_i.

The supervised learning approach is fairly common for training statistical probability models or Support Vector Machines (SVMs) models. Probabilistic approaches such as Maximum Entropy (MaxEnt) or Conditional Random Fields (CRF) build a probability model which provides a probability of association of an instance to a label class. The SVMs classify unseen examples using a hyperplane separating two data classes represented as points in space. Both methods, statistical probability model in form of MaxEnt modeling and SVMs, are applied in this thesis for argument identification.

Maximum Entropy

Maximum Entropy modeling is frequently used for classification decisions in a supervised approach. The rationale behind MaxEnt is to choose a model for any collection of facts, which is consistent with all the facts but otherwise as uniform as possible. The entropy is the measure for the uniformity of an MaxEnt model. The argument classification problem can be rephrased in terms of statistical modeling as prediction of the behavior of a random process which provides an output $y \in \mathcal{Y}$ taking into account element $x \in \mathcal{X}$. $p(y|x)$ is a conditional probability for y given an element x. The model P is the set of all conditional probability distributions. To construct such a model we use a sample set of size n of output values with corresponding elements, the training data. The training data is used to create feature functions as presented in (4.6). Below I describe the MaxEnt presentation by Berger et al. (1996) for use in NLP applications:

The statistics that the model creator is interested in is the expected value of f with respect to the empirical distribution of the training sample $\tilde{p}(x, y)$ defined as

$$\tilde{p}(f) = \sum_{x,y} \tilde{p}(x, y) f(x, y) \tag{4.7}$$

The MaxEnt model p should accord with these statistics, thus it is contained in the set $C \subseteq P$ defined by

$$C = \{p \in P | p(f_i) = \tilde{p}(f_i) \text{ for } i \in \{1, ..., n\}\} \tag{4.8}$$

where $p(f_i)$ is the expected value of f_i with respect to the model $p(y|x)$ and n the number of f_i. With respect to the maximum entropy philosophy we select the model $p \in C$, which is the most uniform. Given that the measure of uniformity of a conditional distribution is provided by the conditional entropy H, our model p is defined by

$$p_\star = \underset{p \in C}{\operatorname{argmax}} \, H(p) \tag{4.9}$$

The MaxEnt approach is precisely based on this simple principle for modeling all that is known and assuming nothing about that which is unknown. MaxEnt modeling is successfully applied to solve a range of NLP problems, e.g., sentence splitting, tokenization, POS tagging, parsing, named entity recognition, and argument classification. A more detailed description of MaxEnt modeling for NLP applications is given in the introduction to MaxEnt models by Berger et al. (1996).

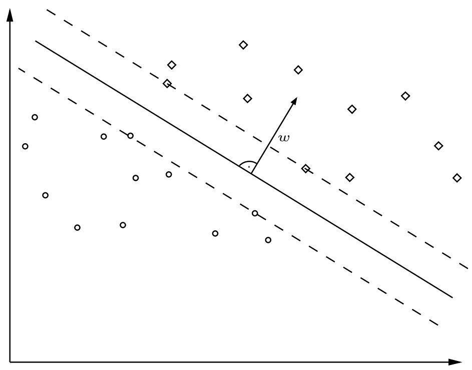


Figure 4.15: Hyperplane for a linear separation of two classes. The figure is taken from (Faessler (2009), p. 18).

Support Vector Machines

Another classification model applied for NLP are Support Vector Machines (SVM). The main purpose of an SVM is to construct a hyperplane[28] $\mathcal{H}$ separating two data classes represented as data points x in space $\mathbb{R}^k$ where k is the dimension of the feature size. In the training phase, the goal is to draw a hyperplane with the largest margin, i.e., to achieve the largest distance to the nearest data points of both classes. This can be formally defined as follows:

$$\max_{w \in \mathbb{R}^k, b \in \mathbb{R}} \min \{ \|x - x_i\| \mid x \in \mathbb{R}^k, \langle w, x \rangle + b = 0, i = 1, \ldots, n \} \tag{4.10}$$

where w is a normal on a hyperplane (Figure 4.15) and b is a shift of a hyperplane. The condition $\langle w, x \rangle + b = 0$ for $x \in \mathbb{R}^k$ describes all data points x, which are located exactly on the hyperplane $\mathcal{H}$.

Unseen examples are then mapped into the space $\mathbb{R}^k$ and detected as belonging to a class based on their positions relative to a hyperplane. The SVM classification can be formulated as follows:

$$y = \operatorname{sgn}(\langle x, w \rangle + b) \tag{4.11}$$

In (4.11) y takes values of 1 or -1 which are representative for class labels.

[28] Hyperplane is a concept from a geometry. A hyperplane separates an n-dimensional vector space and has a dimension of $n - 1$.

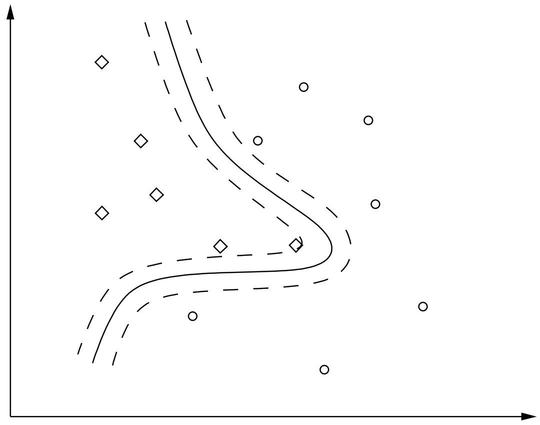


Figure 4.16: Illustration of a non-linear separation function in space $\mathbb{R}^k$. The figure is taken from (Faessler (2009), p. 23).

Figure 4.15 illustrates a linear separation of data points. Usually, the linear separation is not possible due the complexity of high-dimensional classification data. In such case, a non-linear separation function such as a polynomial is required (Figure 4.16). As an SVM is a linear binary classifier, it needs a higher dimensional space $\mathbb{H}$ which represents the data points of the original space $\mathbb{R}^k$ in order to perform a linear separation. The SVM solution does this by exploiting *kernel* functions. Kernels are functions that calculate scalar products of vectors in high-dimensional spaces. A kernel function can be described formally as follows:

$$K(x, y) = \langle \phi x, \phi y \rangle = \sum_i (\phi x)_i (\phi y)_i, \quad \forall x, y \in \mathbb{R}^k$$

where K is a kernel function and ϕx and ϕy are vectors represented in a higher dimensional space $\mathbb{H}$ using the function ϕ (cf. Schölkopf and Smola (2001)). $\langle \phi x, \phi y \rangle$ denotes the scalar product of these vectors.

For a more detailed description of SVMs and kernels, see the introduction to SVMs by Schölkopf and Smola (2001). The kernels are crucial for the classification of NLP data and various kernels have been applied for relation detection, e.g., a shallow parsing kernel (Zelenko et al., 2003), dependency tree kernels (Culotta and Sorensen, 2004), a dependency path kernel (Bunescu and Mooney, 2007), and a graph kernel (Airola et al., 2008a). The graph kernel was shown to perform very well on relational biomedical data. Therefore, it is applied for argument identification in this thesis (see Section 4.6.3).

4.6.2 Feature-based Classifier

Feature-based approaches for solving the relation extraction task are very popular in NLP research. Diverse sources of knowledge have been exploited and studied for their effect on extracting relations, e.g., lexical information and shallow and full parsing results (cf. e.g., CoNLL, ACE and SemEval challenge series). In this thesis, I integrate feature types already classified as being highly useful for solving the relation task. My choice is for various shallow features such as word, chunk and entity information, as extensively evaluated by Zhou and Zhang (2007), and my second choice is for feature types modeling dependency parse results as proposed by Katrenko and Adriaans (2006) and Kim et al. (2008b). The system of Zhou and Zhang (2007) achieved highly competitive results of 74.7% in relation mention extraction on the ACE corpus from the newspaper domain. The feature-based systems of Katrenko and Adriaans (2006) and Kim et al. (2008b), considering only dependency tree information, achieved excellent results on biomedical corpora such as the LLL corpus. Katrenko and Adriaans (2006) report on experiments in which they achieved 72.4% F-score on the LLL corpus. Kim et al. (2008b) report having reached F-score peaks of 77.5% on the LLL corpus.

In general, I distinguish three groups of features in JReX, e.g., *lexical and semantic type features*, *chunking features* and *dependency parse features*. Below, I present the selected feature types in more detail. I distinguish between two semantic object mentions (e.g., named entities or event triggers) in pairwise relations, i.e., mention 1 (M1) and mention 2 (M2). M1 is the semantic object mention that occurs first in the sentence (before M2). If one of the mentions includes another semantic object mention, then the semantic object mention with a larger span is classified as M1.

Lexical and Semantic Type Features: This feature class covers lexical items before, after and between semantic object mentions and their semantic types, as described by Zhou and Zhang (2007) (Table 4.4). The window for the words before M1 and after M2 has a size of two words. Semantic type features account for combinations of semantic types, with flags indicating whether mentions have an overlap.

Chunking Features: The chunking features are concerned with the chunks of both instance mentions, between, before and after these instances (Table 4.5). The head of a chunk should be interpreted as follows. The head of a chunk is selected only if a chunk contains more than one token. The token furthest right is selected as a head. For example in a noun chunk "*Il-2 protein*", "*protein*" is its head.

Table 4.4: *Lexical and Semantic Type Features* used in the argument extraction component.

Feature Name	Feature Description
WM1	bag of words in M1
HM1	head word of M1
WM2	bag of words in M2
HM2	head word in M2
HM12	combination of HM1 and HM2
WBNULL	when no word in between
WBFL	when only one word in between
WBF	first word in between when at least two words in between
WBL	last word in between when at least two words in between
WBO	other words in between except first and last words when at least three words in between
BM1F	first word before M1
BM1L	second word before M1
AM2F	first word after M2
AM2L	second word after M2
MT12	combination of semantic object mention types
MB	number of other mentions in between
WB	number of words in between
M1/M2 or M1/M2	flag indicating whether M1 includes M2 or vice versa
MT12 + M1/M2	feature combination of MT12 and M1/M2
MT12 + M2/M1	feature combination of MT12 and M2/M1
HM12 + M1/M2	feature combination of HM12 and M1/M2
HM12 + M2/M1	feature combination of HM12 and M2/M1

Dependency Parse Features: The dependency parse features are crucial for relation extraction and in particular for argument identification in molecular events. Two general categories of dependency features are integrated in my event extraction approach. The first category provides information about dependency tree level positions and the second category contains information about nodes on the shortest path between semantic objects in potential relation. The first feature category is thoroughly explored in the work of Katrenko and Adriaans (2006), whereas the second is extensively studied by Kim et al. (2008b).

The dependency tree levels are modeled for local dependency tree information by

Table 4.5: *Chunking Features* used in the argument extraction component.

Feature Name	Feature Description
CPHBNULL	when no chunk in between
CPHBFL	the only chunk head when only one chunk in between
CPHBF	first chunk head in between when at least two chunks in between
CPHBL	last chunk head in between when at least two chunk heads in between
CPHBO	other chunk heads in between except first and last chunk heads when at least three chunks in between
CPHBM1F	first chunk head before M1
CPHBM1L	second chunk head before M1
CPHBM2F	first chunk head after M1
CPHBM2L	second chunk head after M1
CPP	path of chunk labels between the two mentions

means of parents and children (dependency nodes) of semantic object mentions (their dependency nodes) and global dependency tree information, by means of the first subsuming dependency node between both semantic object mentions, called LCS (*least common subsumer*). The LCS node is defined as follows: "Given two nodes A and B in a dependency tree T, a least common subsumer LCS(A;B) is a node L, such that L is ancestor for both A and B, and there exists no other node N being an ancestor for A and B, such that L is ancestor of N. There is exactly one LCS for any two nodes in a dependency tree." (cf. Katrenko and Adriaans (2006), p. 64). The parent and children dependency nodes are represented by word and dependency edge label adjacent to the mention and the dependency parent or child considered. The introduced dependency tree levels are categorized by Katrenko and Adriaans (2006) as the most important for relational learning.

In the following, I illustrate the extraction of dependency tree level features for the entity-driven event extraction. Figure 4.17 illustrates the dependency tree levels of Katrenko and Adriaans (2006) for the sentence "*Cdc25 can be activated in vitro in a Raf1-dependent manner*". The extracted features for the potential eventive relation between the entity mentions "*Cdc25*" and "*Raf1*" are represented in Table 4.7. For the node "*Cdc25*" the parent P "*activated*" with an edge label s is extracted, for the node "*Raf1*" the parent P "*dependent*" with an edge label *lexmod* is extracted.

Table 4.6: *Dependency Tree Level Features* used in the argument extraction component.

Feature Name	Feature Description
PM1	parent of the head of M1
PM2	parent of the head of M2
C1M1	first child of the head of M1
C2M1	second child of the head of M1
C1M2	first child of the head of M2
C2M2	first child of the head of M2
LCS	LCS node of M1 and M2 nodes

Table 4.7: Examples of *Dependency Tree Level Features* for the dependency tree from Figure 4.17. The words are lemmatized. The dependency edge labels adjacent to parent and children nodes are connected to a lemma with the help of an underscore.

Feature	*Cdc25*	*Raf1*
C^1	-	-
C^1	-	-
P	activate_s	dependent_lexmod
LCS	activate	activate

Both nodes "*Cdc25*" and "*Raf1*" have no children dependency nodes. Their LCS is the node "*activated*". This example illustrates that entity mentions in relation usually have an LCS node that normally contains a preposition or a verb such as "*activate*", "*bind*", etc. (cf. Table 4.8). Thus, the LCS feature seems suitable for extraction of eventive relations and predicate-argument relations. The LCS and parent node features were evaluated as being the best-performing dependency level features in relational learning. Therefore, in my approach, dependency level features are currently switched on for these types only.[29]

The second dependency feature category comprises dependency path features. The dependency path feature approach is based on the main idea introduced by Bunescu and Mooney (2007), called *shortest path hypothesis*. This hypothesis assumes that the most relevant information for a relational task is located on the shortest path between semantic objects in potential semantic relation. This is because important predicates such as verbs or prepositions are usually located on the shortest path be-

[29] The children node features are still implemented and can be applied if necessary.

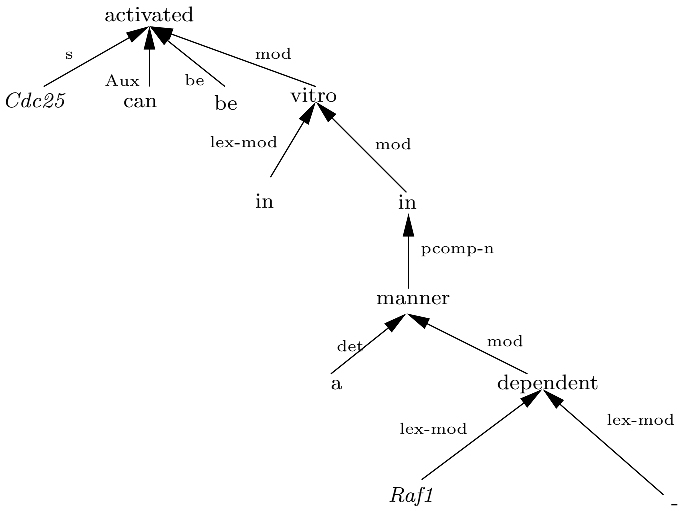


Figure 4.17: Example sentence for the extraction of dependency tree level features. The sentence "*Cdc25 can be activated in vitro in a Raf1-dependent manner*" from the AIMED corpus. The potential arguments are *Cdc25* und *Raf1*. The redrawing by Faessler (2009), taken from Katrenko and Adriaans (2006).

Table 4.8: Least common subsumer (LCS) frequency on the AIMED data (Bunescu et al., 2005). The table is taken from Katrenko and Adriaans (2006).

Words (LCS)	Occurrence
of	434
...	
bind	139
interact	134
complex	59
inhibit	52
show	49
...	
regulate	25
suppress	14

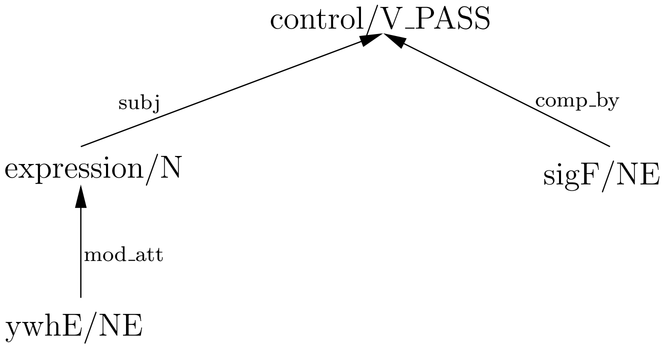


Figure 4.18: Shortest dependency path example from Kim et al. (2008b). The shortest dependency path between named entities *ywhE* and *sigF* from a sentence "*Analysis of the expression of a translational ywhE-lacZ fusion showed that ywhE expression is sporulation-specific, and is controlled predominantly by the forespore-specific sigma factor sigma(F), and to a lesser extent by sigma(G).*" extracted from the LLL corpus. The redrawing by Faessler (2009), taken from Kim et al. (2008b).

tween entities in relation (cf. LCS node of Katrenko and Adriaans (2006)). Bunescu and Mooney (2007) were the first to apply shortest dependency path information for extraction of semantic relations between named entities. Stevenson and Greenwood (2009) compare various models for pattern generation on dependency trees, e.g., predicate-argument, chain and shortest path models. The shortest path model was found to be one of the best models for representing relations between named entities (cf. Stevenson and Greenwood (2009) for detailed explanations of models). The authors conclude that the shortest path contains the most relevant information for extraction of semantic relations. The shortest path is illustrated in Figure 4.18.

For modeling the structural information on the shortest dependency path, Kim et al. (2008b) propose to create *walks*. The walks are short fractions of a dependency graph. If a dependency graph is $D = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ are vertices and $E = \{e_1, \ldots, e_n\}$ are edges of this graph, the path in D is defined as alternating sequences of vertices and edges, $v_i, e_{i,i+1}, v_{i+1}, e_{i+1,i+2}, \ldots, v_{i+n-1}$, and the path starts and ends with a vertex. The walk of length 1 is defined as a fraction of a path, namely a sequence of vertex, adjacent edge and vertex, or $v_i, e_{i,i+1}, v_{i+1}$. These kind of walks are called v-walks as they start and end with a vertex. Kim et al. (2008b) additionally introduce the e-walks that start and end with an edge, namely $e_{i,i+1}, v_{i+1}, e_{i+1,i+2}$. For the v-walks and e-walks, Kim et al. (2008b) extract lexical and morpho-syntactic information from dependency nodes. Thus, while a

Table 4.9: *Walk Features* for the shortest path in Figure 4.18. The table is taken from Kim et al. (2008b).

Lexical Walks	NE+mod_att(UP)+expression, mod_att(UP)+expression+subj(UP), expression+subj(UP)+control, subj(UP)+control+comp_by(DN), control+comp_by(DN)+NE
Syntactic Walks	NE+mod_att(UP)+N, mod_att(UP)+N+subj(UP), N+subj(UP)+V_PASS_PRED, subj(UP)+V_PASS_PRED+comp_by(DN), V_PASS_PRED+comp_by(DN)+NE

lexical walk contains stem or lemma forms of words from vertices plus dependency edge labels, a syntactic walk contains POS tags of words from vertices plus dependency edge labels. The nodes which contain named entities, usually gene names, are replaced by an NE tag. Both walks add to dependency edge labels an additional label for a walk direction, e.g., UP or DOWN in a dependency graph, and the least common subsumer node is attached the label PRED (LCS node of Katrenko and Adriaans (2006)). This node is usually one where the direction may change, as presented in Figure 4.18, where the walk direction from *ywhE* to the node *control* is an up direction while the walk from *control* to the *sigF* is a down direction. It is possible to use the shortest path in either direction as there is no restriction. Table 4.9 exemplifies walk features extracted for the shortest path from Figure 4.18.

I assemble all feature types presented in this section for solving the event extraction task.

4.6.3 Graph Kernel Classifier

Airola et al. (2008a) present a graph kernel classifier for PPI extraction that is based on mathematical operations on an adjacency matrix. The graph kernel had already been presented for other application domains. The graph kernel uses a converted form of dependency graphs in which each dependency node is represented by a set of labels associated with that node. The dependency edges are also represented as nodes in the new graph, such that they are connected to the adjacent nodes in the dependency graph. Subgraphs which represent, e.g., the linear order of the words in the sentence can be added, if required. Airola et al. (2008a) build two subgraphs, where the first subgraph reflects the dependency structure of the underlying sentence and the second subgraph represents the linear order of the words in the sentence (Figure 4.19). Protein names are replaced with a placeholder `PROT'X`. Furthermore, the edges in a created graph are assigned weights. Airola et al. (2008a) chose a simple weighting scheme (determined in experiments on the large PPI corpus AIMed) where all edges of the shortest dependency path receive a weight of 0.9 and all other edges receive a weight of 0.3. Thus, the relevance of the shortest path is represented through graph kernel parameters. The entire graph is represented in terms of an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ where V is the set of vertices. The rows and columns of the adjacency matrix are indexed by the vertices and $A_{i,j}$ contains the weight of the edge adjacent to both vertices $v_i \in V$ and $v_j \in V$. The adjacency matrix is further processed to contain the summed weights of paths connecting two nodes of the graph. The set of possible labels L is a set of possible labels that a vertex can take. All labels are represented as a label allocation matrix $L \in \mathbb{R}^{|L|\times|V|}$ so that $L_{i,j} = 1$ if the j-th vertex has the i-th label but otherwise takes $L_{i,j} = 0$. After calculating the sum of the weights for all paths in the training data connecting two vertices, Airola et al. (2008a) form a new adjacency matrix W with these summed weights. The instance G that represents a candidate interaction is defined as $G = LWL^T$ where L is the label allocation matrix and W is the final adjacency matrix with summed weights between vertices. The graph kernel is defined as

$$K(G^1, G^2) = \sum_{i=1}^{|\mathcal{L}|}\sum_{j=1}^{|\mathcal{L}|}[G^1]_{ij} \cdot [G^2]_{ij} \tag{4.12}$$

where G^1 and G^2 are two instances formed as introduced above.[30] The grade of the similarity between G_1 and G_2 is represented by the sum of the connection strengths between pairs of vertices where the values $[G^1]_{ij} \cdot [G^2]_{ij}$ always belong to the same pair of vertices. For more details cf. Airola et al. (2008b).

For my experiments, I investigated some variants of the original graph kernel. In the original version each dependency graph edge is represented as a node. This means that connections between *token graph nodes* are expressed through *dependency edge graph nodes* (Figure 4.20; (1)). To represent the connections between original tokens as direct connections in the graph, I removed the edge graph nodes and each token was assigned the edge label (its dependency label; Figure 4.20; (2)). Further variants included encodings for: first, the shortest dependency path (*sp*) between two

[30] Please note that numbers denote here an index rather than an exponent.

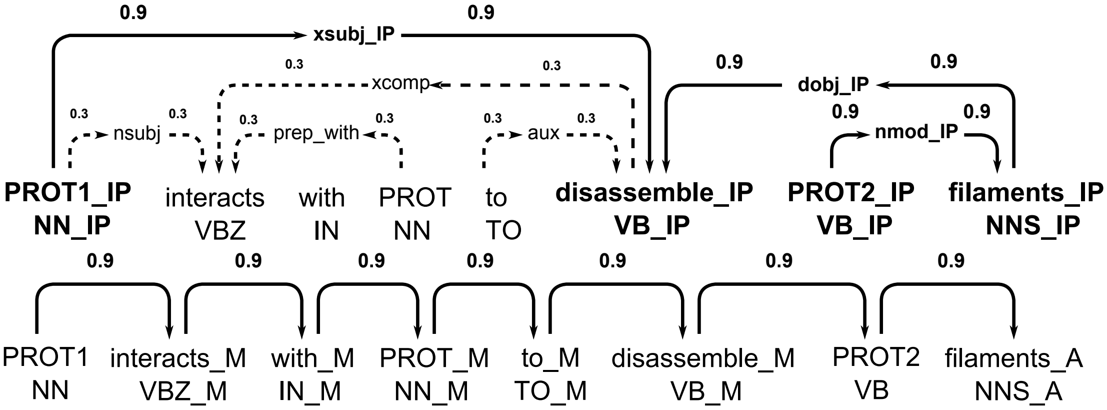


Figure 4.19: Graph representation generated for a sample sentence. The candidate pair in relation is marked as PROT1 and PROT2, the third protein is marked as PROT. The shortest path between the proteins is shown in bold. In the dependency based subgraph all nodes in a shortest path are specialized using a post-tag (IP). In the linear order subgraph possible tags are (B)efore, (M)iddle, and (A)fter. For the other two candidate pairs in the sentence, graphs with the same structure but different weights and labels would be generated. The redrawing by Faessler (2009), taken from Airola et al. (2008a).

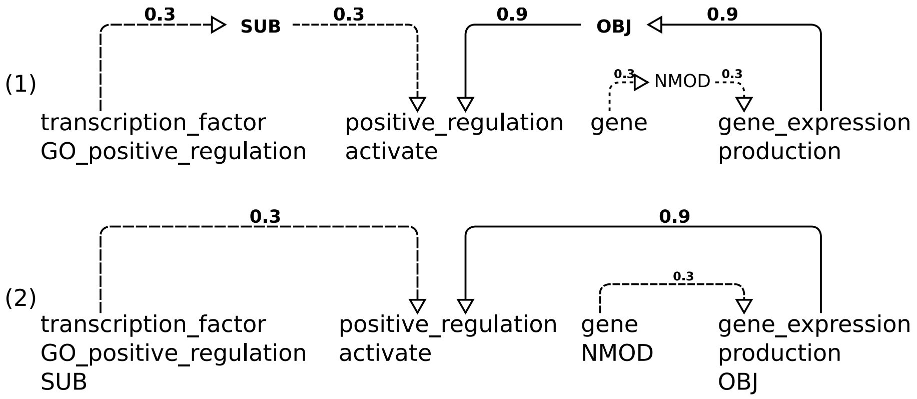


Figure 4.20: Graph Kernel representation for a trimmed dependency graph — (1) original representation (*with dependency-edge-nodes*), (2) representation without dependency edge graph nodes (*without dependency-edge-nodes*) (weights {0.9, 0.3} are taken from Airola et al. (2008a)).

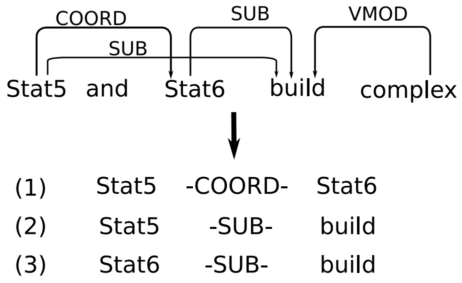


Figure 4.21: Three shortest dependency paths for `Binding` event (`Binding`(*complex*, `Theme`:Stat5, `Theme2`:Stat6)).

mentions (argument and trigger); second, the complete dependency graph (*sp-dep*); and third, the complete dependency graph and linear information (*sp-dep-lin*) (the original configuration from Airola et al. (2008a)). All tokens that are annotated with selected semantic information are replaced by respective semantic labels for further processing (Figure 4.20, steps (1), (2)). The linear information considered for molecular events now takes into account all words in the window of size 2 before and after the target units to relate and all the words between these units, and not the complete sentence as proposed by Airola et al. (2008a). My preliminary experiments with the linear graph revealed that the token window of this size was the best configuration on Shared Task training data (Buyko et al., 2011a). Furthermore, I allowed only *one* shortest dependency path to be modeled in the graph kernel, namely the path between the heads of semantic object mentions (e.g., named entities and triggers). In their initial version, Airola et al. (2008a) allowed all shortest paths between all tokens of putative triggers and their arguments to be modeled in the graph kernel. This does not lead to an increased performance for event extraction as detected in my preliminary experiments (Buyko et al., 2011a).

For detecting events with n `Theme` arguments where $n > 1$, (e.g., binary `Binding` events) graph kernel also allows to model the shortest paths between the n arguments in addition to modeling the shortest paths between the trigger and arguments only (Figure 4.21). Thus, the graph kernel exploits, for such events, multiple shortest paths extracted from the dependency graph.

4.7 Event Extraction Pipeline – The JReX System

The presented event extraction approach was implemented in the Jena Relation eXtraction (JReX) system, which constitutes a major part of this thesis work. The JReX system disposes of a pipeline architecture. The pipeline consists of two major parts, the text pre-processor and the dedicated event extractor. As far as pre-processing is concerned, the JReX pipeline contains a range of text analysis tools adapted by Buyko et al. (2006) to the biomedical domain by re-training on the Genia and PennBioIE treebank corpora. The pipeline starts with sentence splitting, tokenization and POS tagging with OpenNLP and JCore (Hahn et al., 2008) tools. The JReX system integrates two morphological analysis approaches, i.e., stemming as produced by Porter stemmer and lemmatization algorithms from Specialist Lexicon tools. The shallow syntactic analysis is performed with the OpenNLP chunker adapted by Buyko et al. (2006) to the biomedical domain. Dependency parsing is performed with an adapted dependency parser from a collection of freely available dependency parsers such as GDep and MST parser. The named entity recognition part uses GeNo (Wermter et al. (2009)), gene name tagger and normalizer, which achieved a top-rank performance of 86.4% on the gene normalization task of BioCreAtIvE II with a mapping to the Uniprot database. In addition, a number of regex- and dictionary-based entity taggers (covering promoters, binding sites, and transcription factors) are applied. The MeSH terms and experimental method terms can be annotated in text with a dictionary-based approach using, for example, Lingpipe Dictionary Chunker.[31]

As far as event extraction is concerned, JReX starts with trigger word identification using the Lingpipe Dictionary Chunker and collected dictionaries (Section 4.4.1), and performs trigger disambiguation (Section 4.4.2) in the trigger-driven extraction approach. The trimming of dependency parse results is executed in the next analysis step. In this work I apply the *jgrapht* package[32] used for performing operations on graphs. The argument extraction is then performed, either on pairs of named entities or on pairs of triggers and named entities (Section 4.6). The Maximum Entropy (MaxEnt) classifier from the Mallet package (McCallum, 2002) was selected to implement the feature-based approach, which integrated all feature types presented in Section 4.6.2. An MaxEnt model can successfully deal with a wide range of features (Jurafsky and Martin, 2009). This makes an additional feature selection process to a great extent obsolete, as feature weighting is integrated in the MaxEnt modeling (cf. Manning and Schütze (1999)). For the graph kernel approach, I chose the LibSVM (Chang and Lin (2001)) Support Vector Machine as classifier.

[31] http://alias-i.com/lingpipe/

[32] http://www.jgrapht.org/javadoc/overview-summary.html/

The MaxEnt and SVM models are even applied in an *ensemble* configuration, i.e., results are merged by considering each example as a positive event that had been classified as a positive event by either the SVM or the MaxEnt model. After the completion of these steps, JReX creates event instances. During the event creation step, an event instance is created based on the predicate-argument relations extracted in the trigger-driven approach or on the eventive propositional relations extracted in the entity-driven approach. The creation of an event instance in the entity-driven approach is accomplished by building an event instance with an argument structure which corresponds exactly to the argument structure of eventive relations detected. In contrast, for the trigger-driven approach, the event creation step can vary. It depends on the argument structure defined for events (e.g., argument types and numbers) (cf. Section 5.2.1 for event creation).

The JReX system is fully implemented in Java and integrated in the *Unstructured Information Management Architecture* (UIMA) software framework (Götz and Suhre, 2004). UIMA is an Apache-licensed open source platform for unstructured information management solutions, in particular the natural language processing applications. JReX system consists of a range of components (including the JReX argument extraction component) integrated in the UIMA framework, assembled together into a pipeline for event extraction. The JReX component for argument extraction is a so called UIMA *analysis engine* which adds annotations to the analysed data called CAS (Common Analysis Structure) in this framework. The data structure backbone of the JReX system is a comprehensive annotation type system (Hahn et al. (2007a), Buyko and Hahn (2008)), which covers major annotation layers for the NLP processing, e.g., text segmentation, syntactic and semantic analysis. The integration of the JReX machinery in the UIMA framework as an UIMA pipeline allows to re-assemble and modify event extraction pipelines easily and flexibly by plugging-in or plugging-out NLP modules (e.g., tokenizers, parsers).

Chapter 5

Evaluation of JReX in "BioNLP 2009 Shared Task on Event Extraction"

This chapter provides evaluations of the Jena Relation eXtraction (JReX) system on the "BioNLP 2009 Shared Task on Event Extraction" data. The official data sets of this challenge allow a good comparison of event extraction systems and therefore enable us to identify the state-of-the-art performance for event extraction in the biomedical domain. I first describe Task 1 from the "BioNLP 2009 Shared Task on Event Extraction" (referred to as Shared Task in this chapter), presenting data and resources available to Shared Task participants (Section 5.1). Section 5.2 gives official run results for the JReX system, followed by the results of the updated JReX system and in-depth evaluations of various JReX configurations. The overall description of the JReX performance is extended in this chapter with a detailed evaluation of a range of dependency representation formats and with a discussion of the impact of syntactic and semantic trimming of dependency graphs on the effectiveness of event extraction (Section 5.3). This chapter is concluded with an in-depth error analysis and discussion about learning progress for the argument detection task (Section 5.4).

5.1 BioNLP 2009 Shared Task on Event Extraction

The "BioNLP 2009 Shared Task on Event Extraction" presented an important challenge to the biomedical relation extraction community. This initiative was concerned with the detection of the detailed behavior of bio-molecules involved in molecular events (Kim et al., 2009). Although all relevant named entities (i.e., proteins) were pre-tagged for this competition, a hitherto unseen number of event types (nine in total) differing in their structural and conceptual complexity had to be recognized and their arguments found and properly ordered. Even the arguments were rather complex as nesting of events were also allowed. The challenge targeted three subtasks addressing event extraction at various levels of specificity:

- Task 1 – "Event Detection and Characterization" required detection of nine molecular event types with arguments (proteins).
- Task 2 – "Event Enrichment" required recognition of additional arguments (not proteins) in detected events from Task 1.
- Task 3 – "Negation and Speculation Detection" required detections of negations and speculation statements concerning detected events from Tasks 1-2.

The major task of the challenge and its backbone was Task 1, which was mandatory for all challenge participants. Tasks 2 and 3 were optional and considered as enrichment tasks for event instances detected in Task 1. The demands placed on text analytics to deal with the complexity of Task 1 in terms of event diversity and specificity were unmatched by earlier information extraction challenges. Below I describe Task 1 in depth (Section 5.1.1), paying particular attention to the data sets and resources provided publicly for this competition (Section 5.1.2). The evaluation settings are explained in Section 5.1.3 before the performance results are discussed in the subsequent sections.

5.1.1 BioNLP 2009 Shared Task 1

Task 1 of the "BioNLP 2009 Shared Task on Event Extraction" featured the extraction of typed event mentions from a final set of event types. The set of nine event types given includes `Localization`, `Binding`, `Gene Expression`, `Transcription`, `Protein Catabolism`, `Phosphorylation`, `Positive Regulation`, `Negative Regulation`, and (unspecified) `Regulation` events. For the test data (a sample of 260 Medline abstracts), each participant was required to determine all events mentioned and link them appropriately with *a priori* supplied protein annotations, after the training period of 12 weeks.

The argument identification task for the Shared Task data can be decomposed into three layers dependent on the complexity of a considered event. *Level (1)* incorporates five event types (`Gene Expression`, `Transcription`, `Protein Catabolism`, `Localization`, and `Phosphorylation`) which involve a single participant with a `Theme` role only. *Level (2)* is concerned with one event type (`Binding`) that provides an n-ary argument structure where all arguments occupy the `Theme(`*n*`)` role. *Level (3)* comprises three event types (`Positive Regulation`, `Negative Regulation`, and unspecified `Regulation`) that represent a regulatory relation between

Table 5.1: Argument structures of Shared Task event types. n is the number of arguments allowed given the event type (first column) and argument types (second column $\{$Type$\}$).

Event Class Name	Argument Roles$\{$Types$\}$
`Gene Expression`	`Theme`$\{\text{Protein}\}_n$ $n = 1$
`Transcription`	`Theme`$\{\text{Protein}\}_n$ $n = 1$
`Protein Catabolism`	`Theme`$\{\text{Protein}\}_n$ $n = 1$
`Phosphorylation`	`Theme`$\{\text{Protein}\}_n$ $n = 1$
`Localization`	`Theme`$\{\text{Protein}\}_n$ $n = 1$
`Binding`	`Theme`$\{\text{Protein}\}_n$ $n \geqslant 1$
`Positive Regulation`	`Theme`$\{\text{Protein,Event}\}_n$ $n = 1$, `Cause`$\{\text{Protein,Event}\}_n$ $n = 0 \vee n = 1$
`Negative Regulation`	`Theme`$\{\text{Protein,Event}\}_n$ $n = 1$, `Cause`$\{\text{Protein,Event}\}_n$ $n = 0 \vee n = 1$
`Regulation`	`Theme`$\{\text{Protein,Event}\}_n$ $n = 1$, `Cause`$\{\text{Protein,Event}\}_n$ $n = 0 \vee n = 1$

the above-mentioned event classes or proteins. These events usually have a binary structure, with a `Theme` argument and a `Cause` argument. I summarize the argument structures from the Shared Task annotation guidelines for nine molecular events in Table 5.1. Argument types and roles are specified and parameter n indicates the size of arguments allowed for a single event instance. For argument extraction, JReX builds sentence-wise pairs of putative triggers and their putative argument(s), the latter involving ontological information about the event type.

In general, in Shared Task 1 data, all events are expressed using *text-bound* entities. Text-bound entities are annotations on text spans associated with a typed semantic class. These entities are protein annotations (`Protein`) and event triggers (e.g., `Gene Expression`, `Binding`). A text-bound entity is represented with a text offset span plus entity type, and is assigned an identifier (id). This id contains the prefix T and is non-ambiguous within a single Medline abstract. An *event* annotation is a reference to an n-tuple of text-bound entities. An event is assigned the prefix E and is non-ambiguous within a single Medline abstract. Below are annotation examples from the Shared Task data for simple and nested events, and multiple-type event triggers.

- An example of a simple event involving a single `Theme` argument (`Protein`) (PMID 9710600):

  Tax expression promotes N-terminal phosphorylation ...

T6 Protein Tax
T34 Gene_Expression expression
E3 Gene_Expression:T34 Theme:T6

This example from the training data contains a Gene Expression event annotation *E3* with "*Tax*" protein *T6* as a Theme argument and "*expression*" as a trigger word *T34*.

- An example of a event involving simple events as arguments (PMID 9710600):

Tax expression promotes N-terminal phosphorylation and degradation of IkappaB alpha ...

T6 Protein Tax
T7 Protein IkappaB alpha
T35 Positive_Regulation promotes
T37 Phosphorylation phosphorylation
E3 Gene_Expression:T34 Theme:T6
E4 Positive_Regulation:T35 Theme:E6 Cause:E3
E6 Phosphorylation:T37 Theme:T7

This example illustrates nesting of events. The Positive Regulation event *E4* has two simple events as arguments, the Gene Expression event *E3* as presented above (here as Cause argument), and the Phosphorylation event *E6* as a Theme argument. Thus, the complexity of event extraction in Task 1 is raised in terms of event nesting.

- An example of multiple-type event triggers, Positive Regulation and Gene Expression event (PMID 9710600)

Transfection of kinase-deficient mutants of IKKalpha and IKKbeta ...

T20 Protein IKKbeta
T46 Positive_Regulation Transfection
T47 Gene_Expression Transfection
E16 Positive_Regulation:T46 Theme:E17
E17 Gene_Expression:T47 Theme:T20

This example illustrates the annotation of multiple-type event trigger spans which refer to at least two events. Here, an event trigger span "*Transfection*" has two ids *T46* and *T47* which are referred to in the Positive Regulation event *E16* and Gene Expression event respectively. The annotation of

Table 5.2: Statistics on the Shared Task data sets.

Item	Training	Development	Test
Abstract	800	150	260
Sentence	7,449	1,450	2,447
Word	176,146	33,937	57,367
Event	8,615	1,789	3,182

different events on the same span concerns some event triggers such as "*overexpression*" or "*transfection*". The multiple-type event triggers are rather rare in the Shared Task data.

5.1.2 Data Sets and Supporting Resources

The Shared Task organizers prepared three data sets for Task 1, the training, development and test data sets. These sets comprise annotated MEDLINE abstracts extracted from the GENIA event corpus. The size of the data sets is 800 abstracts for training, 150 abstracts for development and 260 abstracts for testing. The training and development data were given to participants with entity and event mentions, while the test data was supplied only with protein span information. The statistics for all data sets are provided in Tables 5.2 and 5.3. The Table 5.2 provides statistics for abstract, sentences and tokens plus event instances. The Table 5.3 presents distribution of all nine event types. It reveals that some events are represented only by a few annotated examples, such as `Protein Catabolism` with 110 instances and `Phosphorylation` events with 169 instances in the training data. Other events are more frequently annotated, such as `Positive Regulation` in the lead with 2,847 instances in the training data, followed by `Gene Expression` (1,738) and `Negative Regulation` event with 1,062 instances in the training data.

The Shared Task data can be considered as partially annotated data, where instances of molecular events are restricted to protein-involving events only. There is a strong correlation between event annotations and the occurrence of proteins as arguments. For example, "*activation of NF-kappaB*" is not annotated as a `Positive Regulation` event because "*NF-kappaB*" is a protein complex (cf. Section 4.3). Thus, effective learning of events might be constrained by the partial annotation. Given the strong correlation between trigger annotations and the occurrence of proteins as arguments, JREX learns events in one step (Section 4.6) with the help of relation

Table 5.3: Event annotation statistics on the Shared Task data sets. This data is extracted from the official download data of the Shared Task.

Event type	Training	Development	Test
Gene Expression	1,738	356	722
Transcription	576	82	137
Protein Catabolism	110	21	14
Phosphorylation	169	47	135
Localization	265	53	174
Binding	887	248	347
Regulation	961	173	291
Positive Regulation	2,847	618	983
Negative Regulation	1,062	196	379
TOTAL	8,615	1,789	3,182

classification between a putative event trigger and an event argument. To achieve this goal, JReX preprocessed all the original training, development and test data by enriching the original data with automatically predicted and disambiguated event triggers (Section 4.4) in order to generate more negative examples for a more effective learning of true events.

The Shared Task organizers made additional resources available to challenge participants in the form of sentence and token annotations for all data sets and even syntactic parsing data. The syntactic data was automatically generated by parsing the Shared Task data with various state-of-the-art parsers. The parsers applied were three constituency-based parsers, the Bikel parser (Bikel, 2004), the re-ranking Charniak-Johnson parser (Charniak and Johnson, 2005) with the externally trained biomedical parsing model from McClosky and Charniak (2008) (M+C parser), and the Combinatory Categorial Grammar parser (Curran et al., 2007) adapted to the biomedical domain (Rimell and Clark, 2009), as well as at least one dependency-based parser, GDep (Sagae and Tsujii, 2007). The output of constituency-based parsers was converted to the Stanford dependency representation in its *collapsed* mode (Section 4.5). Thus, given the Shared Task data pre-processed with syntactic parsers and supplied with gold protein annotation, Task 1 participants could focus only on the event detection task proper. This allows a better comparison between participating systems using common evaluation settings. The evaluation measures and settings used in the Shared Task are explained in the next section.

5.1.3 Evaluation Settings

The evaluation configuration in the Shared Task exploits standard recall, precision and F-score metrics (cf. Glossary for definitions). Recall and precision are normally used for the evaluation of a range of NLP tasks.

For calculating recall and precision metric values, we need to investigate whether an event is correctly identified. This setting is represented through the Shared Task equality criteria for events (Kim et al. (2009)). Event equality is defined as follows: two events are equal if (1) the event types are the same, (2) the event triggers are equal, and (3) the arguments are equal. Two arguments are considered to be equal if (1) their roles are the same, (2.1) both are text-bound entities and are equal, and (2.2) both are events and are equal. The equality of text-bound entities is defined as follows: two text-bound entities are equal if (1) their types are the same and (2) their spans are the same. The spans are the same for two text-bound entities with spans $(start1, end1)$ and $(start2, end2)$ if $start1 = start2$ and $end1 = end2$.

The parametrization of the equality criteria results in three equality modes applied to the Shared Task output data listed below:

- *Strict Matching* – this matching mode requires exact equality as defined above.
- *Approximate Span Matching* – this matching mode relaxes the requirements for span matching for text-bound entities. Two spans are defined as equal if the span of a detected text-bound entity is entirely contained within the span of a gold instance with an extension of a gold span by one token to the left and the right as follows: $start1 \geq estartgold$ and $end1 \leq eendgold$ where $(estartgold, eendgold)$ is the extended gold span and $(start1, end1)$ is the span of the given span of a detected instance.
- *Approximate Recursive Matching* – this matching mode is based on approximate span matching and relaxes the equality requirements for regulation events. In the strict matching, a regulation event is correct if its argument events are strictly correct. This mode relaxes this requirement and allows the argument events to be partially correct. Furthermore, the `Cause` argument is ignored.

In addition, the Shared Task introduced the *event decomposition mode* which decomposes events with an argument size greater than one argument into events with single arguments. This relaxation concerns `Binding`, `Regulation` events and subtype events. These events with n arguments are converted to n events with a single argument.

In the official evaluation of the Shared Task, the *Approximate Recursive Matching* has been applied as the main evaluation mode.

5.2 JReX Shared Task Results

This section presents results of the JReX system on the official Shared Task data. I start with the presentation of JReX configuration in solving the Shared Task 1 on event extraction.

5.2.1 Configuration of the JReX System

In evaluating event extraction performance, one should bear in mind that this semantic process depends crucially on preceding lexical and sentential analysis, coverage and quality of lexical resources, and efficiency of applied event extraction techniques. The JReX system has a pipeline architecture, described in detail in the previous chapter. For the Shared Task experiments the JReX standard configuration is applied (Section 4.7), except for the parsing part. In the official run, the GDep parsing results provided by the Shared Task organizers were integrated into the JReX solution in order to make the results comparable in this challenge. Given the pre-processing and parsing data integration, the JReX event extraction tool starts by distinguishing between the event trigger detection using a dictionary-based approach and disambiguation (cf. Section 4.4), trimming the dependency graphs (cf. Section 4.5), and the trigger-argument classification proper (cf. Section 4.6). As for semantic enrichment, in my experiments JReX applies full conceptual decoration for the kernel-based representation and only partial decoration for the dependency parse features (only gene/protein annotation was exploited here). This is because graph representations allow many semantic labels to be associated with a node. Therefore, for the kernel-based representation lexical nodes are replaced with entity information enriched with GOA and experimental methods information. The event trigger lexical nodes are enriched with a corresponding event trigger type(s). In the official run, the trimming approaches were available only for *aux* and *preps* procedures, but this was extended after the competition to include *coords* and *np action* procedures (cf. Section 4.5). For Task 1, the event creation step varies between the three different event levels. For each event trigger of Level (1) (e.g., `Gene Expression`), JReX generates one event per relation comprising the trigger and its argument. For the second Level (`Binding`), JReX creates a `Binding` event with two arguments for triples (trigger, protein$_1$, protein$_2$). For Level (3), JReX creates for each event trigger and its associated arguments $n \times m$ events, for n `Cause` arguments and m `Theme` arguments.

In some cases the event creation step needs a range of post-processing heuristics. For example, JReX removes those predicted events of which the arguments were already included in other events with a higher number of arguments. Furthermore, JReX duplicated events involving special triggers, such as "*overexpression*" or "*transfection*", according to the annotation guidelines of the Shared Task data for multiple-type event triggers (Section 5.1.1).

In the next section I present JReX evaluation results on the Shared Task data.

5.2.2 Event Trigger Detection and Disambiguation

The evaluation results for detecting event triggers and their disambiguation (Section 4.4) are introduced below. The results presented in Table 5.4 should be interpreted carefully. This is because recall reflects the number of triggers that JReX finds using a dictionary-based approach and correctly disambiguates. The figures for precision, however, reflect the ambiguity of putative triggers. They reveal how many lexical cues team up with potential arguments as positive learning examples for the event detection (cf. Section 4.6). The candidate pairs are then classified during the argument detection step and a proportion of pairs will be classified as not being events.

The recall values of approximate matching (see approximate span matching in Section 5.1.3) range between 72.6% (`Positive Regulation`) and 100% (`Protein Catabolism`, `Phosphorylation`). For most of the events the recall ranges between 70% and 90%. Using collected dictionaries JReX detects in total more than 80% of all event triggers in the development data set. The precision values range between 21.3% (`Regulation`) and 82.6% (`Protein Catabolism`). Events with the highest ambiguity include all `Regulation`, `Localization` and `Transcription` events.

The ranking in trigger detection performance corresponds to a great extent to the overall event extraction ranking (cf. Tables 5.4 and 5.9). The highest F-scores are achieved for `Phosphorylation` and `Protein Catabolism`, while the lowest F-scores are determined for the `Regulation` events. Obviously, the coverage of lexical cues and their ambiguity play a crucial role in overall event extraction. In addition, I evaluated the heuristics for event trigger disambiguation (Section 4.4.2) in terms of accuracy values (cf. Glossary for the accuracy definition). The results reveal that for most event types, the *Importance*-based disambiguation criterion ((4.4) in Section 4.4.2) outperforms the frequency and TF-IDF score-based heuristics (Table 5.5). In particular, the `Protein Catabolism` and `Binding` event triggers can successfully be disambiguated. This can be explained by the distribution of event instances in the Shared Task data. The annotation figures of nine molecular events are different

Table 5.4: Evaluation of event trigger words detection and *Importance*-based disambiguation. Exact matching/approximate matching result values on the Shared Task development data.

Event Class	**gold**	**recall**	**prec.**	**F-score**
`Gene Expression`	282	77.7/88.6	55.3/64.4	64.6/74.6
`Transcription`	68	66.2/80.3	22.7/29.6	33.8/43.3
`Protein Catabolism`	19	100.0/100.0	82.6/82.6	90.5/90.5
`Phosphorylation`	40	97.5/100.0	75.0/77.4	84.8/87.3
`Localization`	40	92.5/92.5	24.3/24.3	38.5/38.5
`Binding`	180	86.1/92.7	35.7/40.0	50.5/55.9
`Regulation`	138	67.4/75.7	18.0/21.3	28.4/33.2
`Positive Regulation`	462	68.0/72.6	23.3/25.4	34.7/37.6
`Negative Regulation`	153	85.6/87.8	26.4/27.5	40.4/41.8
TOTAL	**1382**	**76.1/82.6**	**29.1/32.6**	**42.1/46.8**

Table 5.5: Evaluation of event trigger disambiguation heuristics, accuracy values on the Shared Task development data.

Event Class	**gold**	***trigger_imp***	***trigger_freq***	***trigger_tf-idf***
`Gene Expression`	282	94.6	86.8	86.5
`Transcription`	68	82.3	77.9	86.5
`Protein Catabolism`	19	100.0	100.0	100.0
`Phosphorylation`	40	97.5	100.0	100.0
`Localization`	40	97.5	85.0	80.8
`Binding`	180	100.0	97.2	97.2
`Regulation`	138	96.3	83.3	79.7
`Positive Regulation`	462	85.9	88.9	86.5
`Negative Regulation`	153	92.1	98.0	92.1

in the corpus. For example the training corpus contains 1,738 instances of `Gene Expression` events and only 265 instances of `Localization` events. In the case of the ambiguity, it makes sense to consider event types as equally important and this is achieved with the help of the *Importance*-based disambiguation criterion.

In the next step, the detected and disambiguated putative event triggers team up with their potential arguments to build pairs which are classified in the JReX system during the argument extraction step evaluated in the following sections.

5.2.3 Baseline for Argument Extraction

The baseline against which I compared the JReX argument extraction methods can be captured in a single rule. Crucial for this rule is the availability of dependency information. The GDep parsing results provided by the Shared Task organizers were integrated into the baseline. For each pair of a putative trigger and a putative argument, the baseline extracts the shortest dependency path between them. If the shortest dependency path does not contain any direction change, i.e., the argument is either a direct child or a direct parent of the trigger, and if the path does not contain any other intervening event triggers, the argument is assigned the `Theme` role. Pairs of mentions not connected by a dependency path could not be detected.

Table 5.6: Baseline results on the Shared Task development data for Approximate Span Matching/Approximate Recursive Matching.

Event Class	**gold**	**recall**	**prec.**	**F-score**
`Localization`	53	75.47	30.30	43.24
`Binding`	248	33.47	20.80	25.66
`Gene Expression`	356	76.12	75.07	75.59
`Transcription`	82	68.29	40.58	50.91
`Protein Catabolism`	21	76.19	66.67	71.11
`Phosphorylation`	47	76.60	72.00	74.23
`Regulation`	169	14.20	15.09	14.63
`Positive Regulation`	617	15.40	20.83	17.71
`Negative Regulation`	196	11.73	13.22	12.43
TOTAL	**1789**	**36.00**	**34.02**	**34.98**

I performed evaluations on the Shared Task development and test set. The baseline achieved competitive results of 36.0% recall, 34.0% precision, and 35.0% F-score on the development set (Table 5.6), but 30.4% recall, 35.7% precision, and 32.8% F-score on the test set (Table 5.7). In particular, the single-argument events, i.e., `Gene Expression`, `Phosphorylation`, and `Protein Catabolism`, were effectively extracted with an F-score of 69.7%, 76.4%, and 69.0%, on test data respectively. Recognition of more complex events, in particular events of Level (3), i.e., (`Regulation`) and its subtypes, was much worse (ranging from a 10% to a 16% F-score on test data) because of their great internal complexity. Furthermore, `Cause`

Table 5.7: Baseline results on the Shared Task test data. Approximate Span Matching/Approximate Recursive Matching (columns 3-5). Event decomposition, Approximate Span Matching/Approximate Recursive Matching (columns 7-9).

Event Class	**gold**	**recall**	**prec.**	**F-score**	**gold**	**recall**	**prec.**	**F-score**
Gene Expression	722	61.36	80.55	69.65	722	61.36	80.55	69.65
Transcription	137	39.42	35.06	37.11	137	39.42	35.06	37.11
Protein Catabolism	14	71.43	66.67	68.97	14	71.43	66.67	68.97
Phosphorylation	135	65.93	90.82	76.39	135	65.93	90.82	76.39
Localization	174	42.53	44.85	43.66	174	42.53	44.85	43.66
Binding	347	32.28	37.09	34.51	398	44.22	58.28	50.29
EVT-TOTAL	**1529**	**51.14**	**60.90**	**55.60**	**1580**	**53.54**	**65.89**	**59.08**
Regulation	291	9.62	11.72	10.57	338	9.17	12.97	10.75
Positive Regulation	983	10.38	11.33	10.83	1186	14.67	19.33	16.68
Negative Regulation	379	14.25	19.22	16.36	416	14.18	21.00	16.93
REG-TOTAL	**1653**	**11.13**	**12.96**	**11.98**	**1940**	**13.61**	**18.59**	**15.71**
TOTAL	**3182**	**30.36**	**35.72**	**32.82**	**3520**	**31.53**	**41.05**	**35.67**

arguments were not considered in the baseline approach. The restrictions of the baseline approach should be overcome in the JReX approach to argument extraction evaluated in the next section.

5.2.4 JReX Components for Argument Extraction

In preliminary experiments on the development data, the performance of each classifier type and its variants was determined (for the graph kernel), as well as the performance of ensembles of the best-performing (F-Score) graph kernel variant and MaxEnt model. For the ensemble configurations, the union of positive instances was considered. This means that an SVM employing a graph kernel, as well as a MaxEnt model, were used to predict on the data. Both prediction results were merged by considering every example as a positive event that had been classified as such by either the SVM or the MaxEnt model. An outcome of the experiments on the development data is the argument extraction configuration used for the official run in the Shared Task, which I present below. JReX achieved the best performance on the development set with this configuration.[1] For the prediction of `Phosphorylation`, `Localization`, `Protein Catabolism` event types JReX used the graph kernel, while for the prediction of `Transcription`, `Gene Expression` and `Binding` events JReX used an ensemble of the graph kernel and a MaxEnt model, while for regulatory events JReX used MaxEnt models for each regulatory type.

The event extraction approach, in its final configuration, achieved a performance of 50.4% recall, 45.8% precision and 48.0% F-score on the development set (Table 5.8), and 45.8% recall, 47.5% precision and 46.7% F-score on the test set (Table 5.9). This approach clearly outperformed the baseline, with an increase of 14 percentage points on the test data. In particular, the events of Level (2) and (3) were dealt with better than by the baseline. In the event decomposition mode (cf. Section 5.1) JReX achieved a performance of 49.4% recall, 56.6% precision, and 52.7% F-score (Table 5.9).

The experiments on the development set revealed that the ensemble combination of the feature-based and the graph kernel-based approaches can boost results by up to 6 percentage points F-score (for the `Binding` event type). It is interesting that the combination for `Binding` increased recall without penalizing precision. The original graph kernel approach for `Binding` events performed with 38.3% recall, 27.9% precision and 32.3% F-score on the development set. The combined approach comes with a stunning increase of 14 percentage points in recall. The combination also

[1] For the final configurations of the graph kernel, JReX optimized the C parameter (SVM Parameter) in the spectrum between 2^{-3} and 2^{3} on the final training data for every event type separately.

Table 5.8: Event extraction results on the Shared Task development data of the official run of the JULIE Lab Team (JREX system). Approximate Span Matching/Approximate Recursive Matching.

Event Class	**gold**	**recall**	**prec.**	**F-score**
`Gene Expression`	356	75.28	81.46	78.25
`Transcription`	82	60.98	73.53	66.67
`Protein Catabolism`	21	90.48	79.17	84.44
`Phosphorylation`	47	82.98	84.78	83.87
`Localization`	53	71.70	74.51	73.08
`Binding`	248	52.42	29.08	37.41
`Regulation`	169	37.87	36.78	37.32
`Positive Regulation`	617	34.36	35.99	35.16
`Negative Regulation`	196	41.33	33.61	37.07
TOTAL	**1789**	**50.36**	**45.76**	**47.95**

boosted the recall of the `Gene Expression` and `Transcription` by 15 percentage points and 5 percentage points, respectively, without seriously impairing precision (4 points lower for each type).

After the official run of the Shared Task, the JREX system was modified by updating its functionality in several ways. First, the simplification strategy was extended to account for coordinations and noun phrases. Second, the graph kernel information was updated with a special kernel variant for `Binding` events (see Section 4.6.3). Third, post-processing heuristics were developed for events and filtering and sampling scenarios for negative instances, which I describe below.

As mentioned previously, some biological events are lexically triggered by a set of highly ambiguous trigger words. Examples such as *"lead"* or *"follow"* may signal biologically relevant events in the documents but may also occur in expressions which (predominantly) have no biological event reading at all. In particular, all `Regulation` events can be characterized as highly ambiguous. From the perspective of (supervised) machine learning, JREX suffers from the fact that under these circumstances a large number of training instances and, in particular, negative training instances are available so that the proportion of negative to positive training instances amounts to 10:1 in the training data. JREX tried to cope with this asymmetry using under-sampling as a sampling strategy.

The sampling scenario is complemented by employing filtering heuristics for nega-

Table 5.9: Official event extraction results on the Shared Task test data of the Julie Lab Team (JReX system). Approximate Span Matching / Approximate Recursive Matching (columns 3-5). Event decomposition, Approximate Span Matching/Approximate Recursive Matching (columns 7-9).

Event Class	**gold**	**recall**	**prec.**	**F-score**	**gold**	**recall**	**prec.**	**F-score**
`Gene Expression`	722	64.82	80.27	71.72	722	64.82	80.27	71.72
`Transcription`	137	35.77	62.03	45.37	137	35.77	62.03	45.37
`Protein Catabolism`	14	78.57	84.62	81.48	14	78.57	84.62	81.48
`Phosphorylation`	135	76.30	91.15	83.06	135	76.30	91.15	83.06
`Localization`	174	43.68	77.55	55.88	174	43.68	77.55	55.88
`Binding`	347	49.57	35.25	41.20	398	63.57	54.88	58.91
EVT-TOTAL	**1529**	**57.49**	**63.97**	**60.56**	**1580**	**60.76**	**71.27**	**65.60**
`Regulation`	291	31.27	30.13	30.69	338	35.21	38.14	36.62
`Positive Regulation`	983	34.08	37.18	35.56	1186	40.64	50.00	44.84
`Negative Regulation`	379	40.37	31.16	35.17	416	42.31	39.55	40.88
REG-TOTAL	**1653**	**35.03**	**34.18**	**34.60**	**1940**	**40.05**	**45.15**	**42.45**
TOTAL	**3182**	**45.82**	**47.52**	**46.66**	**3520**	**49.35**	**56.62**	**52.73**

Table 5.10: Final configuration of the modified JReX system. (Under-sampling proportion of negative instances to positive instances in parentheses.)

Event Class	Machine Learning approach
`Gene Expression`	graph kernel & MaxEnt
`Transcription`	graph kernel & MaxEnt
`Protein Catabolism`	graph kernel
`Phosphorylation`	graph kernel
`Localization`	graph kernel
`Binding`	graph kernel & MaxEnt
`Regulation`	graph kernel (2:1) & MaxEnt (2:1), Pre-filtering of event triggers
`Positive Regulation`	MaxEnt (2:1)
`Negative Regulation`	graph kernel (2:1) & MaxEnt

tive instances. Considering the characteristics of the shortest paths, for `Regulation` and its sub-events JReX eliminated all paths longer than eight nodes.[2] It also eliminated all instances with proteins as potential arguments, if proteins were already involved in other non-regulatory events (`Gene expression`, `Transcription`, `Binding`, `Localization`, `Protein Catabolism`, `Phosphorylation`), and instances whose shortest paths contained true event triggers of non-regulatory events. Another filtering strategy JReX used for particularly ambiguous lexical triggers (related to `Regulation` events) was to perform the first classification step as a trigger recognition task and classify trigger words as event triggers, and subsequently to carry out a second classification step that considered only the remaining triggers.[3] This pre-filtering step relaxes the one-step event extraction performed by our system. In this step, JReX pre-processes the data with a graph kernel classifier that integrates all shortest dependency paths between the trigger word and all proteins in the sentence.

Table 5.10 summarizes the final configuration of the modified JReX system, which comes as an ensemble configuration of both ML approaches for most of the event types. For pre-processing Shared Task data JReX used the Julie Lab sentence splitting and tokenization tools (Tomanek et al. (2007b)) and the MST parser (Mc-

[2] The parameter length for path length was found in my experiments on the Shared Task training data.

[3] This approach to reduce the number of instances on a per-task basis is taken by most ML-based event prediction systems.

Table 5.11: Event extraction results of the modified JReX system on the development and test data (pre-processed with the MST parser). Approximate Span Matching/Approximate Recursive Matching.

	Development Data			Test Data		
Event Class	**recall**	**prec.**	**F-score**	**recall**	**prec.**	**F-score**
`Gene Expression`	74.16	80.73	77.31	66.76	79.28	72.48
`Transcription`	62.20	68.00	64.97	33.58	59.74	42.99
`Protein Catabolism`	76.19	80.00	78.05	71.43	90.91	80.00
`Phosphorylation`	85.11	80.00	82.47	79.26	84.92	81.99
`Localization`	83.02	80.00	81.48	46.55	91.01	61.60
SVT-TOTAL	**74.24**	**78.75**	**76.43**	**61.42**	**79.69**	**69.37**
`Binding`	52.02	51.39	51.70	46.69	52.09	49.24
EVT-TOTAL	**67.41**	**69.92**	**68.64**	**58.08**	**72.67**	**64.56**
`Regulation`	31.95	51.92	39.56	25.09	41.24	31.20
`Positive Regulation`	40.52	44.56	42.44	36.11	45.75	40.36
`Negative Regulation`	44.39	43.28	43.83	37.73	40.40	39.02
REG-TOTAL	**39.82**	**45.15**	**42.32**	**34.54**	**43.69**	**38.58**
TOTAL	**52.26**	**56.87**	**54.47**	**45.85**	**57.69**	**51.09**

Donald et al. (2005)) retrained on the Genia treebank converted to the CoNLL'07 representation.[4] All these improvements produced a system that achieved a 6.5 percentage point improvement in the F-score over the basic JReX system on the development set and a 4.4 percentage point improvement in the F-score on the test set (Table 5.11). Recognition of `Binding` events particularly benefits from the recent changes, with a 14.3 percentage point improvement in the F-score on the development data set (8.0 percentage points on the test data).

5.2.5 Evaluation of Feature Types and Graph Kernel Variants

The evaluation experiments done on the development and test data are complemented by a further evaluation study of the argument detection subtask on the Shared Task training data. The ten-fold cross-validation of the JReX argument

[4] The Genia Treebank version 1.0 was applied, available from `http://www-tsujii.is.s.u-tokyo.ac.jp`. The conversion script is accessible via `http://nlp.cs.lth.se/pennconverter/`.

extraction component was performed on the training data and searched for better configurations of the JReX event extraction approaches.

First, a feature-based approach was evaluated for suitability of various feature types. Five configurations of the feature-based JReX argument extraction component were evaluated in this study, e.g., *Lexical and Semantic Type Features*, *Chunking Features*, *Walk Features* (shortest dependency path features), *Walk and Katrenko Features* (dependency path and tree level features), and *All Features* (see Section 4.6.2 and cf. Tables 5.12, 5.13, 5.14).[5] The evaluation results reveal that walk features clearly outperform all other feature types for all events except `Protein Catabolism` events which can be effectively extracted (74.6% F-score) using lexical features only. Interestingly, the chunk features are shown in my study to be less useful for argument extraction than all other feature types. They are outperformed even by the JReX variant with lexical features only. The JReX variant with the combination of all features, achieves best results for all Shared Task events (cf. Table 5.14). The performances of this JReX variant increase compared to the configuration with walk features only (cf. Tables 5.13 and 5.14). For example, the F-score values for complex regulatory events change from 60.5% to 64.9% for `Regulation` events, from 66.5% to 69.9% for `Positive Regulation` events, but the F-score remains almost unchanged for `Negative Regulation` events. My conclusion from this study is that the shortest path dependency features in the form of walk features are the most suitable feature types for argument extraction for the Shared Task events and thus for a range of representative molecular event descriptions. Still, lexical and semantic type features can enhance the overall argument extraction performance. This outcome confirms the results of the feature evaluation study on PPI corpora by Faessler (2009).

In a further study, I took a close look on various configurations of the graph kernel approach in the JReX argument extraction component. Three graph kernel variants have been evaluated in a ten-fold cross-validation on the Shared Task training data, e.g., the shortest dependency path (*sp*) between two mentions (argument and trigger); second, the complete dependency graph (*sp-dep*); and third, the complete dependency graph and linear information (*sp-dep-lin*) (cf. Section 4.6.3). The evaluation results are presented in Table 5.15. The JReX graph kernel *sp* variant achieves F-score results for events of Level (1) between 75.9% (`Localization`) and 87.7% (`Phosphorylation`). These results do not improve if the complete dependency graph and linear lexical information are applied (see Table 5.15). However, one event type, *viz.* `Protein Catabolism`, is shown to profit from the linear information in the graph kernel with an increase of more than three percentage points (from

[5] I provide in this section F-score results for `Gene Expression` and `Transcription` event as one single value, because arguments of these events are learned in a common argument extraction model.

Table 5.12: Results of a ten-fold cross-validation of JREX feature-based variants on the Shared Task training data. *Lexical and Semantic (LS) Type Features*, *Chunking Features* are applied.

	LS Type Features			*Chunking Features*		
Event Class	**recall**	**prec.**	**F-score**	**recall**	**prec.**	**F-score**
`Gene Expression & Transcription`	66.4	71.3	68.8	38.6	50.9	43.9
`Protein Catabolism`	72.9	76.4	74.6	75.6	65.1	70.0
`Phosphorylation`	70.9	74.5	72.7	56.3	73.8	63.9
`Localization`	50.9	71.1	59.3	33.9	74.7	46.7
`Binding`	53.6	66.1	59.2	42.5	63.6	51.0
`Regulation`	50.5	67.5	57.8	37.4	59.7	46.0
`Positive Regulation`	55.7	71.5	62.6	37.3	59.7	45.9
`Negative Regulation`	55.0	72.7	62.6	40.7	63.7	49.7

Table 5.13: Results of a ten-fold cross-validation of JREX feature-based variants on the Shared Task training data. *Walk Features*, *Walk and Katrenko Features* are applied.

	Walk Features			*Walk and Katrenko Features*		
Event Class	**recall**	**prec.**	**F-score**	**recall**	**prec.**	**F-score**
`Gene Expression & Transcription`	74.5	77.2	**75.8**	74.3	76.4	75.4
`Protein Catabolism`	72.9	68.6	**70.7**	73.9	68.3	70.9
`Phosphorylation`	86.0	73.5	**79.3**	85.4	75.0	79.8
`Localization`	58.3	83.1	**68.5**	56.0	81.7	66.5
`Binding`	59.7	71.9	**65.2**	58.7	71.8	64.6
`Regulation`	52.7	71.2	**60.5**	52.7	72.1	60.9
`Positive Regulation`	60.1	74.2	**66.5**	60.5	75.5	67.2
`Negative Regulation`	57.5	75.4	**65.3**	55.9	74.7	64.0

81.8% to 85.0% F-score). This confirms the evaluation results from the feature-based JREX study above where `Protein Catabolism` events can be effectively extracted using lexical linear information only. The events of Level (2) and (3) can be extracted with an F-score performance ranging between 65.2% (`Binding`) and 70.5% (`Negative Regulation`) using the JREX graph kernel *sp* variant only. The modeling of complete dependency graphs or even integration of linear information does not enhance the F-score results for solving this task for these events. While the

Table 5.14: Results of a ten-fold cross-validation of the JReX feature-based approach on the Shared Task training data. *All Features* are applied.

	All Features		
Event Class	**recall**	**prec.**	**F-score**
`Gene Expression & Transcription`	76.3	79.4	**77.8**
`Protein Catabolism`	79.2	75.8	**77.5**
`Phosphorylation`	84.8	80.4	**82.5**
`Localization`	61.6	86.0	**71.8**
`Binding`	65.2	76.0	**70.3**
`Regulation`	56.3	76.5	**64.9**
`Positive Regulation`	61.9	80.1	**69.9**
`Negative Regulation`	54.9	79.9	**65.1**

additional information from complete dependency trees is not helpful, the use of full linear sentence information can result in a worse performance. Thus, the evaluation results reveal that the modeling of the shortest path only is sufficient for a performant argument extraction.

Both evaluation studies reveal that the shortest dependency path is the most crucial information source for solving the argument extraction task in feature-based and graph kernel-based approaches. The shortest dependency path information between the trigger and its potential argument is sufficient for extracting event arguments. Given this outcome, my next question is whether various dependency representations have an effect on the overall performance in solving this major event extraction subtask. The next section presents insights to answer this question.

5.3 An Empirical Assessment of Dependency Graph Trimming

The JReX system is currently one of three top-performing systems in the event extraction task, the others being the Tokyo and Turku systems (see Section 4.2). All three systems rely on dependency graphs for solving the event extraction task. While the Turku system exploits the Stanford dependencies from the McClosky-Charniak parser (M+C parser) and the JReX system uses the CoNLL-like dependencies from the GDep parser or the CoNLL dependencies from the MST parser, the Tokyo

Table 5.15: Ten-fold cross-validation of graph kernel configurations on the Shared Task training data, e.g., "Graph Kernel – Shortest Path" (*sp*), "Graph Kernel – Dependency Graph" (*sp-dep*), and "Graph Kernel – Dependency Graph and Linear Information" (*sp-dep-lin*).

	Graph *sp*			Graph *sp-dep*			Graph *sp-dep-lin*		
Event Class	**recall**	**prec.**	**F-score**	**recall**	**prec.**	**F-score**	**recall**	**prec.**	**F-score**
`Gene Expression & Transcription`	80.2	86.5	83.3	82.1	85.0	83.5	82.1	85.1	83.6
`Protein Catabolism`	81.0	82.6	81.8	81.0	82.5	81.9	81.0	89.1	85.0
`Phosphorylation`	86.7	88.8	87.7	88.5	87.0	87.7	84.8	89.7	87.2
`Localization`	69.0	84.2	75.9	64.9	82.6	72.7	68.6	82.3	74.8
`Binding`	60.3	71.0	65.2	56.8	68.4	62.0	57.2	74.8	64.8
`Regulation`	60.6	74.5	66.8	60.1	75.4	67.0	57.5	75.5	65.3
`Positive Regulation`	66.2	78.0	71.6	60.1	75.4	67.0	66.2	78.7	72.0
`Negative Regulation`	63.2	79.7	70.5	62.1	79.1	69.6	61.8	78.4	69.1

system overlays the Shared Task data with two parsing representations, *viz.* Enju PAS structure (Miyao and Tsujii, 2002) and GDep parser dependencies. Obviously, the question arises as to what extent the performance of these systems depends on the choice of the parser and its output representations. Miyao et al. (2008) has already assessed the impact of different parsers for the task of biomedical relation extraction (PPI). Here I perform a similar study for the task of event extraction and focus in particular on the impact of various dependency representations such as Stanford and CoNLL dependencies, and additional trimming procedures.

The results reported in this section are obtained from experiments performed with the JReX system. The main goal was to investigate the crucial role of proper representation structures for dependency graphs. Therefore, I focus on the implications of various dependency representations for the identification of trigger-argument relations. Furthermore, the effects of trimming dependency graphs (Section 4.5) on event extraction task will be discussed in this section, both with respect to syntactic simplification and semantic enrichment.

5.3.1 Dependency Graph Distances between Trigger and Arguments

Before addressing the effects of various dependency graph representations and benefits of dependency graph trimming in terms of F-score metrics, I present the effects of alternative dependency representations on event extraction by examining the distances between the event trigger word and its potential arguments for an event in terms of the number of dependency edges between them. Figures are given for the `Phosphorylation` event (Figures 5.1, 5.2), the event with the highest F-score in my event extraction approach (cf. Table 5.9), and the `Transcription` event (Figures 5.3, 5.4), one of the most problematic events in terms of a low F-score. These figures show the distances between trigger words and potential arguments for the GDep (CoNLL-style) and for the M+C parser (SD) parsing results. The M+C parsing results have been converted to SD representation in three configurations — *basic*, *collapsed* and *ccprocessed* (Section 4.5). GDep parsing results were modified for the representation of auxiliary and modal verbs, and the collapsing of prepositions.[6] The distances of the paths that represent true event relations (true event paths) and those that do not represent any event relation (false event paths) are depicted. It is evident that the distances indeed depend on the selected dependency representation (Figures 5.1, 5.2, 5.3 and 5.4). For example, the original GDep dependencies provide the longest dependency paths between the objects of interest. The GDep auxiliary modified (see *aux* procedure in Section 4.5.2) paths correspond to the distances of

[6] Since the representation of coordinations in the GDep parser differs from the CoNLL coordination representation format, this phenomenon was disregarded in this experiment.

Table 5.16: Distances between trigger and potential arguments, M+C SD *ccprocessed*. Mean values (μ) of distances (column 2-3), and standard deviations (SD) of distances for true and false event paths (columns 5-6).

Event	μ **true**	μ **false**	**ratio of** μ	SD **true**	SD **false**
`Gene Expression`	1.6	4.0	0.40	1.0	1.9
`Transcription`	1.9	4.0	0.48	2.0	1.0
`Protein Catabolism`	1.8	4.4	0.40	1.3	2.0
`Phosphorylation`	1.6	4.3	0.38	1.0	2.0
`Localization`	1.8	3.9	0.46	1.0	1.9
`Binding`	2.4	4.3	0.56	1.4	2.3
`Regulation`	1.7	3.4	0.52	1.0	1.9
`Positive Regulation`	1.7	3.3	0.50	1.0	1.9
`Negative Regulation`	1.6	3.3	0.48	0.8	1.8
TOTAL	**1.8**	**3.5**	**0.50**	**1.1**	**2.0**

the *basic* SD. For `Phosphorylation`, for example, there is an increase of about five percentage points for distance 1. By collapsing the prepositions, distance 1 increases by about 30 percentage points. The propagation of conjunct dependencies in SD also brings some benefit. Thus, the distances between the trigger word and its arguments can be properly reduced. This increase can be demonstrated for all events under scrutiny.

Even more telling are the differences between distances of various events. I limit the discussion to the SD *ccprocessed* representation scheme as they are similar with the trimmed CoNLL scheme results. For the `Phosphorylation` event, about 60% of paths between event triggers and their arguments (true event paths) have a distance of 1 edge (90% up to 2 edges). The remaining paths between potential event triggers and their potential arguments (false event paths) only come to about 12% for the same distances. In contrast to `Phosphorylation`, for the `Transcription` event only about 43% of true event paths could be found with distance 1 (about 80% with distance up to 2) and a number of false event paths with distance up to 2 (25%). This seems to indicate that the structures of the `Transcription` events might be harder to learn and that, in particular, the ambiguity of the triggers is higher than for the `Phosphorylation` event (cf. Table 5.4). The same effects can be observed for other problematic events.

Next, the mean value (μ) of distances of true and false event paths (Table 5.16) and the ratio between them were measured. For the `Transcription` event, the μ

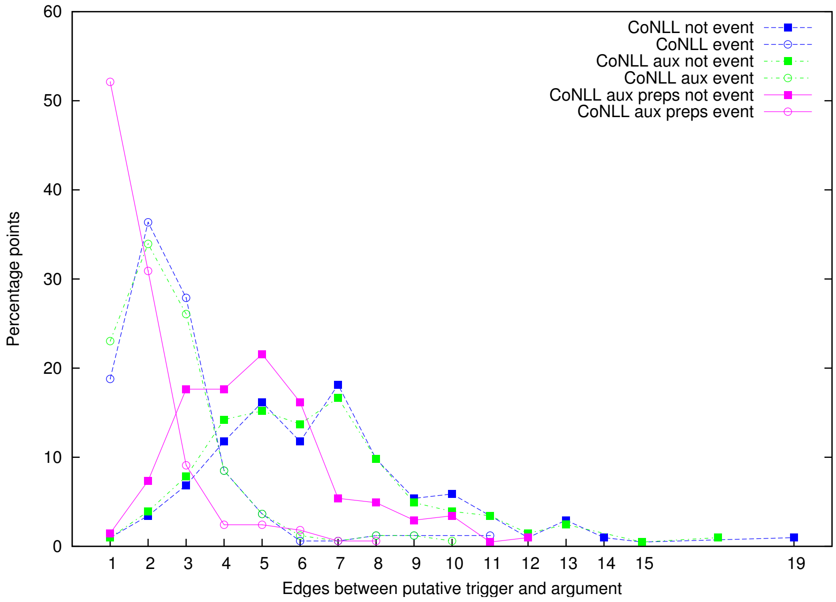


Figure 5.1: `Phosphorylation` event, distances between trigger word and proteins in basic and simplified (trimmed) CoNLL dependency trees.

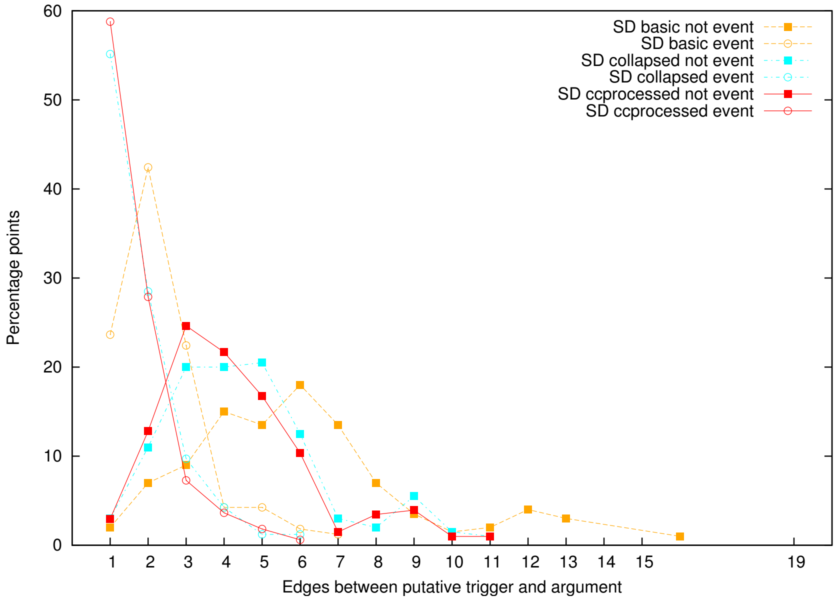


Figure 5.2: `Phosphorylation` event, distances between trigger word and proteins in *basic* and *collapsed/ccprocessed* SD dependency trees.

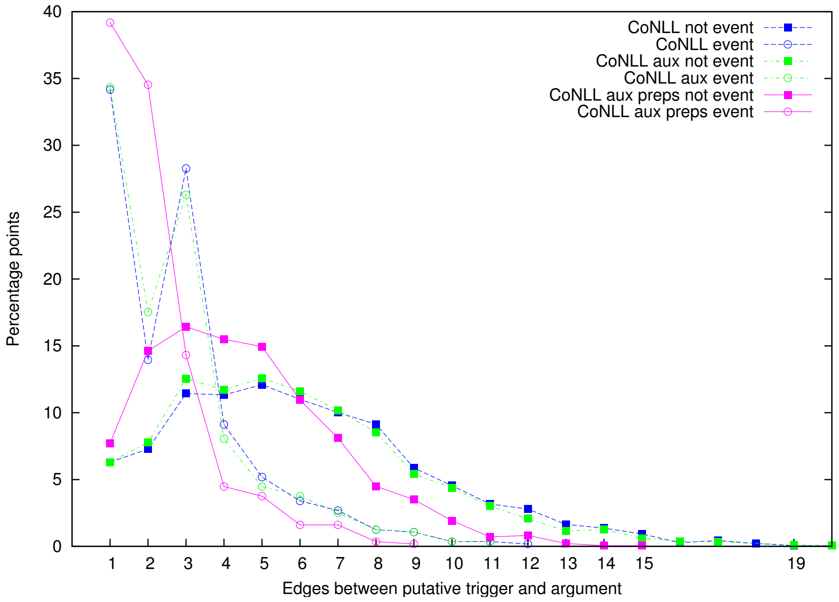


Figure 5.3: `Transcription` event, distances between trigger word and proteins in basic and simplified (trimmed) CoNLL dependency trees.

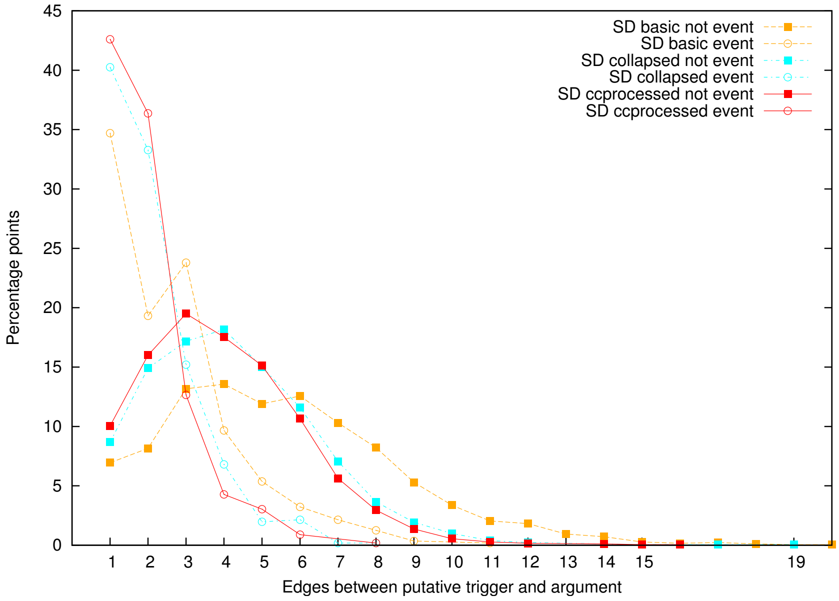


Figure 5.4: `Transcription` event, distances between trigger word and proteins in *basic* and *collapsed/ccprocessed* SD dependency trees.

of distances of true events (1.9) is higher than for the `Phosphorylation` event (1.6) and the μ of distances of false events (4.0) is lower than for the `Phosphorylation` event (4.3). This explains the higher ratio of μ values for the `Transcription` event in comparison with the one for the `Phosphorylation` event. When the ranking of F-scores of the events (see Table 5.9) is compared with the distribution in Table 5.16, we can see that events with a lower ratio have a higher F-score than events with a higher ratio. In general, two complexity groups were detected, one with a ratio of around 0.4 and the other with a ratio of around 0.5. The μ of distances of true and false event paths is thus translated into the F-scores measured for the events in question (Section 5.2.4).

This analysis suggests that the closer the true arguments and the further the false arguments are positioned from their respective triggers, the easier it is to extract the corresponding events. When these findings are transferred to dependency graph trimming, one might then estimate the benefit of disambiguation strategies and trimming of dependency paths between triggers and associated arguments.

Furthermore, impacts of various dependency graph representations can be illustrated with the help of Self-Organizing Maps (SOM).[7] SOMs (Kohonen (2000)) are used to map high-dimensional vectors into a space of lower dimensionality, e.g., 2D. Data vectors are mapped to a grid of units which are typically arranged in a rectangular or hexagonal fashion. Thus, each grid unit constitutes a kind of cluster representing all feature vectors mapped to this unit. Various feature-based models built upon several dependency representations on the development data set were visualized (Figures 5.5 and 5.6), i.e., SD *basic*, SD *ccprocessed* and CoNLL. We can see that most of the instances after the trimming step (which are shown here in green) are mapped onto units distant from their original placement on the map. Apparently, the simplification moves the samples in the feature space, potentially allowing more accurate classification models to be learned. This visualization illustrates the fact that feature spaces of dependency feature-based models of JReX built upon different dependency graph representations can differ. In order to show and measure the influence of dependency graph trimming on event extraction performance, I completed an in-depth evaluation study on the Shared Task data sets presented in the next section.

[7] The R Kohonen package (Wehrens and Buydens (2007)) was used to train our SOMs.

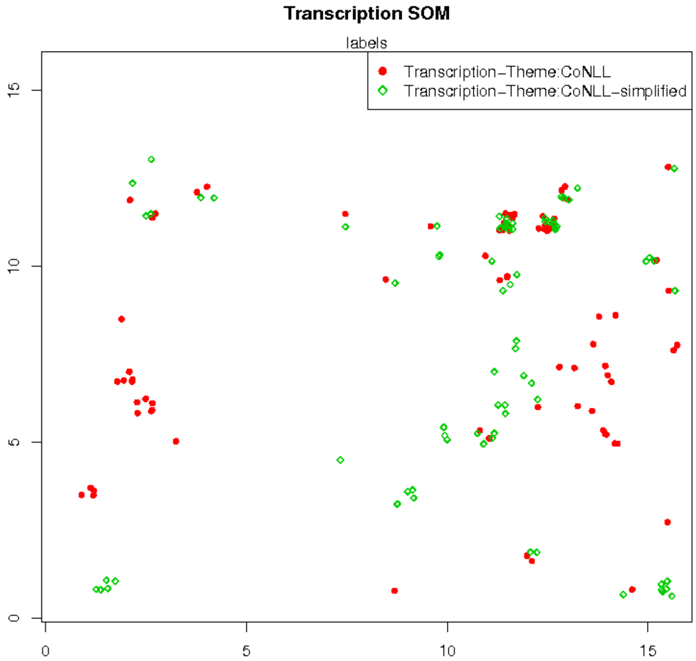


Figure 5.5: `Transcription` event, Self-Organizing Maps visualization of two dependency feature-based models built upon various parses of the Shared Task development data (1) CoNLL basic (2) CoNLL simplified/trimmed.

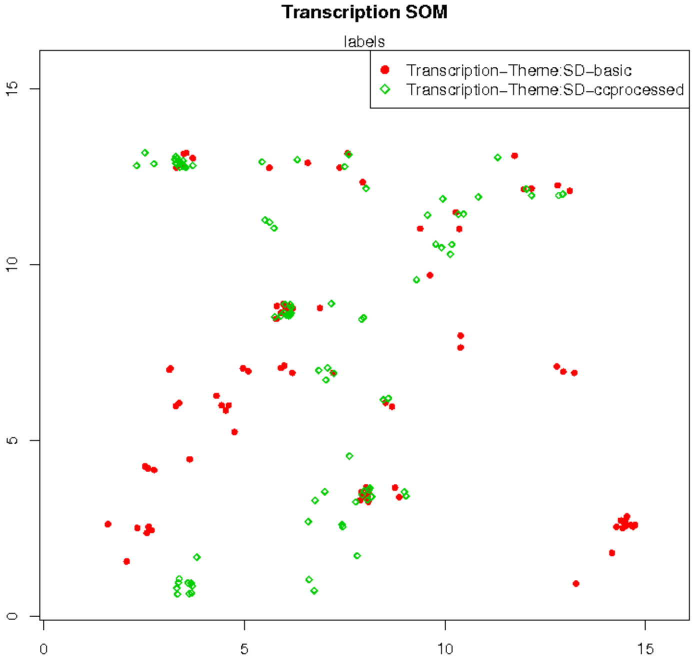


Figure 5.6: `Transcription` event, Self-Organizing Maps visualization of two dependency feature-based models built upon various parses of the Shared Task development data (1) SD *basic* (2) SD *ccprocessed*.

5.3.2 Evaluation of Dependency Graph Representations and Trimming

In this section I describe the results and outcomes of the experiments of event extraction tasks based on different dependency graph representations from various state-of-the-art parsers. For these experiments the following top-performing parsers are selected:

- **Bikel**, Bikel parser Bikel (2004) with the WSJ-trained parsing model.
- **C+J**, Charniak and Johnson's re-ranking parser Charniak and Johnson (2005), with the WSJ-trained parsing model.
- **M+C**, Charniak and Johnson's re-ranking parser Charniak and Johnson (2005), with the self-trained biomedical parsing model from McClosky (2010).
- **GDep** Sagae and Tsujii (2007), a native dependency parser developed specifically for parsing biomedical data.
- **MALT** Nivre et al. (2007), a native dependency parser re-trained on the Genia Treebank (Tateisi et al., 2005).
- **MST** parser (McDonald et al. (2005)), a native dependency parser re-trained on the Genia Treebank (Tateisi et al. (2005)).

Thus, in this study I use a set of six parsers, three of them constituency-based and three dependency-based. These parsers can output different forms of dependency representations.

For this study, the parsers were re-trained on the biomedical treebank data in the form of various dependency graph representations. The native dependency parsers were re-trained on the Genia Treebank (Tateisi et al., 2005) conversions.[8] The Genia Treebank conversions,[9] i.e., Stanford *basic*, CoNLL'07 and CoNLL'08, have been produced with available conversion scripts. For the Stanford dependency conversion, the Stanford parser tool[10] was used; for CoNLL'07 and CoNLL'08, the treebank-to-CoNLL conversion scripts[11] were used, as made available by the CoNLL'X Shared Task organizers. The phrase-structure-based parsers were applied with models that were already available, i.e., the Bikel and C+J parsers trained on the WSJ part of the Penn Treebank corpus (Marcus et al., 1993), and M+C trained on the Genia Treebank corpus. For these experiments, I converted the prediction results of

[8] For the training of dependency parsers I used only Stanford *basic* from the available Stanford conversion variants. The *collapsed* and *ccprocessed* variants do not provide dependency trees and are not recommended for training native dependency parsers.

[9] I used the Genia Treebank version 1.0, available from `www-tsujii.is.s.u-tokyo.ac.jp`.

[10] `http://nlp.stanford.edu/software/lex-parser.shtml`

[11] `http://nlp.cs.lth.se/software/treebank_converter/`

Table 5.17: Best configurations for dependency representations for event extraction task on the Shared Task development data. Recall (R), precision (P), and F-score (F) values are provided.

Event Class	Best Parser	Best Configuration	R	P	F
Gene Expression	MST	CoNLL'08, *aux, coords*	79.5	81.8	80.6
Transcription	MALT	CoNLL'07, *aux, coords*	67.1	75.3	71.0
Protein Catabolism	MST	CoNLL'08, *preps*	85.7	100	92.3
Phosphorylation	MALT	CoNLL'08	80.9	88.4	84.4
Localization	MST	CoNLL'08, *aux*	81.1	87.8	84.3
Binding	MST	CoNLL'07, *aux, coords, np action*	51.2	51.0	51.1
Regulation	MALT	CoNLL'07, *aux, coords*	30.8	49.5	38.0
Positive Regulation	M+C	CoNLL'07	43.0	49.9	46.1
Negative Regulation	M+C	CoNLL'07	49.5	45.3	47.3

Table 5.18: Effects of trimming of CoNLL dependencies on the Shared Task development data for Binding events. Approximate Span Matching / Approximate Recursive Matching. The data was processed by the MST parser. Recall (R), precision (P), and F-score (F) values are provided.

Binding	R	P	F
CoNLL'07	47.3	46.8	47.0
CoNLL'07 *aux, coords*	46.8	48.1	47.4
CoNLL'07 *aux, coords, np action*	51.2	51.0	**51.1**

the phrase-structure-based parsers into five dependency graph representations, *viz.* Stanford *basic*, Stanford *collapsed*, Stanford *ccprocessed*, CoNLL'07 and CoNLL'08, using the same scripts as for the conversion of the Genia Treebank.

The JReX argument extraction tool was retrained on the Shared Task data enriched with different outputs of syntactic parsers, as described above. The results of the event extraction task are represented in Table 5.19. I provide here the summarized results of important event extraction subtasks, i.e., results for basic events (Gene Expression, Transcription, Localization, Protein Catabolism) are summarized under SVT-TOTAL; regulatory events are summarized under REG-TOTAL; and the overall extraction results are listed in ALL-TOTAL (Table 5.19).

The event extraction system trained on various dependency representations obviously produces very different results. The difference in terms of F-score is a maxi-

Table 5.19: JReX results on the Shared Task development data. Approximate Span Matching/Approximate Recursive Matching. Recall (R), precision (P), and F-score (F) values are provided.

Parser	SD basic			SD collapsed			SD ccprocessed			CoNLL’07			CoNLL’08		
SVT-TOTAL															
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Bikel	70.5	75.5	**72.9**	70.7	74.5	72.5	71.6	73.5	72.5	69.4	75.9	72.5	69.7	75.7	72.6
C+J	73.0	77.4	75.1	73.2	77.3	75.2	72.8	77.2	75.0	73.5	78.3	**75.8**	73.0	77.9	75.4
M+C	76.4	78.0	77.2	76.4	77.6	77.0	76.4	77.2	76.8	76.4	79.0	77.7	76.6	79.3	**77.9**
GDEP	77.1	77.5	**77.3**	N/A	N/A	N/A	N/A	N/A	N/A	72.5	80.2	76.1	72.6	77.2	74.8
MALT	73.1	78.2	75.6	N/A	N/A	N/A	N/A	N/A	N/A	75.9	80.3	**78.0**	73.7	78.2	75.9
MST	76.4	78.5	77.4	N/A	N/A	N/A	N/A	N/A	N/A	74.8	78.4	76.6	76.7	80.8	**78.7**
REG-TOTAL															
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Bikel	35.3	40.6	**37.8**	33.8	40.3	36.8	34.3	39.6	36.8	33.9	39.2	36.3	34.0	41.0	37.2
C+J	36.2	41.8	38.8	37.3	41.8	39.4	36.5	41.9	39.0	38.1	43.9	**40.8**	37.4	44.0	40.4
M+C	39.4	45.5	42.3	38.8	45.3	41.8	38.5	43.7	40.9	41.9	47.4	**44.5**	40.1	47.9	43.7
GDEP	39.6	42.8	41.6	N/A	N/A	N/A	N/A	N/A	N/A	38.4	43.7	40.9	39.8	44.4	**42.0**
MALT	38.8	44.3	41.4	N/A	N/A	N/A	N/A	N/A	N/A	39.0	44.3	41.5	39.2	46.4	**42.5**
MST	39.5	43.6	41.4	N/A	N/A	N/A	N/A	N/A	N/A	39.6	45.6	42.4	40.6	45.8	**43.0**
ALL-TOTAL															
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Bikel	47.4	51.5	**49.4**	46.3	50.8	48.5	46.9	50.2	48.5	44.8	50.7	47.6	44.7	51.8	48.0
C+J	49.3	53.8	51.5	49.6	52.8	51.2	49.0	53.0	50.9	50.3	54.4	**52.3**	49.5	54.3	51.8
M+C	52.3	56.4	54.3	51.8	55.7	53.7	51.3	54.3	52.8	53.2	57.5	**55.3**	52.2	58.2	55.0
GDEP	52.7	54.5	**53.6**	N/A	N/A	N/A	N/A	N/A	N/A	50.6	55.2	52.8	51.3	55.0	53.1
MALT	50.4	54.7	52.4	N/A	N/A	N/A	N/A	N/A	N/A	51.5	56.0	53.7	51.2	56.8	**53.8**
MST	52.3	54.8	53.5	N/A	N/A	N/A	N/A	N/A	N/A	51.7	56.4	53.9	52.4	56.9	**54.6**

Table 5.20: Results on the Shared Task development data. Approximate Span Matching/Approximate Recursive Matching. Recall (R), precision (P), and F-score (F) values are provided.

	JReX (Julie Lab)			JReX (Julie Lab)			Tokyo		
	(M+C, CoNLL'08)			Final Configuration					
Event Class	R	P	F	R	P	F	R	P	F
`Gene Expression`	79.2	80.3	79.8	79.5	81.8	80.6	78.7	79.5	79.1
`Transcription`	59.8	72.0	65.3	67.1	75.3	71.0	65.9	71.1	68.4
`Protein Catabolism`	76.2	88.9	82.0	85.7	100	92.3	95.2	90.9	93.0
`Phosphorylation`	83.0	81.2	82.1	80.9	88.4	84.4	85.1	69.0	76.2
`Localization`	77.4	74.6	75.9	81.1	87.8	84.3	71.7	82.6	76.8
SVT-TOTAL	76.4	79.0	**77.7**	78.2	82.6	**80.3**	77.3	77.9	**77.6**
`Binding`	45.6	45.9	45.8	51.2	51.0	51.1	50.8	47.6	49.1
EVT-TOTAL	66.9	68.7	**67.8**	69.9	72.5	**71.2**	69.1	68.1	**68.6**
`Regulation`	32.5	46.2	38.2	30.8	49.5	38.0	36.7	46.6	41.1
`Positive Regulation`	42.3	49.0	45.4	43.0	49.9	46.1	43.9	51.9	47.6
`Negative Regulation`	48.5	44.0	46.1	49.5	45.3	47.3	38.8	43.9	41.2
REG-TOTAL	41.9	47.4	**44.5**	42.2	48.7	**45.2**	41.7	49.4	**45.2**
ALL-TOTAL	53.2	57.5	**55.3**	54.7	60.0	**57.2**	54.1	58.7	**56.3**

mum of 2.4 percentage points for the SVT-TOTAL) events (see MALT parser, difference between SD *basic* (75.6% F-score) and CoNLL'07 (78.0% F-score)), a maximum of 3.6 points for REG-TOTAL (see M+C parser, difference between SD *ccprocessed* (40.9% F-score) and CoNLL'07 (44.5% F-score)) and a maximum of 2.5 points for ALL-TOTAL (see M+C parser, difference between SD *ccprocessed* (52.8% F-score) and CoNLL'07 (55.3% F-score)).

The top three event extraction results on the development data based on different syntactic parsers results were achieved with M+C parser – CoNLL'07 representation (55.3% F-score), MST parser – CoNLL'08 representation (54.6% F-score) and MALT parser – CoNLL'08 representation (53.8% F-score) (Table 5.19, ALL-TOTAL). The CoNLL'08 and CoNLL'07 format clearly outperformed Stanford representations on all event extraction tasks. Stanford dependencies seem to be useful here only in the *basic* mode. The *collapsed* and *ccprocessed* modes produced even worse results for the event extraction tasks. This will be discussed later in this section.

Table 5.21: Results on the Shared Task test data. Approximate Span Matching/Approximate Recursive Matching. Recall (R), precision (P), and F-score (F) values are provided.

	JReX (Julie Lab) Buyko et al. (2011a)			JReX (Julie Lab) Final Configuration			Tokyo		
Event Class	R	P	F	R	P	F	R	P	F
`Gene Expression`	66.3	79.6	72.4	67.0	77.2	71.8	68.7	79.9	73.9
`Transcription`	33.6	61.3	43.4	35.0	60.8	44.4	54.0	60.7	57.1
`Protein Catabolism`	71.4	90.9	80.0	71.4	90.9	80.0	42.9	75.0	54.6
`Phosphorylation`	80.0	85.0	82.4	80.7	84.5	82.6	84.4	69.5	76.3
`Localization`	47.7	93.3	63.1	45.4	90.8	60.5	47.1	86.3	61.0
SVT-TOTAL	61.4	80.3	**69.6**	61.8	78.2	**69.0**	65.3	76.4	**70.4**
`Binding`	47.3	52.4	49.7	47.3	52.2	49.6	52.2	53.1	52.6
EVT-TOTAL	58.2	73.1	**64.8**	58.5	71.7	**64.4**	62.3	70.5	**66.2**
`Regulation`	24.7	40.5	30.7	26.8	38.2	31.5	28.9	39.8	33.5
`Positive Regulation`	35.8	45.4	40.0	34.8	45.8	39.5	38.0	48.3	42.6
`Negative Regulation`	37.2	39.7	38.4	37.5	40.9	39.1	35.9	47.2	40.8
REG-TOTAL	34.2	43.2	**38.2**	34.0	43.3	**38.0**	35.9	46.7	40.6
ALL-TOTAL	45.7	57.6	**51.0**	45.8	57.2	**50.9**	48.6	59.0	**53.3**

My second experiment focused on trimming operations on CoNLL dependency graphs. Here I performed event extraction after trimming the dependency trees in different modes: *coords* – restructuring coordinations; *preps* – collapsing of prepositions; *aux* – propagating dependency relations of auxiliaries and modals to main verbs; *np action* – restructuring of noun phrases containing action adjectives (cf. Section 4.5.2). My second experiment showed that the extraction of selected events can benefit in particular from the trimming procedures *coords* and *aux*, but there is no evidence for a general trimming configuration for the overall event extraction task. Table 5.17 summarizes the best configurations that were found for the events under consideration. We can see that the CoNLL'08 and CoNLL'07 dependencies modified for auxiliaries and coordinations are the best configurations for four events (out of nine). For three events no modifications are necessary and only one event benefits from trimming of prepositions (`Protein Catabolism`). Only the `Binding` event benefits significantly from noun phrase modifications (Table 5.18). The increase in F-score for trimming procedures is 4.1 percentage points for `Binding` events.

In my next experiment the best configurations for each of the Shared Task events were connected as presented in Table 5.17. The overall event extraction results of this final configuration are shown (Tables 5.20 and 5.21). JReX achieved an increase of 1.9 percentage points F-score in the overall event extraction compared to the best performing single parser configuration (M+C, CoNLL'07) (Table 5.20, ALL-TOTAL). The reported results on the development data outperform the results of the Tokyo system by 2.6 percentage points F-score for all basic events including `Binding` events (Table 5.20, EVT-TOTAL) and by 0.9 percentage points in the overall event extraction task (Table 5.20, ALL-TOTAL). On the test data JReX achieved an F-score similar to the current JReX system trained on modified CoNLL'07 dependencies from the MST parser (Table 5.21, ALL-TOTAL). Results on the official test data reveal that the performance differences between various parsers may not play such a characteristic role as different dependency representations.[12] My empirical findings that the best performance results could only be achieved by event-specific dependency graph configuration, reveal that syntactic representation of different semantic events vary considerably at the level of dependency graph complexity and that the automatic prediction of such syntactic structures can vary from one dependency parser to another.

The evaluation results presented in this section show that an increased F-score is basically due to a better performance in terms of precision (Table 5.19). For example, the M+C evaluation results in the Stanford *basic* mode provide an increase in precision of 2 percentage points compared to the Stanford *ccprocessed* mode. Therefore, the focus here is on the analysis of false positives that the JReX system extracts in various modes.

The first analysis used the outputs of systems based on the M+C parsing results. I scrutinized the Stanford *basic* and *ccprocessed* false positives and compared the occurrences of dependency labels in two data sets, namely the intersection of false positives from both system modes (set A) and the false positives produced only by the system with a worse performance (set B, *ccprocessed* mode). About 70% of all false positives are contained in set A. My analysis revealed that some dependency labels have a higher occurrence in set B, e.g., `nsubjpass, prep_on, prep_with, prep_in, prep_for, prep_as`. Some dependency labels occur only in set B, such as `agent`, `prep_unlike, prep_upon`. It seems that collapsing some prepositions, such as *"with", "in", "for", "as", "on", "unlike", "upon"*, does not have a positive effect on the extraction of argument structures. In a second step, the Stanford *basic* and CoNLL'07 false positives sets were compared. The false positives of both systems have an intersection of about 70%. I also compared the intersection of false positives between two outputs (set A) and the set of additional false positives of

[12] Trimmed CoNLL dependencies are used in both system configurations.

the system with worse results (Stanford *basic* mode, set B). Dependency labels such as `abbrev, dep, nsubj, nsubjpass` occur more frequently in set B than in set A. This analysis provides evidence of the distinction between `nsubj` and `nsubjpass` not having been properly learned for event extraction.

The second analysis round took the outputs of the MST parsing results. As in the previous experiments, I compared false positives from two mode outputs, here the CoNLL'07 mode and the CoNLL'07 modified for *auxiliaries* (see *aux* procedure in Section 4.5.2) and *coordinations* (see *coords* procedure Section 4.5.2) mode. The false positives have an intersection of 75%. Dependency labels such as `VC, SUBJ, COORD`, and `IOBJ` occur more frequently in the additional false positives from the CoNLL'07 mode than in the intersection of false positives from both system outputs. It is clear that trimming auxiliary and coordination structures in CoNLL dependency graphs has a direct positive effect on argument extraction, reducing false positives numbers, especially with corresponding dependency labels in shortest dependency paths.

The presented analysis of false positives shows that the distinction between active and passive subject abbreviation labels, as well as collapsing prepositions in the Stanford dependencies, could not have been properly learned by the JReX argument extraction component, leading to an increased rate of false positives. The trimming of auxiliary structures and the subsequent coordination collapsing on CoNLL'07 dependencies does indeed have event-specific positive effects on event extraction.

In a further study on trimming procedure, I measured the performance of configurations with semantic enrichment in comparison with those without (Table 5.22), using three-round cross-validation on training data. I selected for this study training data as no significant semantic enrichment effects could be measured on the development data. The study distinguished between basic and full semantic enrichment modi. The basic configuration without enrichment is based on stemming and normalization of gene and protein representations with a common placeholder `Gene`. The full semantic enrichment is based on the basic enrichment and on the normalization of other entities, event triggers, experimental methods and GOA annotations (Section 4.5.2). The F-scores show that events can benefit from semantic enrichment strategies, in particular from entity normalization — `Localization`, `Binding`, `Phosphorylation`, from trigger normalization — `Binding`, `Protein Catabolism`, and from enrichment by experimental methods and GO terms — `Protein Catabolism`, `Phosphorylation`. I found significant differences ($p \leq 0.05$) only for the `Localization` and `Binding` events. For most of the remaining events the benefit is statistically not significant. In my view this might be explained by the high variance in the cross-validation results (the mean value of F-score variance for all events is about 7 percentage points).

Table 5.22: Performance of models with semantic trimming. Results of three rounds of a ten-fold cross-validation on the Shared Task training data of the JREX system as used in the official run. GDEP parser has been applied. In the *basic* semantic trimming mode, gene and protein names are replaced with a placeholder *gene*.

Event Class	***Basic (Gene/Protein)***			***Entity and MeSH***			***Event Trigger***		
	recall	prec.	F-score	recall	prec.	F-score	recall	prec.	F-score
Gene Expression & Transcription	81.3	84.7	83.0	81.0	84.6	82.8	80.6	86.6	83.5
Protein Catabolism	81.0	84.1	82.5	80.1	84.7	82.4	85.5	84.8	85.1
Phosphorylation	86.0	86.5	86.2	87.8	86.8	87.3	87.8	86.8	87.3
Localization	64.2	78.0	70.4	67.1	77.4	71.9	62.3	82.0	70.8
Binding	44.2	65.0	52.6	47.9	61.3	53.8	44.7	67.5	53.8
Regulation	60.8	72.6	66.2	63.7	70.7	67.0	62.2	72.6	67.0
Positive Regulation	64.3	75.9	69.6	64.7	76.3	70.0	64.6	76.7	70.1
Negative Regulation	68.2	78.0	72.8	68.0	78.0	72.7	69.2	77.3	73.0
Event Class	***Methods***			***Gene Ontology***			**All Layers**		
	recall	prec.	F-score	recall	prec.	F-score	recall	prec.	F-score
Gene Expression & Transcription	81.0	84.2	82.6	80.8	84.6	82.7	80.3	86.8	83.4
Protein Catabolism	84.6	85.4	85.0	84.6	85.4	85.0	81.9	85.8	83.8
Phosphorylation	89.0	84.4	86.7	88.4	85.8	87.1	87.2	87.2	87.2
Localization	64.9	76.1	70.1	74.9	75.8	75.3	65.3	83.0	73.1
Binding	43.0	67.0	52.4	43.8	66.3	52.8	50.8	62.2	55.9
Regulation	61.9	71.9	66.5	63.3	70.4	66.7	64.5	71.1	67.6
Positive Regulation	64.2	77.4	70.2	66.7	74.5	70.4	65.4	76.6	70.6
Negative Regulation	69.2	77.8	73.2	68.2	78.0	72.8	67.8	78.9	72.9

This section investigated the role that different dependency representations may play in accomplishing the event extraction task exemplified by biological events. Diverse representation formats (Stanford *vs.* CoNLL) were experimentally compared within diverse parsers (e.g., Bikel, M+C, GDep, MST, MALT). From these experiments I conclude that the dependency graph representation has a crucial impact on the achievement of event extraction task. The CoNLL dependencies outperform the Stanford dependencies for four out of six parsers. With additionally trimmed CoNLL dependencies JReX achieved an F-score of 50.9% on the official test data and an F-score of 57.2% on the official development data of the "BioNLP 2009 Shared Task on Event Extraction" (Table 5.21, ALL-TOTAL). These findings confirm my hypothesis that dependency graph representation variants have an effect on semantic event extraction task and that trimming helps.[13]

Although the main focus of the study presented in this section has been on the evaluation of effects of different dependency graph representations on IE task achievement (here the task of event extraction), this analysis also targeted the task-oriented evaluation of top-performing syntactic parsers. The results of this work indicate that the Genia-trained parsers, i.e., M+C parser, the MST, MALT, and GDep, are a reasonable basis for achieving state-of-the-art performance in biomedical event extraction.

However, the choice of the most suitable parser should also take into account its performance in terms of parsing time. Cer et al. (2010) and Miyao et al. (2008) show in their experiments that native dependency parsers are faster than constituency-based parsers. Experiments by Miyao et al. (2008) show that native dependency parsers are up to 19 times faster than constituency-based parsers (here KSDEP *vs.* M-J Re-rank). My experiments relating to parsing time on the Shared Task data point in the same direction.[14] When it comes to scaling up event extraction to huge biomedical document collections, such as Medline, the selection of a parser is mainly influenced by its run-time performance. MST, MALT and GDep parsers, or the M+C parser with reduced re-ranking (Cer et al., 2010), would thus be an appropriate choice for large-scale event extraction under these constraints.[15] This is why native dependency parsers such as the GDep and the MST parser were selected for the JReX system.

In summary, the evaluation studies presented in this chapter confirmed the shortest path hypothesis of Bunescu and Mooney (2007) (Section 5.2.5) and thus validated

[13] Still, we have to bear in mind that the optimal configurations detected in my studies on the Shared task data are not guaranteed to carry across to different ML approaches.

[14] The GDep parser processed the Shared Task training data in 540 CPU seconds, whereas the Stanford parser took 6,600 seconds for the same data set on an Intel(R) Core(TM)2 Duo CPU E6850.

[15] For large-scale experiments an evaluation of the M+C with reduced re-ranking should be provided.

the application of shortest path informations in the JReX system. Furthermore, it was demonstrated that shortest path representations in the form of trimmed dependency graph variants may have a positive effect on the performance in solving event extraction task (Section 5.3.2). The application of this knowledge enhanced the performance of the JReX system which is now a state-of-the-art system for event extraction in the biomedical domain. Nevertheless, the question arises where the JReX system fails. I provide insights to answer this question in the next section.

5.4 Error Discussion and Outlook

In order to obtain some clues where JReX fails, an expert biologist analyzed 30 abstracts randomly extracted from the erroneous data from the development set. Seven groups of errors were determined based on this analysis. The first group contains examples for which an event should be determined, but a false argument was found (e.g., `Binding` arguments were not properly sorted, or correct and false arguments were detected for the same trigger) (44 examples). The second group comprises examples where no trigger was found (23 examples). Group (3) contains cases where no events were detected, although a trigger was properly identified (14 examples). Group (4) consists of examples detected in sentences which did not contain any events (12 examples). Group (5) lists biologically meaningful analyses, actually very close to the gold annotation, especially for the cascaded regulatory events (12 examples), while Group (6) incorporates examples of a detected event with incorrect type (1 example). Group (7) gathers together misleading gold annotations (10 examples).

This assessment clearly indicates that a major source of errors can be traced to the level of argument identification, in particular for `Binding` events. For example from the sentence "*In coimmunoprecipitation experiments using transfected COS cells, GATA-1 and ER* ***associate*** *in a ligand-dependent manner.*" JReX extracted two `Binding` events with *GATA-1* and *ER* as `Theme` arguments respectively. Thus, JReX failed to associate these two entities in a single `Binding` event. The second major source is at the level of trigger detection (JReX ignored, for example, triggers such as "*in the presence of*", "*when*", "*normal*"). The third major group of errors is due to the level of event detection given a detected putative event trigger. In such a case, e.g., anaphoric mentions of entities are used. For example in the sentence "*In contrast, treatment of HL-60 cells with retinoic acid or DMSO, which results in a granulocytic differentiation of these cells, decreases 4E-BP1 amount without affecting its* ***phosphorylation*** *and strongly increases 4E-BP2 amount*" JReX failed to extract a `Phosphorylation` event with the `Theme` *4E-BP1*. The latter entity is mentioned in this event description by an anaphoric "*its*". About 10% of the errors are due to a slight difference between extracted events and gold events. For example, in the phrase "***role** for NF-kappaB in the **regulation** of FasL **expression***" JReX could not extract the gold event `Regulation` (`Regulation (Gene Expression (FasL))`) associated with the trigger "*role*", but JReX was able to find the (inner) event `Regulation (Gene Expression (FasL))` associated with the trigger "*regulation*". Interestingly, the typing of events is not an error source in spite of the simple disambiguation approach. As Group (6) is an insignificant source of errors in our randomly selected data, the biologist focused the error analysis on the especially ambiguous event type `Transcription`. He found that 14 of 34 errors were due to the disambiguation strategy (in particular for triggers "*(gene) expression*" and "*induction*").

This error analysis could provide some insights into the cases where the JReX fails to solve event extraction task. In order to address these errors, the JReX system might be refined, on the one hand, for the integrated resources in form of refined trigger dictionaries, and, on the other hand, extended with new NLP components such as anaphora resolution tools. The gold standard data might be analysed and improved for misleading annotations. However, the major error source at the level of argument detection proper remains untreated as no evident hints to improve this task can be identified from this error analysis. Therefore, the question arises whether JReX can effectively learn argument extraction with all its intricacies given the amount of annotated data from the Shared Task.

In order to address this issue, I produced learning curves for all Shared Task events. These learning curves have been generated on F-score values from ten rounds of JReX predictions on the development data. In each round, JReX argument extraction component learns on a subsection of training data and is evaluated on the complete development data. The size of a training subsection increases after each round by a size of one tenth (80 documents). At the end of a learning process in the last round, JReX learns on the complete training data (800 documents). In order to provide reliable evaluation data, the learning process was evaluated five times given different randomly identified subsections of the training data. After that, mean values of achieved F-score values have been used for the generation of learning curves. For curve generation I applied the logarithmic function given the size of a training section (documents) and F-score mean values. The logarithmic function reflects well the Shared Task evaluation data for every Shared Task event type.

The learning curves of two selected events, e.g., the `Protein Catabolism` and `Positive Regulation` are represented in Figures 5.7 and 5.8. The remaining figures can be found in Appendix (Section A.3). The visualization demonstrates that JReX performance continually increases, while variation between single F-score values (from various randomly produced rounds) drops. It is interesting that the learning curves

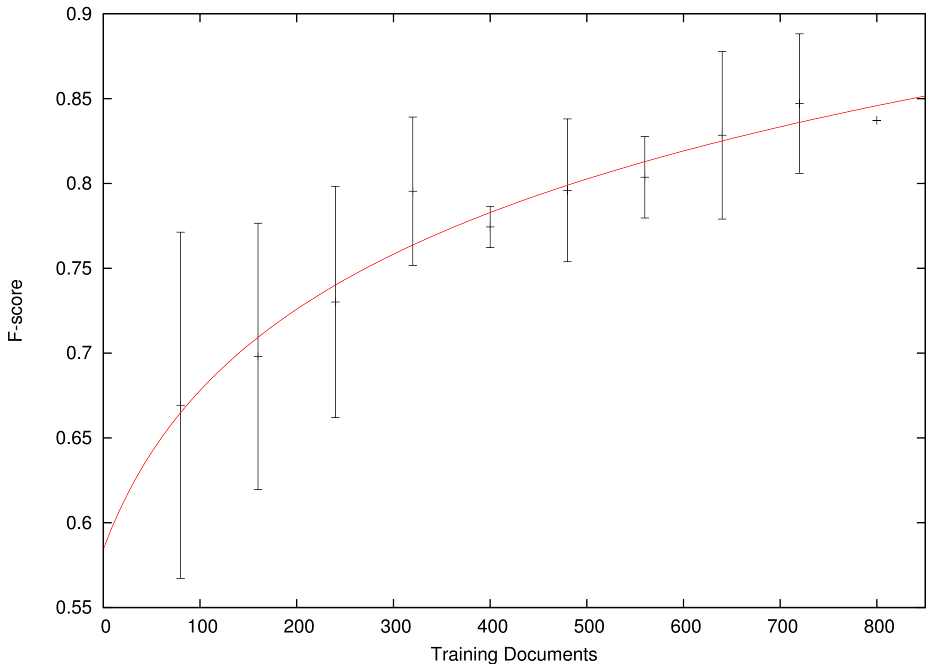


Figure 5.7: Learning curve for argument extraction in `Protein Catabolism` events on the Shared Task development data. Logarithmic function is applied.

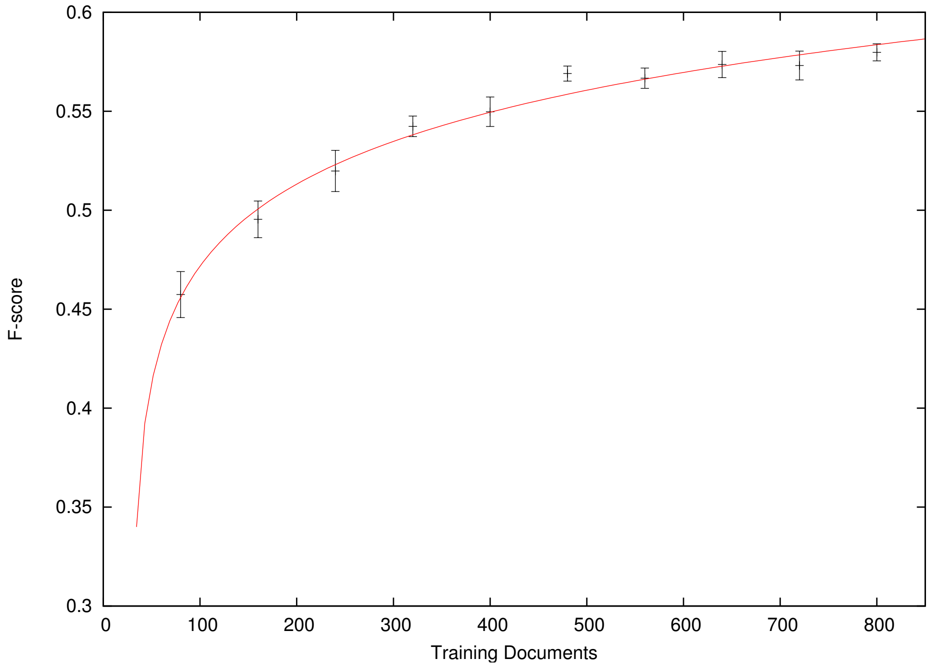


Figure 5.8: Learning curve for argument extraction in `Positive Regulation` events on the Shared Task development data. Logarithmic function is applied.

show a learning progress for all Shared Task events, even for simple events such as `Protein Catabolism` event. None of the Shared Task event curves do flatten off. The trajectory of curves is a good indicator for learning of the argument extraction component. Thus, the JReX argument extraction component continually learns from the training data and there is a room for further learning of argument structures. These statistics demonstrate that the current amount of training data is not sufficient for learning all intricacies of event argument structures. The size of event annotation data should be increased. However, these statistics clearly indicate that the logarithmic function fits the argument learning process. That means, that a further increase of F-score values will require much more training data (exponential increase). As the manual annotation is a time-consuming process, it might be supported by Active Learning (cf. e.g., Tomanek and Hahn (2009)) or a distant supervision approach (cf. e.g., Mintz et al. (2009)) (see discussion in Section 7).

Chapter 6

Extrinsic Evaluation of JReX on Curated Databases

The previous chapter focused on exploring the feasibility of the event extraction task in the biomedical domain. The JReX system has been extensively evaluated on the publicly available event annotated corpus, which is widely accepted in the BioNLP community (see Chapter 5). JReX came second in the official Task 1 of the "BioNLP 2009 Shared Task on Event Extraction" and, after the challenge, the performance gap between it and the best system was narrowed by updating various JReX functionalities. Moreover, the complete JReX system's performance was ranked first in the event extraction task in the U-Compare event server framework (Kano et al., 2011). Thus, the JReX system has been shown to be able to solve the event extraction task in a competitive way. However, the type of evaluation considered up to this chapter was performed *intrinsically*, i.e., under clean lab conditions. It was done on a limited document set with restrictions in the form of corpus annotation guidelines, test corpus size and variety.

Although intrinsic evaluations are of crucial significance, the benefits seem to be limited for the life science community, which works with large-scale life science data sets and has a strong need for semantically rich fact repositories (for a survey of biological databases, currently 1,330, cf. the *Nucleic Acids Research* Online Database Collection[1]). The creation and curation of large fact repositories require skilled expert biologists to produce data abstractions in a time-consuming and labor-intensive process (cf. Section 2.1), particularly when complex decisions are made, such as pathway generation.[2] Due to human resource constraints and time limits, manual curation will inevitably lead to incomplete, and in many cases biased, knowledge repositories because only a tiny fraction of the relevant literature can be processed

[1] Accessible via `http://www.oxfordjournals.org/nar/database/a/`.

[2] The KEGG database (accessible via `http://www.genome.jp/kegg/pathway.html/`.) has at its disposal a range of manually curated molecular pathways, and is a prominent example of manual curation efforts.

properly by human curator teams. This thesis advocates an automatic curation approach for the biomedical fact repositories using the JReX system for interpreting the semantics of biological literature. Therefore, the question arises how robust an advanced event extraction system such as JReX is in large-scale information extraction applications from the perspective of coverage and reliability of extracted data? This chapter explores the robustness of JReX in a real-life *extrinsic* evaluation scenario targeting database curation. It examines the recall and precision of the JReX system and the reliability, and novelty of extracted events.

The JReX system is exposed in this chapter to the knowledge extraction about gene expression regulation that is the subject of ongoing major research in molecular biology affecting a large number of research domains, and is a core field of future research. Gene expression regulation is a complex cellular process and can be described as the process that modulates the frequency, rate or extent of gene expression, where transcription factors play a central role in the transcription of genes into their RNA and subsequent translation into proteins (cf. Section 3.5). This chapter presents two evaluation studies on automatic curation of gene expression regulation events using JReX. For the first study I chose the fact database RegulonDB (Gama-Castro et al., 2011), which is the world's largest manually curated reference database for the transcriptional regulation network of *E. coli*. RegulonDB contents are manually gathered from the scientific literature. With this database as a gold standard, I investigate the performance of automatic RegulonDB re-creation by processing relevant literature sources with the help of the JReX system (Section 6.2). In the second study, I focus on a construction of a smaller regulatory network for the human pathogenic microorganism fungus *Candida albicans* curated by the Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute in Jena (Section 6.3).

6.1 Related Work

With regards to event extraction research in the biomedical domain, to the best of my knowledge there are only few studies which deal exclusively with gene expression regulation events. The Genic Interaction Extraction Challenge (Nédellec, 2005) was organized to determine the state-of-the-art performance of systems designed for detecting gene regulation interactions. The best system, which was based on patterns (learned with finite state automata), achieved a performance of about 50% F-score (Hakenberg et al., 2005) in this competition. In other studies, such as one by Yang et al. (2008), the focus is on detecting sentences that contain mentions of transcription factors (proteins regulating gene expression). Sáric et al. (2004) extract gene regulatory networks and achieve accuracy of up to 90% in this task. They use syntactico-semantic rules, which are based on subcategorization and argument selectional restrictions of verbs relevant to gene expression regulation, such as "*activate*" or "*express*". Sáric et al. (2004) disregard, however, ambiguous instances that potentially result in lower recall (no recall measures are reported in this work). Sasaki et al. (2008b) investigate the extraction of semantic frames for gene expression regulation verbs. They developed a CRF-based event extraction system on the GREC corpus (Thompson et al., 2009), and report recall of 18.6% and precision of 49.0% in a ten-fold cross-validation on this corpus. Kim and Rebholz-Schuhmann (2010) explore usage of the Gene Regulation Ontology ontology and apply hand-crafted inference rules for the extraction of gene expression regulation events. The event extraction system is based on syntactico-semantic rules matching the text for mentions of *explicit* events relevant to gene expression regulation. The inference rules combine mentions of explicit events that have been discovered and domain knowledge from GRO for the extraction of implicit (compositional) events (cf. Section 3.5.3 for the explanation of event nesting and Section 3.5.2 for the explanation of macro and micro views on events). They evaluate the system on 209 Medline abstracts and achieve results of 21.1% recall and 84.0% precision.

The success of text mining tools for automatic database generation complementing the work of human database curators has already been tested in an extrinsic evaluation scenario (e.g., Kim and Rebholz-Schuhmann (2010); Rodríguez-Penagos et al. (2007); Hahn et al. (2009); Buyko and Hahn (2011); Buyko et al. (2011b)). Rodríguez-Penagos et al. (2007) used the approach of Sáric et al. (2004) for a first large-scale automatic reconstruction of RegulonDB. The best results were achieved on the RegulonDB version of June 2006 with 45% recall and 77% precision. Kim and Rebholz-Schuhmann (2010) complement an intrinsic evaluation study (see above) with an external evaluation on RegulonDB database and, given this database, achieve recall of 33.1% and precision of 66.6%. Still, these systems were specifically tuned for the extraction of transcriptional regulation for the *E. coli* organism. In contrast to these studies based on manually defined rules and curated dictionaries tuned for *E. coli* gene names, Hahn et al. (2009) pursued the idea of reconstructing curated databases and compared rule-based and general ML-based system performance in extracting regulatory events. Given the same experimental settings, the ML-based system slightly outperforms the rule-based one, with the additional advantage that the ML approach is intrinsically more general and thus scalable. The extrinsic evaluation on RegulonDB presented in the next section is based on the previous study by Hahn et al. (2009) and its extension by Buyko and Hahn (2011).

6.2 Evaluation of JReX in Extraction of E. coli Regulatory Network

6.2.1 RegulonDB as a Gold Standard

RegulonDB is the largest electronically-encoded reference database containing manually curated knowledge of the transcriptional regulatory network of *E. coli K12* (Figure 6.1). This database is hosted by the Centro de Ciencias Genómicas (CCG) at the Universidad Nacional Autonóma de México.[3] RegulonDB is continuously updated with content from recent research papers on *E. coli.* The current version 7.2 of RegulonDB (May 2011), is used for the study presented in this section. In a range of information about *E. coli* gene expression, RegulonDB provides regulatory network interactions such as interactions between genes and transcription factors.[4] The following relevant information for each regulation event, among other things, is included in this database: regulatory gene (`Agent`), the regulated gene (`Patient`), the regulatory effect on the regulated gene (e.g., activation, suppression). A sample from the RegulonDB network has the form presented in Table 6.1.

Table 6.1: Entries in the regulatory network of the RegulonDB database, 7.2 (May 2011). The columns contain the following information: (1) transcription factor (TF) name, (2) gene names coding for this TF (regulatory gene), (3) Blattner number of the regulatory gene, (4) gene regulated by the TF (regulated gene), (5) Blattner number of the regulated gene, (6) regulatory effect of the TF on the regulated gene (+ activator, - repressor, +- dual, ? unknown).

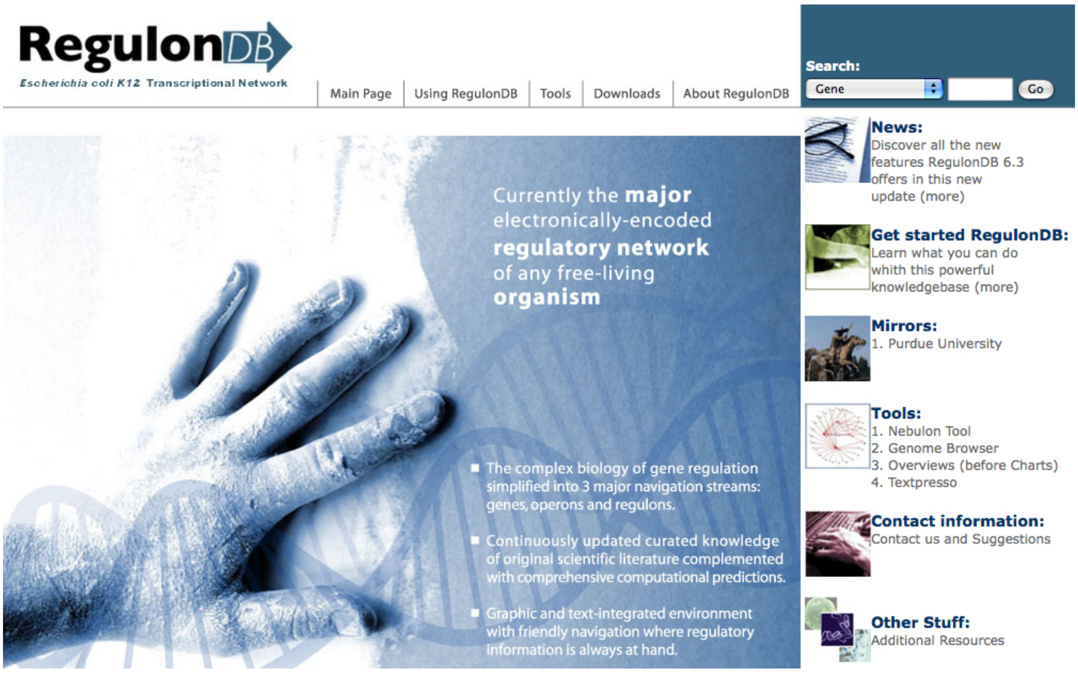
(1)	(2)	(3)	(4)	(5)	
AcrR	acrR	b0464	acrA	b0463	-
Ada	ada	b2213	aidB	b4187	+

RegulonDB in version 7.2 contains 4,005 regulatory relations for 1,621 genes. The reconstruction of this RegulonDB part requires detection of gene names in text, their mapping to Blattner identifiers[5] and, for the relational part, detection of regulatory interaction events between genes and classification of regulations in four regulatory types according to the regulatory effect of a transcription factor. In this

[3] `http://regulondb.ccg.unam.mx/`

[4] RegulonDB provides downloads for the curated data.

[5] Blattner numbers from the *E. coli* Genome Project are available at `http://www.genome.wisc.edu/sequencing/k12.htm`, last access January 2012.



Currently the **major** electronically-encoded **regulatory network** of any free-living **organism**

- The complex biology of gene regulation simplified into 3 major navigation streams: genes, operons and regulons.
- Continuously updated curated knowledge of original scientific literature complemented with comprehensive computational predictions.
- Graphic and text-integrated environment with friendly navigation where regulatory information is always at hand.

News: Discover all the new features RegulonDB 6.3 offers in this new update (more)

Get started RegulonDB: Learn what you can do whith this powerful knowledgebase (more)

Mirrors: 1. Purdue University

Tools:
1. Nebulon Tool
2. Genome Browser
3. Overviews (before Charts)
4. Textpresso

Contact information: Contact us and Suggestions

Other Stuff: Additional Resources

Figure 6.1: RegulonDB homepage, accessible at `http://regulondb.ccg.unam.mx/`, last access in January 2012.

work, I focused on detecting and mapping genes and detecting regulatory interaction events. The detection of a regulatory effect remains future work.

6.2.2 Configuration of the JReX System

The JReX system was slightly adapted for an automatic re-creation of *E. coli* regulatory network.

Entity Identification. To identify gene names in documents, JReX applied GeNo, a multi-organism gene name recognizer and normalizer (Wermter et al., 2009). I used GeNo in its original version, i.e., without special adjustments to the *E.coli* organism. However, only those mentions detected to be genes of *E. coli* were fed into the JReX argument extraction component.[6]

E. coli genes are often said to be regulated as part of operons, i.e., a cluster of genes under the control of a single transcription factor or promoter. The sample sentence "*AtoSC enhanced the transcription of the flhDC and fliAZY operons and to a lesser extent of the flgBCDEFGHIJKL operon.*" contains mentions of regulatory interaction concerning three operons, "*flhDC*", "*fliAZY*", and "*flgBCDEFGHIJKL*". These operons together refer to 16 single genes. Thus, this short sentence contains description of 16 regulatory events. Parsing operon names should considerably increase the recall of regulatory interactions between genes. Therefore, the named entity recognizer component was extended for the detection of operon names using a dictionary collected from RegulonDB data sets. Subsequently, operon names that had been recognized were parsed and each associated with its genes. The operon recognizer finds, for example, an operon name "*flhDC*" and divides it into individual gene names (e.g., "*flhD*", "*flhC*").

Regulation of Gene Expression (ROGE) Event Identification. For event identification, the JReX argument extraction component has to be adapted to this domain with the help of corpora providing annotations of regulation of gene expression (ROGE) events. The GeneReg corpus is the major corpus for regulatory interaction between genes (Section 3.6.4). Another relevant corpus is the "BioNLP 2009 Shared Task on Event Extraction" corpus ("Shared Task" corpus) (Section 3.6.2). It has been shown in this thesis that the regulation of gene expression can be expressed by means of entity-driven and trigger-driven annotations. GeneReg is a result of an entity-driven anotation approach, while the Shared Task corpus was developed with a trigger-driven annotation approach. I presented in Section 3.7 that both annotation approaches can be mapped. For example, a gene expression regulation event can be represented by means of, e.g., nested `Gene Expression` and `Regulation` events from the Genia ontology (cf. Section 3.7). Furthermore, `Binding` event is also relevant for gene expression regulation process (cf. Section 3.5.2 for micro views of events). For example, the sentence "*XapR* ***binds*** *to the of xanthosine phosphorylase (XapA) promoter.*" contains a `Binding` event between *XapR* and *XapA*. As *XapR* is a transcription factor, this `Binding` event describes an initial phase of a regulation of gene expression event and, thus, has to be interpreted as a gene expression regulation event.

The JReX argument extraction component was re-trained separately on both corpora. The following JReX models have been applied for the evaluation study:

[6] GeNo provides Uniprot *E.Coli* identifiers, which can be mapped to Blattner identifiers used in RegulonDB.

- *JReX-GeneReg*
- *JReX-Binding*
- *JReX-Regulation*

In the first variant (*JReX-GeneReg*), I used the GeneReg corpus for the training of JReX. In the second variant (*JReX-Binding*), JReX was re-trained on `Binding` event annotations from the Shared Task corpus. In the third variant (*JReX-Regulation*), JReX was trained on `Regulation` plus its sub-types events, and on `Gene Expression` and `Transcription` event annotations from the Shared Task corpus. Mappings between the entity-driven and trigger-driven representation were applied for extracting RegulonDB-conform annotations (cf. Table 6.1). As `Binding` events do not represent directed relations, I stipulate here that the protein occurring first is assigned an `Agent` role.[7] For argument detection JReX applied the graph kernel and MaxEnt models in an ensemble configuration (Section 4.7).

6.2.3 Evaluation Scenario and Experimental Settings

Evaluation against RegulonDB constitutes a real-life scenario for automatic curation of biomedical knowledge from the literature. The complete JReX extraction system was run, including text segmentation, parsing, gene name recognition and normalization, as well as event detection proper (cf. Section 4.7). Hence, the system's overall performance values might be highly affected by gene name identification and normalization. The results presented in this chapter should be considered as an evaluation of a complete JReX system as a pipeline. To evaluate the JReX system against RegulonDB, I processed various sets of input documents (see below), collected all unique gene regulation events extracted this way, and compared this set of events against the full set of known events in RegulonDB. A true positive (TP) hit is obtained when an event found corresponds to one in RegulonDB, i.e., having the same `Agent` and `Patient`. The type of regulation is not considered. A false positive (FP) hit is counted if an event is found which does not occur in the same way in RegulonDB, i.e., either `Agent` or `Patient` is wrong, or both are wrong. False negatives (FN) are those events covered by RegulonDB but not found by the system automatically. By default, all events extracted by the system are considered in the "transcription-factor-filtered" mode, i.e., only events with an agent from the list of all known transcription factors for *E. coli* are considered. From these hit values, standard precision, recall, and F-score values are calculated. I present the values without decimal place as it has no significance for this evaluation study.

[7] In particular transcription factors that bind to regulated genes are mentioned usually before the mention of regulated genes.

Of course, the system's performance largely depends on the size of the base corpus collection being processed. Various document sets were prepared for the evaluation scenario against RegulonDB and an overview of data sets is presented (Table 6.2). The RA set contains Medline abstracts referenced officially in the RegulonDB (version 7.2). The RF set includes full text journal articles collected by the RegulonDB team during curation and kindly provided for this study. The BA (abstracts) and BF (full texts) sets were collected in the BootSTrep project[8] which aimed to study the automatic extraction of ROGE events (Hahn et al., 2009).

Table 6.2: Document sets collected for the RegulonDB evaluation study (size of documents).

Document Set	Size
RA - RegulonDB abstracts	12,435
RF - RegulonDB full texts	2,528
BA - BootSTrep abstracts	4,344
BF - BootSTrep full texts	5,797

6.2.4 Intrinsic Evaluation on GeneReg

Before the validation of JReX models against RegulonDB was run, I performed validation of all JReX models on the GeneReg corpus that contains 1,164 gene expression regulation event annotations (cf. Appendix, Section A.1). The evaluation results are summarized in Table 6.3. The *JReX-GeneReg* variant achieved a 53.7% F-score on the GeneReg corpus in a ten-fold cross-validation . These results correspond to JReX performance on the Shared Task data (Section 5.2.4). Evaluation scenarios for the models applying Shared Task models plus mappings between Genia classes and ROGE events (see Section 3.7) are created by processing the whole GeneReg corpus. In another configuration, *viz. JReX-Regulation*, JReX achieved only 20.4% F-score on the GeneReg corpus. By combining the results of the *JReX-GeneReg* variant with the *JReX-Regulation*, I was able to increase the performance by under one percentage point F-score (see 54.5% F-score in Table 6.3).

As the GeneReg corpus contains few annotations which represent a binding process of the transcription factor to the regulated gene, the application of the *JReX-Binding* variant resulted in a performance close to zero. The low performance of the *JReX-Regulation* and *JReX-Binding* variants indicate that the Shared Task models

[8] `http:///www.bootstrep.org/`

Table 6.3: ROGE extraction results on the GeneReg corpus for all applied JReX variants/models.

JReX variant	GeneReg corpus		
	recall	prec.	F-score
JReX-GeneReg	49.1	59.3	53.7
JReX-Binding	...	...	...
JReX-Regulation	12.4	58.1	20.4
JReX-GeneReg & *JReX-Regulation*	52.0	57.2	54.5

might be not adequate for extracting ROGE events. This should be confirmed in an extrinsic evaluation scenario.

6.2.5 Extrinsic Evaluation against RegulonDB

For the validation against RegulonDB, the JReX system was applied using re-trained models of argument extractor components on all the document sets introduced above. As the baseline I decided on simple sentence-wise co-occurrence of tentative event arguments and event triggers, i.e., if two gene name mentions and at least one event trigger of a `Regulation` or its subtypes appear in the sentence, that pair of genes is considered to be part of a regulatory event. As the regulatory event assigns roles to its arguments, two regulatory events with interchanged `Agent` and `Patient` were built. The results of the baseline and JReX runs are presented in Table 6.4.

Using the baseline, the best recall was achieved on full texts (the RF set) with 63% recall, followed by the BF set with 45% recall. For the abstract sets, the baseline only achieved 35% recall on the RA set and 33% recall on the BA set. This outcome confirms the reasonable assumption that full text articles contain considerably more events than their associated abstracts. However, the precision of the baseline is miserable (2% on the RF, 9% on the BF, 19% on the RA and BA sets). The baseline achieved 67% recall, 2% precision and 4% F-score using all available data sets. This data indicates that a more sophisticated approach to event extraction, such as the one underlying the JReX system, is much needed.

The performance of JReX trained on the GeneReg corpus and on the “BioNLP 2009 Shared Task on Event Extraction” corpus was evaluated separately. The best

JReX-based results were achieved on full texts, on the RF set, with 24% recall, 50% precision and a 32% F-score (*JReX-GeneReg*, Table 6.4). The *JReX-Regulation* variant achieved only 2% recall, 31% precision and 3% F-score on the RF set. The *JReX-Binding* variant achieved 9% recall, 56% precision and 15% F-score on the RF set. While these results are compared with those of the combination with the *JReX-GeneReg* variant (Table 6.4), we see that the lack of an increase in F-score indicates that regulatory events are repeatedly described in texts using both `Binding` and regulatory event descriptions. The combination of the *JReX-GeneReg*, the *JReX-Regulation* and the *JReX-Binding* variants cannot achieve any performance increases in terms of F-score, only in terms of a recall (1 percentage point on the RF set) (see *JReX-GeneReg* & *JReX-BioNLP* in Table 6.4). Thus, the Shared Task data annotation is shown in this extrinsic evaluation study to be insufficient for high-performance extraction of regulatory networks as it is possible given the training on the GeneReg corpus especially created for learning the structure of gene expression regulation events. In general, JReX achieved a performance of 36% recall, 34% precision and 35% F-score using the *JReX-GeneReg* model only and all data sets. When these results are compared with the baseline (67% recall, 2% precision and 4% F-score, all data sets), we can envisage a considerable advantage for JReX-style analytics in the overall evaluation.

However, JReX fails to detect many regulatory events – nearly half of the possible events described in sentences could not be extracted (36% recall of the *JReX-GeneReg* compared to 67% recall of a co-occurrence approach on all data sets). As full text documents are generally more complex, the relative number of errors is higher here than on abstracts. When the JReX results on abstracts are compared against the baseline, I see that the missing rate for events is lower than on full texts. About 70% of all sentence-wise expressed regulatory events in the RA set can be successfully detected with acceptable precision of more than 50% (21% recall of *JReX-GeneReg* compared to 35% recall of a co-occurrence approach). The higher rate of errors on full texts can be explained as follows.

JReX is trained on the GeneReg corpus which contains Medline abstracts only. The intricacies of event descriptions in full texts could not be learned from this abstract-based corpus.

The overall evaluation study on the complete RegulonDB was complemented with a study on a regulatory network of a single transcription factor from this database. Many curators are working with known transcription factors and are searching for regulatory activities of a selected transcription factor. Therefore, JReX was evaluated in the detection of a regulatory network for the transcription factor *fur,* which is one of the principal transcription factors in *E. coli* (RegulonDB contains 284 regulatory relations with *fur* as an `Agent`). In Table 6.5 I show the results achieved

Table 6.4: ROGE event extraction results evaluated on RegulonDB for all known transcription factors in *E. coli*. Recall/Precision/F-score (R/P/F) values in % are given for each document set. *JReX-BioNLP* variant comprises *JReX-Binding* and *JReX-Regulation* variants.

JReX Variant	RA			RF			BA			BF			ALL		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Co-occurrence	**35**	**19**	**24**	**63**	**2**	**4**	**33**	**19**	**24**	**45**	**9**	**16**	**67**	**2**	**4**
JReX-GeneReg	**21**	**53**	**30**	**24**	**50**	**32**	**21**	**53**	**30**	**23**	**36**	**28**	**36**	**34**	**35**
JReX-Binding	6	57	10	**9**	**56**	**15**	6	57	11	7	38	12	15	43	22
JReX-Regulation	1	33	2	**2**	**31**	**3**	1	28	2	2	20	3	2	25	4
JReX-GeneReg & *JReX-Binding*	22	51	31	25	47	32	22	51	30	23	34	28	38	33	35
JReX-GeneReg & *JReX-BioNLP*	22	50	30	25	45	32	22	49	30	23	34	28	38	32	34

for abstract sets, full text sets and for all documents. The data reveals that JReX detected almost 50% of all sentence-wise expressed ROGE events in abstracts on the RA set (18% recall of *JReX-GeneReg* analytics compared to 41% recall of a co-occurrence approach). From full texts (RF set), about 40% of all sentence-wise expressed regulatory events could be extracted with a considerable gain in precision (34% recall of JReX-style analytics compared to 88% recall of a co-occurrence approach) while the overall F-score on full text documents (RF set) also increased to 47% F-score compared to the F-score of 28% on abstracts only (RA set) (cf. *JReX-GeneReg* in Table 6.5). A considerable increase in recall of 12 percentage points can be achieved using all document sets (46% recall for *JReX-GeneReg* on all data sets compared to 34% recall for *JReX-GeneReg* on the RF set only in Table 6.5). These results are similar to an increase in recall in the overall RegulonDB database evaluation (see Table 6.4) where *JReX-GeneReg* variant achieves 36% recall using all document sets and only 24% recall on the RF set.

In summary, the results of both studies revealed that the *JReX-GeneReg* variant performs better than the *JReX-Binding* or the *JReX-Regulation* models. Thus, the GeneReg corpus, which is annotated as a part of this thesis, was shown to be suitable for training a JReX model to extract regulatory events from the literature (in particular from abstracts). Another major outcome of both extrinsic evaluation studies is that many regulatory events which are described in abstracts cannot be detected in full text articles (although they should be described in a full text). That means that the error rate of event extraction performed by JReX (trained on abstracts only) is higher on full texts than on abstract documents. Therefore, for the future work event annotated corpora should be complemented by full text articles in order to train event extraction engines which can support database curation processes effectively.

6.2.6 Manual analysis of False Positives

RegulonDB was taken as an undisputed gold standard in this evaluation. If a system correctly extracts an event which is not contained in RegulonDB for some reason, this constitutes a false positive (FP). Moreover, all kinds of errors (e.g., `Agent` and `Patient` are mixed up) were considered as FP errors. To analyze the causes and distribution of FPs in more detail, a manual analysis of the FP errors was performed by a student of biology and original FP hits were assigned to one of five FP error categories. The analysis presented in this section summarizes reports about these FP categories reported by Hahn et al. (2009) (original publication) and (Buyko and Hahn, 2011). The five FP categories are:

Table 6.5: Event extraction results evaluated on RegulonDB for the transcription factor *fur*. Recall/Precision/F-score (R/P/F) values in % are given for each document set. *JReX-BioNLP* variant comprises *JReX-Binding* and *JReX-Regulation* variants.

JReX Variant	RA			RF			BA			BF			ALL		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Co-occurrence	**41**	**29**	**34**	**88**	**9**	**16**	**12**	**15**	**13**	**72**	**22**	**34**	**91**	**8**	**15**
JReX-GeneReg	**18**	**65**	**28**	**34**	**74**	**47**	**6**	**83**	**11**	**32**	**61**	**42**	**46**	**57**	**51**
JReX-Binding	1	100	2	16	100	28	2	50	4	4	43	7	20	74	31
JReX-Regulation	1	50	2	4	43	7	1	50	2	1	50	2	5	44	9
JReX-GeneReg & *JReX-Binding*	18	65	28	34	74	47	6	62	11	32	56	41	46	55	50
JReX-GeneReg & *JReX-BioNLP*	18	62	28	34	71	46	46	53	49	32	56	41	46	53	49

Table 6.6: Manual analysis of false positive errors (FP). Percentages of FPs by category are reported on the RA set.

Category	Numbers	(%)
Cat 1	79	(19.5%)
Cat 2a	158	(39.0%)
Cat 2b	108	(26.6%)
Cat 3	5	(1.2%)
Cat 4	14	(3.4%)
Cat 5	42	(10.3%)
Total	405	

Not an Interaction (Cat1): This is really an FP error, since the extracted event in no way constitutes an interaction event.

Interaction (Cat2a): This is not a gene expression regulatory interaction (but may constitute another interaction event).

ROGE but other than transcription (Cat2b): Unlike RegulonDB, which contains only one subtype of ROGEs, namely transcriptional ROGE, JReX identifies all kinds of ROGEs. Hence, JReX is able to identify events which are intentionally excluded from RegulonDB by design and, therefore, are not really FPs.

Partially correct ROGE event (Cat3): This category deals with incorrect arguments of ROGEs where the `Patient` and the `Agent` role are interchanged. Although this is erroneous, human curators might find the only partially incorrect information useful to speed up the curation process.

Correct ROGE event not mapped to RegulonDB (Cat4): Identified gene names were incorrectly normalized so that they could not be found in RegulonDB.

ROGE Event missing in RegulonDB (Cat5): These are events which should be contained in RegulonDB but are missing for some reason. The agent is a correctly identified transcription factor and the sentence contains a mention of a transcription event. There are several reasons why this relation was not found in RegulonDB, as I will discuss below.

With these categories in mind, one student of biology analysed 405 false positives extracted from the RA set. Table 6.6 shows the results of this manual analysis.

The largest source of errors resides in `Cat2a`, i.e., an identified event is a general interaction, though not a regulatory one. However, more than 26% of the FPs are due to the fact that the system found regulatory events which were too general and which, by definition, are not contained in RegulonDB (`Cat2b`). Identified ROGEs that were partially correct constitute 1.2% of the FP errors (`Cat3`).

Furthermore, 3.4% of the FPs are correctly identified transcription events that could not be mapped to the RegulonDB (`Cat4`) due to incorrect gene normalization or gene names that were too general. Finally, 10.3% of the FPs are correct transcription events which are missing in RegulonDB (`Cat5`). There may be several reasons for this. For instance, RegulonDB curators have not yet added an event or simply overlooked it, or events are correctly identified as such in the narrow context of a paragraph of a document but were actually only of a speculative nature (this includes events whose status is unsure, often indicated by words "*likely*" or "*possibly*"). These false positives from `Cat5` now have to undergo manual analysis by the RegulonDB curator team (future work).

In summary, the manual FP analysis shows that about 80% of all FPs are not completely erroneous. From this "correct" false positive set, 10.3% (`Cat5`) are even interesting and relevant for the RegulonDB. Only 19.5% of events are definitely faulty results (`Cat1`). These numbers must clearly be kept in mind when interpreting the raw figures (especially for precision) reported in this study.

The JReX extraction results have been given to the RegulonDB team. Furthermore, these results might be integrated in the Fact Database (FactDB) hosted by the European Bioinformatics Institute (EBI),[9] one outcome of the BootSTrep project[10] which integrated manually curated RegulonDB data and automatically extracted data using IE tools. Julie Lab contributed to the FactDB using the initial argument extraction engine from 2009 (Hahn et al., 2009).

6.3 Construction of a Candida albicans Regulatory Network with JReX

The second extrinsic evaluation study of JReX focused on constructing a regulatory network for the human pathogenic microorganism *Candida albicans* (*C. albicans*), which normally lives as a harmless commensal yeast within the body of healthy humans (Odds, 1988). However, this fungus can change its benevolent behavior and cause opportunistic superficial infections of the oral or genital epithelia. Given this

[9] `http://wwwdev.ebi.ac.uk/tc-test/textmining/FactDBInterface/`

[10] `http://www.bootstrep.org`

pathogenic behavior, knowing the underlying regulatory interactions might help to understand the onset and progression of infections. Despite their importance, the number of known regulatory interactions in *C. albicans* is still rather small. The (manually built) Transfac database[11] collects regulatory interactions and transcription factor binding sites for a number of organisms. However, it includes information about only five transcription factors of this fungus. The Candida Genome Database[12] includes the most up-to-date manually curated gene annotations. Although regulatory interactions might be mentioned, they are not the main focus of this database and often rely solely on micro-array experiments.

Currently, the Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute (HKI) in Jena is manually collecting transcription factor-target gene interactions for the *C. albicans* fungus Regulatory Network (RN). The workflow requires careful reading of research papers and critical interpretation of experimental techniques and results reported therein. However, the most time-consuming step is reading papers and understanding the interactions of a transcription factor or its target genes. In fact, it took this group more than two years to identify information about the 79 transcription factors (TFs) for the *C. albicans* fungus. To speed up this process, JReX was called in to identify regulatory interactions more rapidly and at a higher rate.

Evaluation against *C. albicans* Regulatory Network (RN) constitutes a challenging real-life scenario. Two gold standard sets were used for this evaluation study, *viz. C. albicans* RN and *C. albicans* RN-small. The *C. albicans* RN refers to the collection of regulatory interactions curated by the HKI in Jena. *C. albicans* RN includes the following information for each regulation event: regulatory gene (`Agent` in such an event, a transcription factor), and the regulated gene (`Patient`). *C. albicans* RN contains 114 interactions for 31 transcription factors. As this set does not contain any references from interactions to the full texts they were extracted from, there is no guarantee that the reference full texts are indeed contained in the document sets used for this evaluation study (see above). Therefore, the *C. albicans* RN-small set was built, which is a subset of the *C. albicans* RN, complemented by references to full texts (eight documents) and containing 40 interactions for seven transcription factors. *C. albicans* RN-small serves here as an additional small gold standard set for a more proper evaluation scenario of JReX.

[11] `http://www.biobase-international.com/index.php?id=transfac/`

[12] `http://www.candidagenome.org/`

6.3.1 Configuration of the JReX System

The JReX system was slightly adapted for an automatic construction of *C. albicans* regulatory network. In particular, the named entity recognition part of the system was enhanced with new dictionaries for *C. albicans* gene and transcription factor names.

Entity Identification. To identify *C. albicans* gene names in the documents, JReX applied two named entity detection approaches, i.e., GeNo (cf. Section 6.2.5) and a dictionary-based approach using a dictionary with *C. albicans* gene names collected from the Candida Genome Database (see above).

ROGE Event Identification. For event identification, the JReX argument extraction component has to be adapted to this domain with the help of corpora providing annotations of ROGE events as presented in the evaluation study against the RegulonDB. As the RegulonDB study reveals that only *JReX-GeneReg* is adequate for extracting regulatory networks (Section 6.2.5), I applied only the *JReX-GeneReg* model in the current evaluation study. The *JReX-Regulation* and *JReX-Binding* variants were dropped as both were shown to under-perform in extracting regulatory networks (cf. RegulonDB study).

6.3.2 Evaluation Scenario and Experimental Settings

To evaluate JReX against *C. albicans* RN, various sets of input documents (see below) were processed and all unique gene regulation events extracted were collected and compared with the full set of known events in *C. albicans* RN. The evaluation settings correspond to those in the RegulonDB evaluation study (Section 6.2.3). I present the achieved performance values without decimal place as it has no significance for this evaluation study.

Various *C. albicans* research paper sets were prepared for the evaluation against the *C. albicans* gold standards. I used a PubMed search for "*Candida albicans*" and downloaded 17,750 Medline abstracts (retrieved from PubMed in July 2011). The document set of 6,000 freely available papers, in addition to approximately 1,000 non-free full text articles were kindly provided by the HKI in Jena. An overview of the data sets involved is given in Table 6.7. The CA document set is composed of 17,746 Medline abstracts. The CF-small document set is composed of eight full texts from the CF set (7,024 full texts) that are explicitly used as references for *C. albicans* interactions from *C. albicans* RN-small regulatory network (see above).

Table 6.7: Document sets collected for the *Candida albicans* evaluation study.

Document Set	Number of Documents
CA - *C. albicans* abstracts	17,746
CF - *C. albicans* full texts	7,024
CF-small - *C. albicans* RN-small texts	8

Table 6.8: Event extraction evaluated on full data set *C. albicans* RN for known transcription factors in *C. albicans* . Recall/Precision/F-score (R/P/F) values in % are given for each document set.

JReX Variant	CF			CA			ALL		
	R	P	F	R	P	F	R	P	F
Co-occurrence	**84**	**1**	**3**	27	9	14	**84**	**1**	**3**
JReX-GeneReg	**35**	**18**	**24**	13	29	18	**35**	**18**	**23**

6.3.3 Experimental Results

JReX was applied on all document sets. As a baseline I decided on simple sentence-wise co-occurrence of putative event arguments, as was done for the RegulonDB study. The results of the baseline and JReX runs are presented in Table 6.8 for the *C. albicans* RN data set and Table 6.9 for the *C. albicans* RN-small data set.

Using the baseline, the best recall was achieved analyzing full texts (CF document set), with 84% recall on the *C. albicans* RN (Table 6.8) and 90% recall on the *C.*

Table 6.9: Event extraction evaluated on referenced data set *C. albicans* RN-small for known transcription factors in *C. albicans* . Recall/Precision/F-score (R/P/F) values are given in % for each document set.

JReX Variant	CF			CA			CF-small			ALL		
	R	P	F	R	P	F	R	P	F	R	P	F
Co-occurrence	**90**	**2**	**3**	**31**	**13**	**18**	**72**	**14**	**24**	90	2	3
JReX-GeneReg	**64**	**30**	**41**	13	28	18	**54**	**51**	**52**	**64**	**29**	**40**

albicans RN-small set (Table 6.9). Analyzing the abstract documents (CA document set), the baseline achieved only low recall values of 27% recall on the *C. albicans* RN data set and 31% recall on the *C. albicans* RN-small data set. As these recall figures of the baseline reveal that abstracts contain far fewer ROGE events than full texts, I have further strong evidence that the biomedical NLP community needs to process full-text articles to support database curation effectively (cf. outcomes in the RegulonDB evaluation). The precision of the baseline using full text articles is miserable as in the RegulonDB study as well (1% (CF set) for the *C. albicans* RN data and 2% (CF set) on the *C. albicans* RN-small data). Therefore, the JReX system was applied in the next evaluation step.

On the *C. albicans* RN data, JReX achieved 35% recall, 18% precision and 24% F-score using full texts (see CF set in Table 6.8). Given the lack of literature references from the *C. albicans* RN set, I cannot guarantee that the CF document collection contains all relevant documents. Therefore, the *C. albicans* RN-small data was used for a proper evaluation of JReX. The best JReX-based results were achieved analyzing eight officially referenced full text articles, *viz.* CF-small document set. JReX achieved a 52% F-score here, with 54% recall and 51% precision (see CF-small document set in Table 6.9). This level of performance compares fairly well with the results JReX obtained in the "BioNLP 2009 Shared Task on Event Extraction" Task 1 (Section 5.2). When I used all the data sets for the evaluation on the *C. albicans* RN-small data, the recall peaked at 64% with a reasonable precision of 29% and an F-score of 40% (Table 6.9). Thus, the precision and the F-score values drop if all documents are considered in this study. This can be explained as follows. The *C. albicans* RN-small data contains few regulatory events so the evaluation on the full document collection provides a large set of wrong "false positives".

In addition to the evaluation experiments presented in this section, the data gathered by JReX on the full text set was provided to the HKI team in Jena. This data collection contains 503 genes and 1,016 automatically detected interactions between them. The HKI team used this regulatory network to evaluate the networks inferred using other bioinformatics methods which find regulatory events on DNA sequencing and gene expression data. Linde et al. (2011) applied this *C. albicans* data generated by JReX as a gold standard for predicting regulatory networks based on expression data and achieved very promising results. Thus, system biology applications are supported by this data set.

6.4 Discussion of Extrinsic Evaluation Studies

Extrinsic evaluation studies presented in this section dealt with the issue of automatically re-creating real-life biological fact databases. Both evaluations, e.g., against RegulonDB and the *C. albicans* regulatory network, constituted a challenging real-life scenario where the databases served as the undisputed basic truth. The evaluation studies built on various sets of documents used in part by human expert curators of this database. Thus, the JReX application tried to replicate the work of human curators automatically. Indeed, it was possible to construct considerable portions of regulatory network databases for *C. albicans* and *E. coli* model organisms automatically by processing the relevant literature sources with the JReX system. For the RegulonDB database the JReX system was able to extract more than a third of known regulatory events with a precision of 34%. The HKI *C. albicans* database was reconstructed with 35% recall and 18% precision. It is difficult to perform an extrinsic evaluation as experimental comparison with manual curation work that requires knowledge of curation guidelines and should involve the availability of complete document data sets applied in the curation process. In both studies I had no guarantee of the completeness of document sets (there may be entries in both databases that were taken from documents not available for the JReX system). The curation guidelines (which might be fuzzy) were not available for both studies. Given the evaluation results and restrictions mentioned above (incomplete document sets and lack of guidelines), I may conclude that this evaluation data constitutes a more or less reasonable lower bound on recognition recall and precision of the JReX system in a real-life evaluation scenario.

The extrinsic evaluation presented in this chapter investigates the performance of three JReX models trained on Shared Task corpus and on the GeneReg corpus for ROGE event extraction. My conclusions are on two levels. First, the JReX argument extraction component, which is one of the best-performing event extraction systems in the Shared Task if trained on the Shared Task event corpus, cannot outperform the same system if it is trained on the GeneReg corpus. JReX re-trained on GeneReg performs much better than the JReX models trained on the Shared Task data. Thus, the GeneReg corpus, which is developed in a entity-driven annotation approach (cf. Section 3.6.4), is more suitable for the extraction of regulatory networks in real-life applications.

Second, both evaluation experiments reveal that full texts have to be screened in addition to informationally poorer abstracts. As full-text documents are linguistically more complex and thus harder to process, the relative number of errors is higher than on abstracts. The manual analysis of false negatives revealed that we miss, in particular, events that are described in a cross-sentence manner using coreferential expressions. As JReX extracts events only within the same sentence, the next step should be to incorporate cross-sentence mentions of entities as well. Furthermore, the GeneReg corpus data will be complemented with annotated full text documents. This is the future work.

The detailed analysis of false positives from the RegulonDB study revealed that the strict evaluation criteria applied can be considered in another light if human curators evaluate the data. JReX confused agents and patients, for example or detected information not contained in RegulonDB which might be useful for curation. This manual analysis study of the "false positives" generated illustrated that the JReX system recognizes relevant facts which have not yet been included in the reference database (based on the same document collection). This finding might indicate that text mining methodology is capable not only of replicating, to some degree, human performance in this knowledge-rich domain, but also of unveiling knowledge contained in these documents disregarded by human experts. According to this set-up, 10.3% of the correct biological knowledge was identified not only as being correct but also as being "relevant" for the RegulonDB. However, curators still have to screen this data for "interesting" facts as it is a considerable part of the curator's work.

Given the results of extrinsic evaluation studies, I can conclude that the use of automatic event extraction systems such as JReX as a pre-processor complements human curation efforts. The data extracted automatically serves as additional input for the pathway and regulatory network integration procedure in order to visualize and explore complex biological interaction processes. In summary, the JReX-based curation approach can harvest large numbers of molecular event instances from the literature, thus complementing efforts by curators and bioinformaticians and in general helping to widen the knowledge acquisition bottleneck in the field of molecular biology.

Chapter 7

Conclusion

The emphasis of this thesis is on information extraction solutions for the biosciences, focusing more precisely on the topic of automatic event extraction. I tackled this subject in various ways, 1) through analysis of molecular event descriptions in literature, with a subsequent discussion of their modeling for information extraction analytics; 2) by designing, implementing and extensively evaluating a state-of-the-art event extraction system which relies on dependency graphs as a major knowledge source for semantic applications; and 3) by demonstrating how far such an event extraction engine is able to go in solving real-life problems for large-scale knowledge curation. The three research questions posed at the beginning of this thesis (cf. Chapter 1), are comprehensively answered in corresponding sections.

Research on the first question, relating to particularities of molecular events, revealed in detail the complexity of event descriptions, which can be captured under the *eventuality* concept. In consequence I subsumed under the term *(molecular) event* the events proper, static views of events (in the sense of Croft (1990)) and facts about events. Based on these theoretical outcomes, an ontology-based large-scale annotation approach was selected as a suitable solution for event modeling and annotation of event instances in scientific literature. The second question, on the design of an automatic event extraction approach and knowledge sources, constitutes the major part of this thesis and results in the implementation of a high-performance event extraction engine, the JReX (Jena Relation eXtraction) system. This system was applied by the Julie Lab team in the "BioNLP 2009 Shared Task on Event Extraction" competition and ranked second among 24 competing teams. After the competition, the usefulness of various knowledge sources for solving event extraction task was investigated. This revealed that syntactic dependency graphs or, more precisely, shortest dependency paths, constitute crucial knowledge for high-performance event extraction. Furthermore, this knowledge source was selected in its most suitable configuration (CoNLL representation) and additionally elaborated by syntactic and semantic trimming procedures developed in this thesis. These experiments on dependency graphs helped to increase the overall JReX performance in terms of F-score. The updated JReX system currently achieves 57.6% precision, 45.7% recall and a 51.0% F-score (Buyko et al., 2011a), and is the top scorer on the U-Compare event extraction server, now outperforming the competing systems from the "BioNLP 2009 Shared Task on Event Extraction" (Kano et al., 2011). The similar F-score performance of 54.5% is achieved on another independently created event-annotated corpus, the GeneReg corpus, created as part of this thesis. These evaluations reveal that the F-score of slightly more than 50.0% is a state-of-the-art performance of an event extraction approach tackling molecular events. The question thus arose as to whether the selected event extraction approach is working to its limits. My investigation into the learning progress of the JReX argument extraction component showed that JReX has the potential to perform better if the machinery is provided with more annotated data on which to learn. The JReX approach crucially relies upon training data. The learning curves created on "BioNLP 2009 Shared Task on Event Extraction" data revealed that the JReX requirement for new annotated data increases exponentially.

Thus, the bottleneck of the selected event extraction approach is in the lack of annotated data. Manual corpus annotation is known to be a costly and time-consuming process. Therefore, alternative annotation scenarios should be considered as potential solutions to this problem. For example, an interactive annotation process based on *Active Learning*, can support more rapid creation of annotated corpora in the biomedical domain. Active Learning can accelerate corpus creation because it only selects examples that are useful for learning. Active Learning was demonstrated to be suitable for speeding up the creation of semantically (named entity and relationship) annotated corpora in different language domains, including the biomedical field (e.g., Tomanek (2010), Vlachos (2009)). Another model creation scenario is based on the idea of training models on automatically annotated corpora, in a *distant supervision* mode (e.g., Mintz et al. (2009)). The distant supervision method scans available (large-scale) knowledge databases for manually curated information, searches for this information in literature, and collects together in a corpus textual evidence of this knowledge that has been detected. This fully automatic approach to corpus creation contains a risk of creating noisy training data. However, various studies have shown that created corpora are very close to gold data standards. For example, Buyko et al. (2012) automatically gathered a large corpus on genotype-phenotype relationships from the knowledge in the PharmGKB database,[1] the largest database of genetic variants and their phenotypic manifestations (Klein et al., 2001). The manual analysis of this corpus, which contains 1,980 Medline abstracts with about 20,000 annotated relationships, revealed that it was created with nearly 90% accuracy. JReX, re-trained on this automatically compiled data, achieved F-scores of the order of 80% in an intrinsic evaluation scenario and 73% in an extrinsic one (on

[1] http://www.pharmgkb.org/

PharmGKB) for the extraction of genotype-phenotype association relationships. While it is possible to criticise intrinsic evaluation of the automatically generated gold standard substitute, the extrinsic evaluation scenario clearly demonstrates the ability of JReX to learn effectively from this corpus and to find previously invisible PharmGKB data (which was not present in the training set, cf. for details Buyko et al. (2012)).

In general, an extrinsic evaluation scenario can provide evidence whether an event extraction engine that has successfully been intrinsically evaluated can solve real-life problems, and so demonstrate the robustness of the machinery. In this thesis, the extrinsic evaluation study takes on the challenging task of automatically reproducing the content of large databases manually curated from the scientific literature. I selected two sources of gold standard data for evaluating the ability of the JReX system to reconstruct manually curated knowledge. These were RegulonDB, the world's largest database for the transcriptional regulation network of *E. coli*, and *Candida albicans* RN, the database on the regulatory network of the candida fungus created at the Leibniz Institute for Natural Product Research and Infection Biology - Hans Knöll Institute in Jena. JReX was indeed able to re-create considerable portions of both regulatory network databases. More than a third of the known regulatory information of both databases, with 34% precision for RegulonDB and 18% precision for *Candida albicans* RN, was found in relevant scientific literature. Given the restrictions faced by evaluation studies that have been carried out, such as incomplete document sets, lack of curation guidelines or even incompleteness of curated knowledge (*Candida albicans* RN is under development)[2], these results constitute a reasonable lower bound on recognition recall and precision of the JReX system in a real-life evaluation scenario. Both studies demonstrated that the JReX system can partly re-create manually curated databases and can even complement human curation efforts.

Although the work of human curators cannot currently be fully replicated by an automatic solution, JReX can support real-life efforts of manual curation and can thus prevent the loss of usefulness of curated data where the gaps between published data and curated data are increasing dramatically (e.g., active research on regulatory networks and pathways). The automatic generation and early update of regulatory networks and biological pathway diagrams from scientific literature with the help of information extraction analytics is a first step in biomedical text mining towards automatic knowledge management. In the next step, text mining envisages creating more complete and even new knowledge that could not have been generated due to the lack of human resources or that is not even reported in the scientific literature.

[2] The incompleteness of the *Candida albicans* RN might explain low precision rates achieved in the evaluation study on the literature collection considered.

In this thesis I focused on the first issue, *viz.* automatic large-scale harvesting of relevant knowledge from scientific literature, while the next step will be the subject of future work.

Appendix A

Appendix

A.1 GeneReg Corpus, Description of the Annotation Process

GeneReg (Gene Regulation Corpus) is the result of an annotation campaign led by the Julie Lab (Buyko et al., 2008). It contains annotations of gene expression regulation (ROGE) events. The structure of GeneReg and some of its quantitative characteristics will be described in this section. GeneReg provides three levels of semantic annotations:

- *named entities* involved in gene regulatory processes, such as TFs (transcription factors, cofactors and regulators) and genes,
- *events in the form of pairwise relations* between TFs and genes,
- *event triggers* (e.g., clue verbs) essential for the description of gene regulation events.

For all three annotation levels, the annotation vocabulary was taken from Gene Regulation Ontology GRO (see GRO regulatory branch in Figure A.1). GRO describes gene regulation processes occurring at the intra-cellular level (such as the binding of transcription factors to DNA binding sites) and the physical entities that are involved in these processes (such as genes and transcription factors).

GeneReg annotation guidelines treat the four major annotation categories (Section 3.6.1) as follows:

- Event extent is set to a single sentence.
- Sentence must contain lexical anchors indicating gene expression regulation events.
- Syntactic structures are unrestricted; anaphoric mentions are allowed.

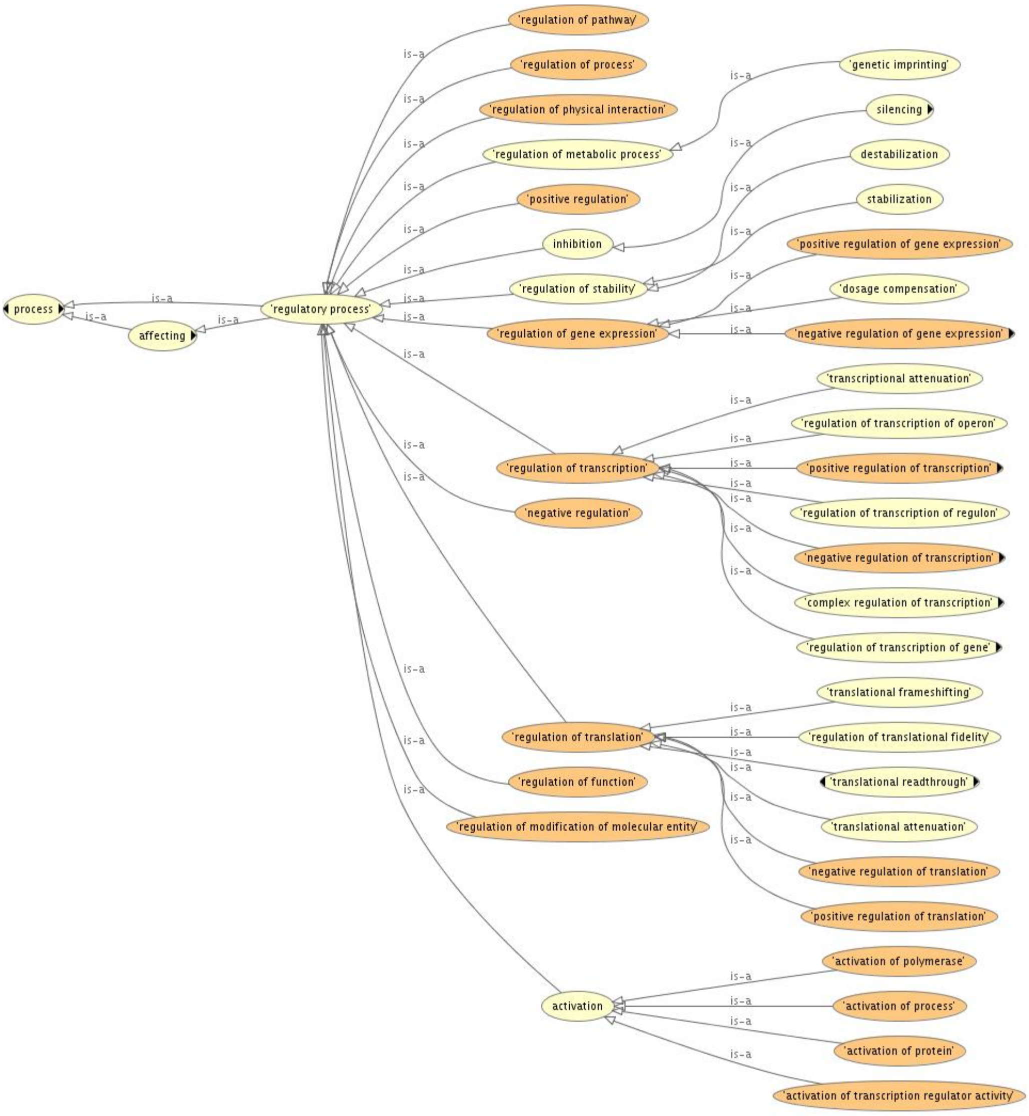

process
affecting
'regulatory process'
is-a
'regulation of pathway'
'regulation of process'
'regulation of physical interaction'
'regulation of metabolic process'
'positive regulation'
inhibition
'regulation of stability'
'regulation of gene expression'
'regulation of transcription'
'negative regulation'
'regulation of translation'
'regulation of function'
'regulation of modification of molecular entity'
activation
'genetic imprinting'
silencing
destabilization
stabilization
'positive regulation of gene expression'
'dosage compensation'
'negative regulation of gene expression'
'transcriptional attenuation'
'regulation of transcription of operon'
'positive regulation of transcription'
'regulation of transcription of regulon'
'negative regulation of transcription'
'complex regulation of transcription'
'regulation of transcription of gene'
'translational frameshifting'
'regulation of translational fidelity'
'translational readthrough'
'translational attenuation'
'negative regulation of translation'
'positive regulation of translation'
'activation of polymerase'
'activation of process'
'activation of protein'
'activation of transcription regulator activity'

Figure A.1: Gene Regulation Ontology, regulatory branch.

- Inferring annotations is allowed provided that lexical anchors are mentioned inside an event extent.

In the following I present the annotation steps in creating GeneReg. The annotation of named entities (Section A.1.1), annotation of event triggers (Section A.1.2), and finally `Regulation of Gene Expression` (ROGE) eventive relations (Section A.1.3).

A.1.1 Named Entities

In the first step, named entities, which play a central role in formal gene expression regulation events, were annotated by a graduate student of biology taking into account the semantic categories presented in Table A.1 which also gives the GRO definitions. This annotation step was supervised by Elena Beisswanger (Julie Lab). The annotated named entity types cover `Transcription Regulator` with its subtypes `Transcription Factor` and `Transcription Cofactor`, `Gene` and `Gene Group`, `Polymerase`, and `Ligand`. The annotation was carried out following a set of guidelines that were established especially for this annotation task in the Julie Lab. The number of annotations per semantic category is presented in Table A.2.

To assess the Inter-Annotator Agreement (IAA) for the entity annotation, a second graduate student of biology annotated a subset of 248 abstracts. For this subset the IAA was computed by applying three standard IAA measures for the named entity task: Strict IAA (69% (R), 62% (P), 65% (F))[1], Correct-Span IAA (74% (R), 76% (P), 72% (F)) and Correct-Category IAA (79% (R), 81% (P), 80% (F)). The correct-category IAA results were encouraging. They show that the annotation span is one of the issues in annotating named entities. Whether or not modifiers are integrated plays an important role here.

An annotator also annotated anaphoric mentions of `Gene` and `Gene Group` entities (see also Table A.2). This annotation task was implemented to provide more instances of entity mentions for the annotation of gene regulation relations. Anaphoric mentions were only annotated in sentences containing gene regulation relations. The numbers of anaphoric annotations are presented in Table A.2.[2]

[1] R/P/F stand for Recall/Precision/F-score values.

[2] We did not assess the IAA for this annotation subtask.

Table A.1: GRO definitions of GeneReg named entity annotations per semantic category.

Named Entity Category	Definition
`Transcription Regulator`	Protein that has transcription regulator activity.
`Transcription Factor`	A transcription factor that binds to a specific DNA sequence in order to modulate transcription. The transcription factor may or may not also interact selectively with a protein (other transcription factors of cofactors) or protein or macromolecular complex.
`Transcription Cofactor`	A transcription factor that binds to other transcription factors / the core RNA polymerase II complex but does not bind DNA itself.
`Gene`	A unit of inheritance; a working subunit of DNA that contributes to phenotype/function and carries a particular set of instructions, usually coding for a particular protein.
`Gene Group` (GRO `Operon`)	A genetic regulatory system in which genes (structural genes) coding for functionally related proteins are clustered along the DNA. The expression of structural genes is controlled by the regulatory elements (operator genes) that respond to environmental cues.
`Polymerase`	Enzymes that catalyze the synthesis of nucleic acids on preexisting nucleic acid templates, assembling RNA from ribonucleotides or DNA from deoxyribonucleotides.
`Ligand`	Binding to a specific receptor protein and activating intracellular signaling cascades that alter the behavior of the cell.

A.1.2 Trigger Words

In a second step, a graduate student of biology annotated trigger words referring to mentions of events relevant to gene expression regulation. To prepare for this trigger word annotation, one graduate student of biology and I manually screened the abstracts in the corpus and compiled a list of frequently occurring ways in

Table A.2: Number of GeneReg entity annotations per semantic category.

Named Entity Category	Annotations
`Transcription Factor`	2496
`Transcription Cofactor`	14
`Transcription Regulator`	40
`Gene`	2547
`Gene Group/Operon`$_{\mathrm{GRO}}$	1180
`Gene` (anaphoric)	24
`Gene Group/Operon` (anaphoric)	71
`Polymerase`	348
`Ligand`	633

which molecular processes were verbalized in the description of gene regulation relations. During this extensive manual analysis of biomedical texts, trigger words classified as relevant for the process of gene expression regulation were grouped into nine categories based on GRO concepts. These categories were `Gene Expression`, `Transcription`, `Regulation`, `Positive Regulation`, `Negative Regulation`, and `Experimental Intervention` with subtypes `Genetic Modification`, `Artificial Increase`, and `Artificial Decrease` (Table A.3).[3] The annotator was thus provided with an initial list of event triggers which they could extend during annotation. Trigger words indicating textual mentions of the listed atomic events were annotated with the nine semantic categories presented in Table A.3. Trigger words in GeneReg are basically main verbs, verb nominalizations and adjectives.

For example, the sentence

(A.1) *"H-NS and StpA proteins* ***stimulate expression*** *of the maltose regulon in Escherichia coli."*

contains two trigger words: first, "*stimulate*" is a trigger for a process in the category `Positive Regulation` and is a main verb, second, "*expression*" is a trigger for a process belonging to the category `Gene Expression` and is a verb nominalization.

The annotation numbers (Table A.4) reveal that most frequent are `Regulation` event triggers followed by `Positive Regulation`, and `Genetic Modification` trig-

[3] `Experimental Intervention` events extend the original GeneReg version presented by Buyko et al. (2008).

Table A.3: GRO definition of GeneReg trigger word annotations per semantic category.

Semantic Category	Annotations
`Gene Expression`	The process by which the information encoded in a gene is converted into protein or some form of RNA. The DNA sequence is first transcribed into RNA and then usually translated into protein.
`Transcription`	The synthesis of either RNA on a template of DNA or DNA on a template of RNA.
`Regulation` / `Regulatory Process`$_{\text{GRO}}$	Any process that modulates the frequency, rate or extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule.
`Positive Regulation` / `Activation`$_{\text{GRO}}$	Any process that activates, maintains or increases the frequency, rate or extent of an action.
`Negative Regulation` / `Inhibition`$_{\text{GRO}}$	Any process that stops, prevents or reduces the frequency, rate or extent of an action.
`Genetic Modification`	Genetic modification of DNA.
`Artificial Increase`/ `Increase`$_{\text{GRO}}$	A process of becoming larger, more numerous, more important, or more likely.
`Artificial Decrease`/ `Decrease`$_{\text{GRO}}$	A process of becoming smaller, less numerous, less important, or less likely.

gers. The most infrequent atomic events in the GeneReg corpus are mentions of `Artificial Increase`, `Artificial Decrease` and `Transcription`.

Most frequent among the variety of event triggers is `Positive Regulation` (110), `Negative Regulation` (93) and `Regulation` (82) event triggers followed by the description of `Genetic Modification` (56). The triggers to express `Gene expression` are limited to 15 variants and those for `Transcription` to only 12. The descriptions of `Artificial Decrease` and `Artificial Increase` have the lowest frequencies in

Table A.4: Number of GeneReg trigger word annotations in each semantic category (in brackets, unique triggers (not morphologically normalized)).

Semantic Category	Annotations
`Gene Expression`	495 (15)
`Transcription`	46 (12)
`Regulation` (unspecified)	896 (82)
`Positive Regulation`	835 (110)
`Negative Regulation`	441 (93)
`Genetic Modification`	558 (56)
`Artificial Increase`	52 (8)
`Artificial Decrease`	47 (5)

the whole corpus.

These statistics on event triggers clearly show that the GeneReg texts contain various descriptions of positive and unspecified regulation of gene expression given genetic modification as an experimental condition. This helps illustrate regulation and the influence of genes knocked out in mutant cells. Here is a characteristic GeneReg example:

(A.2) "*Nitrate-response by narG-lacZ and narK-lacZ was* ***reduced*** *by about 50 in a modE* ***mutant****.*"

This example describes positive regulation of the *narG* and *narK* genes by the *modE* protein. The text contains a mention of a reduced response of these genes to nitrate, and this event happens in a mutated cell in which the *modE* protein is absent after the modification of the DNA (*modE mutant*). This means that the *modE* normally has a positive regulatory effect on *narG* and *narK* expression. This can only be shown, however, by knocking out expression of the *modE* transcription factor. To determine the IAA a set of 65 randomly selected abstracts was annotated by the second graduate student of biology. A strict IAA of 82% (R), 84% (P), 83% (F) was achieved for the trigger annotation task. The IAA numbers are therefore very promising.

After annotation of event triggers, a domain expert directly annotated pairwise eventive relations between genes and the regulators affecting the expression of these genes. This is presented in the following section.

A.1.3 ROGE Eventive Relations

In the third step of creating GeneReg, ROGE events were annotated in the form of relations between genes and entities affecting the expression of these genes (such as transcription factors, transcription regulators, polymerases and ligands). This third step of GeneReg annotation is based on the GRO class `Regulation of Gene Expression` (`ROGE`) with its two subclasses `Positive Regulation of Gene Expression` (`Positive ROGE`) and `Negative Regulation of Gene Expression` (`Negative ROGE`). The GRO definitions for these concepts are presented in Table A.5.

An annotated instance must contain two arguments, an `Agent`, the entity that modifies gene expression, and `Patient`, the entity whose expression is modified. Agents can be transcription factors (in core regulatory relations), or by polymerases and ligands (in auxiliary regulatory relations).[4] The sentence *"**H-NS** and **StpA** proteins stimulate expression of the **maltose regulon** in Escherichia coli."* contains two `Positive ROGE` instances with *H-NS* and *StpA* as regulators and *maltose regulon* as regulated `Gene Group`. Table A.6 summarizes the overall annotation results.

Table A.5: GRO definitions for regulation of gene expression (ROGE) event annotations in each semantic category.

Semantic Category	Definition
`Regulation of Gene Expression`	Any process that modulates the frequency, rate or extent of gene expression.
`Positive Regulation of Gene Expression`	Any process that activates or increases the frequency, rate or extent of gene expression.
`Negative Regulation of Gene Expression`	Any process that stops, prevents or reduces the frequency, rate or extent of gene expression.

The following is an example of a `ROGE` events from the GeneReg corpus. This formal representation is conform with the entity-driven annotation scheme presented in Section 3.6.4:

(A.3) "*Acid-mediated induction of the asr gene in the Delta(phoB-phoR) mutant strain was restored by introduction of the plasmid with cloned phoB-phoR genes.*"

[4] Auxiliary regulatory relations (606 instances) extend the original GeneReg corpus.

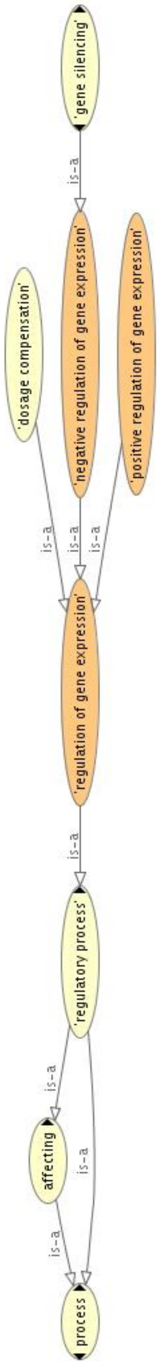

'gene silencing'
is-a
'dosage compensation'
'negative regulation of gene expression'
'positive regulation of gene expression'
is-a
is-a
is-a
'regulation of gene expression'
is-a
'regulatory process'
is-a
affecting
is-a
is-a
process

Figure A.2: Gene Regulation Ontology snapshot for regulation of gene expression process.

(A.4) $\texttt{Positive\_ROGE}(Event_1) \wedge \texttt{Agent}(Event_1,\ phoB) \wedge \texttt{Patient}(Event_1,\ asr)$

(A.5) $\texttt{Positive\_ROGE}(Event_2) \wedge \texttt{Agent}(Event_2,\ phoR) \wedge \texttt{Patient}(Event_2,\ asr)$

As the transcription process is a part of the gene expression process, it was integrated in events during this annotation campaign in the same way as gene expression. This is illustrated in the next example:

(A.6) "*Purified MarA and MalE-SoxS proteins stimulated mar transcription about 6- and 15-fold, respectively, when the RNA polymerase/DNA ratio was 1.*"

(A.7) $\texttt{Positive\_ROGE}(Event_1) \wedge \texttt{Agent}(Event_1,\ MarA) \wedge \texttt{Patient}(Event_1,\ mar)$

(A.8) $\texttt{Positive\_ROGE}(Event_2) \wedge \texttt{Agent}(Event_2,\ SoxS) \wedge \texttt{Patient}(Event_2,\ mar)$

Table A.6: Number of regulation of gene expression (ROGE) event annotations in each semantic category.

Semantic Category	Core	Auxiliary	TOTAL
`ROGE` (unspecified)	417	192	609
`Positive ROGE`	465	325	790
`Negative ROGE`	282	89	371
TOTAL	1164	606	1770

Annotation numbers are presented in Table A.6. The most frequent annotations are the annotations of positive ROGE events followed by unspecified ROGE events. Negative ROGE event annotations occur less frequently than other regulation events. This difference can be explained by the experimental work-flow where positive regulators can be more easily identified after they are modified or deleted from cells. Most annotations are done with transcription factors as agents (core ROGE). A third of annotations contain auxiliary ROGE events with polymerase and ligands as regulators.

A set of 65 randomly selected abstracts was annotated by the second graduate student of biology in order to determine the IAA. An IAA of 78.4% (R), 77.3% (P), 77.8% (F) was measured for the task of correct identification of pairs of interacting named entities in gene regulation processes. IAA of 67% (R), 67.9% (P), 67.4% (F) were achieved for the identification of interacting pairs plus the three-way classification of the interaction relation. The three-way classification of ROGE events in positive, negative and unspecified regulations is hard to perform. But the first IAA numbers show that it is feasible for annotators to correctly identify ROGE events with entities involved and to attribute roles.

GeneReg is freely available for academic purposes at `http://www.julielab.de/`. Licensed under a Creative Commons Attribution-Noncommercial 3.0 Germany.

A.2 Additional Material for Chapter 3

A.2.1 Event Trigger Lists

The following lists contain the most frequent event triggers from the GeneReg and "BioNLP 2009 Shared Task on Event Extraction" corpora for the `Transcription`, `Gene Expression`, `Regulation`, `Positive Regulation` and `Negative Regulation` event types.

`Transcription`: *transcription, expression, express, level, transcribe, transcriptional, transcript, mrna, detect, transcriptional activity, mrna expression, synthesis, induction, mrna levels.*

`Gene Expression`: *expression, express, production, produce, overexpression, synthesis, gene expression, overexpress, level, detect, transfect, cotransfection, induction, present, detectable, product, transfection.*

`Regulation`: *regulate, regulation, effect, control, affect, role, involve, dependent, target, change, modulate, alter, responsible, response, influence, unaffected, modulation, depend, sensitive, involvement, through.*

`Positive Regulation`: *induce, increase, induction, activation, activate, enhance, require, mediate, stimulate, upregulation, overexpression, upregulate, stimulation, result, inducible, lead, augment, dependent, in response to, accumulation, transactivation, necessary, transactivate, overexpress, requirement, promote, essential, cause, transfect, high levels, after, elevate, contribute, cotransfection, role, important, high, active, through, activity, trigger, transfection, responsible.*

`Negative Regulation`: *inhibit, decrease, inhibition, reduce, block, supress, prevent, inhibitor, downregulate, abolish, reduction, downregulation, bloc, loss, impair, repress, suppression, diminish, downregulation, decline, deprivation, defective, abrogate, inhibitory effect, repression, lack, absence, negative regulation, attenuate.*

A.2.2 Definitions of Event Types from "BioNLP 2009 Shared Task on Event Extraction"

Table A.7 contains definitions of the "BioNLP 2009 Shared Task on Event Extraction" event types from the Genia ontology.

Table A.7: Definition of BioNLP Shared Task event types. The definitions are taken from the GENIA ontology.

Event Class Name	Event Definition
`Localization`	The processes by which a cell, a substance, or a cellular entity, such as a protein complex or organelle, is transported to or maintained in, a specific location.
`Protein Catabolism`	The breakdown into simpler components of a protein by the destruction of the native, active configuration, with or without the hydrolysis of peptide bonds.
`Phosphorylation`	The process of attaching a phosphoric group to a protein.
`Gene Expression`	The process by which genetic material undergoes a detectable and heritable structural change. There are three categories of mutation: genome mutations, involving addition or subtraction of one or more whole chromosomes; chromosome mutations, which alter the structure of chromosomes; and gene mutations, where the structure of a gene is altered at the molecular level.
`Transcription`	The synthesis of either RNA on a template of DNA or DNA on a template of RNA.
`Binding`	The selective, often stoichiometric interaction of a molecule with one or more specific sites on another molecule.
`Positive Regulation`	Any process that activates or increases the rate, frequency or extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule.
`Negative Regulation`	Any process that stops, prevents or reduces the rate, frequency or extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule.
`Regulation`	Any process that mediates interactions between a cell and its surroundings. Encompasses interactions such as signaling or attachment between one cell and another cell, between a cell and an extracellular matrix, or between a cell and any other aspect of its environment.

A.3 Additional Material for Chapter 5

A.3.1 Learning Curves for Event Types from "BioNLP 2009 Shared Task on Event Extraction"

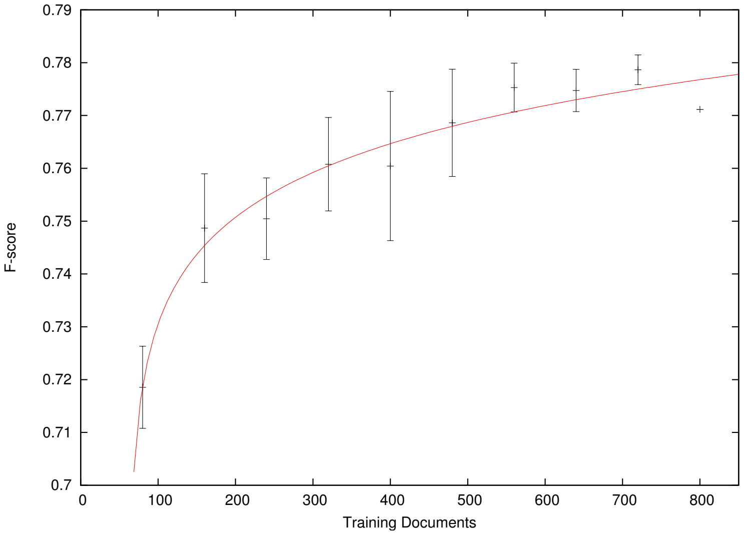


Figure A.3: Learning curve for argument extraction in `Gene Expression` and `Transcription` events on the Shared Task development data. Logarithmic function is applied.

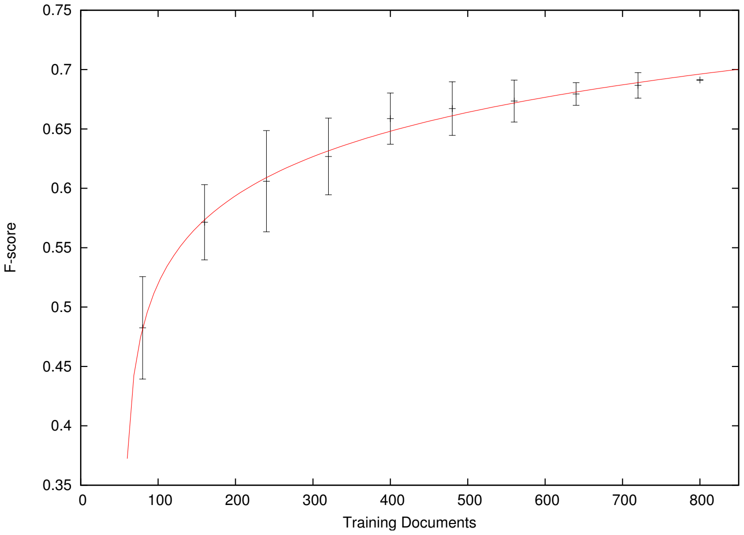


Figure A.4: Learning curve for argument extraction in `Binding` events on the Shared Task development data. Logarithmic function is applied.

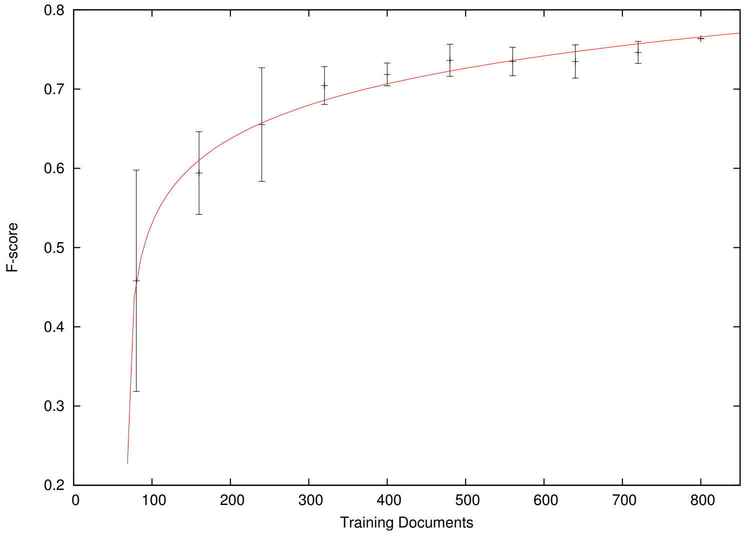


Figure A.5: Learning curve for argument extraction in `Localization` events on the Shared Task development data. Logarithmic function is applied.

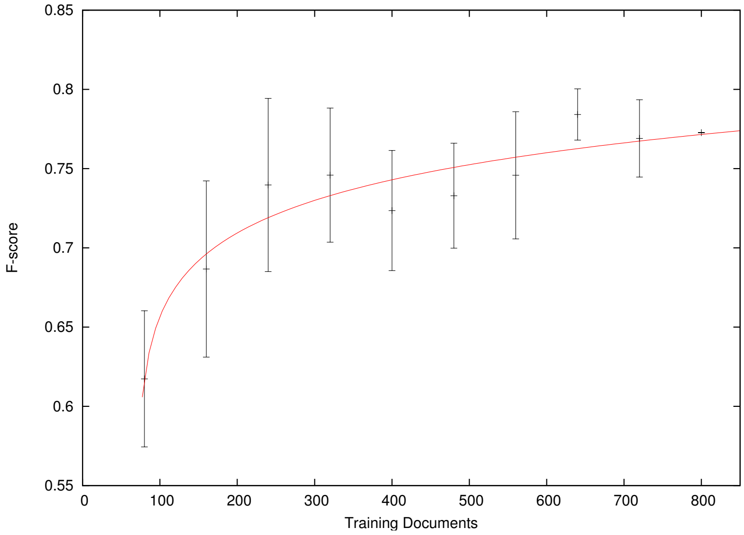


Figure A.6: Learning curve for argument extraction in `Phosphorylation` events on the Shared Task development data. Logarithmic function is applied.

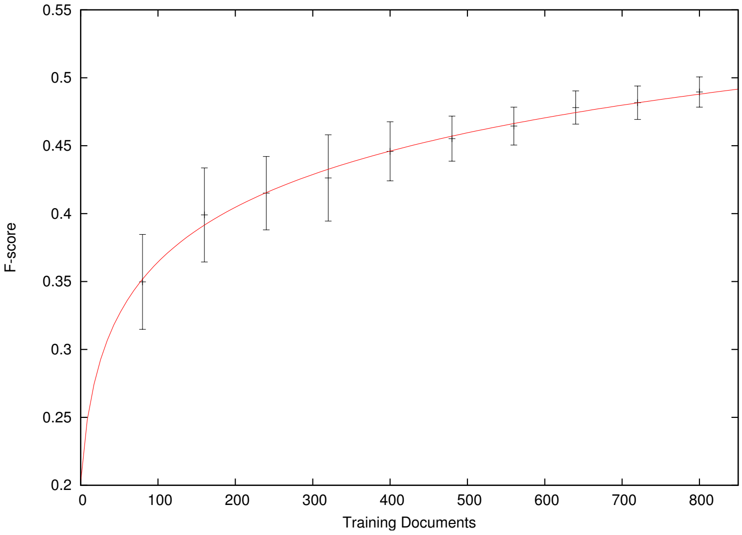


Figure A.7: Learning curve for argument extraction in `Negative Regulation` events on the Shared Task development data. Logarithmic function is applied.

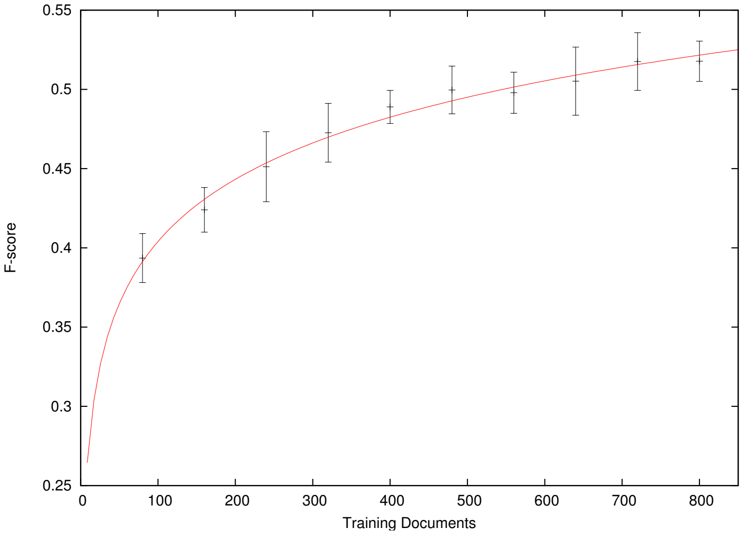


Figure A.8: Learning curve for argument extraction in `Regulation` events on the Shared Task development data. Logarithmic function is applied.

Glossary

Accuracy

Accuracy is defined formally as:

$$Accuracy = \frac{tp}{n} \tag{A.9}$$

where tp (true positives) are correctly predicted positive instances, and n is the size of instance set to be classified.

Concept

A unit of thought constituted through abstraction on the basis of properties common to a set of objects.

Dictionary

Structured collection of lexical units, with linguistic information about each of them.

DNA

Deoxyribonucleic acid (DNA) is a nucleic acid containing the genetic instructions used in the development and functioning of all known living organisms (with the exception of RNA viruses).

DNA sequencing

DNA sequencing includes several methods and technologies that are used for determining the order of the nucleotide bases—adenine, guanine, cytosine, and thymine—in a molecule of DNA.

F-score

The F-score metric combines recall and precision measures in the harmonic mean of both metrics. The traditional F-score used in this thesis is defined as follows:

$$\text{F-score} = \frac{2(precision \times recall)}{precision + recall} \tag{A.10}$$

Gene

A gene is a molecular unit of heredity of a living organism. It is a name given to some stretches of DNA and RNA that code for a polypeptide or for an RNA chain that has a function in the organism.

Inter-Annotator Agreement

The Inter-Annotator Agreement statistics provide evidence for the decision made by human annotators in corpora. Kappa coefficient is a statistical measure of inter-rater agreement or inter-annotator agreement for qualitative (categorical) items.

Machine Learning

Mitchell (Mitchell, 1997) formulates the fundamental idea of machine learning (ML) in the following sentence – "*We say that a machine learns with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E.*". The aim of ML is thus to study the systems that automatically improve with experience.

Model Organism

A model organism is a non-human species that is extensively studied to understand particular biological phenomena, with the expectation that discoveries made in the organism model will provide insight into the workings of other organisms

Name

Designation of an object by a linguistic expression

Object

Any part of the perceivable or conceivable world.

Ontology

Ontology is a formal specification of a conceptualization.

Precision

Precision is defined formally as:

$$Precision = \frac{tp}{tp + fp} \tag{A.11}$$

where tp (true positives) are correctly predicted positive instances, fp (false positives) are unexpected positive instances, and fn (false negatives) are not identfied positive instances.

Protein

Proteins are biochemical compounds consisting of one or more polypeptides typically folded into a globular or fibrous form, facilitating a biological function.

Recall

Recall is defined formally as:

$$Recall = \frac{tp}{tp + fn} \tag{A.12}$$

where tp (true positives) are correctly predicted positive instances, fp (false positives) are unexpected positive instances, and fn (false negatives) are not identfied positive instances.

RNA

Ribonucleic acid, or RNA, is one of the three major macromolecules (along with DNA and proteins) that are essential for all known forms of life.

Semi-supervised Machine Learning

Semi-supervised learning is halfway between supervised learning (with labeled training data) and unsupervised learning (with unlabeled training data). The semi-supervised approach makes use of both labeled and unlabeled training data as the ML community has found that enrichment of a labeled data set with unlabeled data can improve a system's accuracy.

Supervised Machine Learning

In a *supervised* learning approach, the system is given an input set of labeled input examples called training data. The goal is to learn to output correct labels for unseen examples, which means that the model generalizes over training data.

Term

Designation of a defined concept in a special language by a linguistic expression.

Terminology

Set of terms representing the system of concepts or a particular subject field.

Unsupervised Machine Learning

In the *unsupervised* approach the input examples are given to the learner without any target labeling or feedback from its environment. The main technique used in the unsupervised approach is clustering to discover similar groups in the data.

Vocabulary

Dictionary containing the terminology of a subject field.

Acronyms

ACE	Automatic Content Extraction Programme
BioNLP	Biomedical Natural Language Processing
CoNLL	Conference on Natural Language Learning
CRF	Conditional Random Fields
GeneReg	Gene Regulation Corpus
GO	Gene Ontology
GRO	Gene Regulation Ontology
IAA	Inter-Annotator Agreement
IE	Information Extraction
JReX	Jena Relation eXtraction
MaxEnt	Maximum Entropy
ML	Machine Learning
MUC	Message Understanding Conference
NER	Named Entity Recognition
NLP	Natural Language Processing
PMID	PubMed identifier
PPI	Protein-protein interaction
PTB	Penn Treebank
ROGE	Regulation of Gene Expression
SRL	Semantic Role Labeling
SVM	Support Vector Machine
WSJ	Wall Street Journal

Bibliography

[ACE-Event-Annotation-Guidelines 2005] ACE (Automatic Content Extraction) English Annotation Guidelines for Events / Linguistic Data Consortium. 2005. – Technical Report.

[Abney 1996] Abney, Steven: Partial Parsing via Finite State Cascades. In: *Natural Language Engineering* 2 (1996), No. 4, pp. 337–344.

[Airola et al. 2008a] Airola, Antti; Pyysalo, Sampo; Björne, Jari; Pahikkala, Tapio; Ginter, Filip; Salakoski, Tapio: A Graph Kernel for Protein-Protein Interaction Extraction. In: *BioNLP 2008 – Proceedings of the ACL/HLT 2008 Workshop on Current Trends in Biomedical Natural Language Processing.* Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 1–9. – URL `http://www.aclweb.org/anthology/W/W08/W08-0601`.

[Airola et al. 2008b] Airola, Antti; Pyysalo, Sampo; Björne, Jari; Pahikkala, Tapio; Ginter, Filip; Salakoski, Tapio: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. In: *BMC Bioinformatics* 9(Suppl 11):S2 (2008).

[Alfarano et al. 2005] Alfarano, C.; Andrade, C. E.; Anthony, K.; Bahroos, N.; Bajec, M.; Bantoft, K.; Betel, Doron; Bobechko, B.; Boutilier, K.; Burgess, E.; Buzadzija, K.; Cavero, R.; D'Abreo, C.; Donaldson, Ian; Dorairajoo, D.; Dumontier, M. J.; Dumontier, M. R.; Earles, V.; Farrall, R.; Feldman, Howard J.; Garderman, E.; Gong, Y.; Gonzaga, R.; Grytsan, V.; Gryz, E.; Gu, V.; Haldorsen, E.; Halupa, A.; Haw, R.; Hrvojic, A.; Hurrell, L.; Isserlin, Ruth; Jack, F.; Juma, F.; Khan, A.; Kon, T.; Konopinsky, S.; Le, V.; Lee, E.; Ling, S.; Magidin, M.; Moniakis, J.; Montojo, J.; Moore, S.; Muskat, B.; Ng, I.; Paraiso, J. P.; Parker, B.; Pintilie, Greg; Pirone, R.; Salama, John J.; Sgro, S.; Shan, T.; Shu, Y.; Siew, J.; Skinner, D.; Snyder, Kevin A.; Stasiuk, R.; Strumpf, D.; Tuekam, Brigitte; Tao, S.; Wang, Z.; White, M.; Willis, R.; Wolting, Cheryl; Wong, S.; Wrong, A.; Xin, C.; Yao, R.; Yates, B.; Zhang, S.; Zheng, K.; Pawson, Tony; Ouellette, B. F. F.; Hogue, Christopher W. V.: The Biomolecular Interaction Network Database and related tools 2005 update. In: *Nucleic Acids Research* 33(Database Issue) (2005), pp. D418–D424.

[Allan et al. 1998] Allan, James; Carbonell, Jaime; Doddington, George; Yamron, Jonathan; Yang, Yiming: Topic Detection and Tracking Pilot Study: Final Report. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Lansdowne, VA, USA, 1998, pp. 194–218.

[Ando 2007] Ando, Rie K.: BioCreative II gene mention tagging system at IBM Watson. In: *Proceedings of the Second BioCreative Challenge Evaluation*, 2007, pp. 101–103.

[Bach 1986] Bach, Emmon: The Algebra of Events. In: *Linguistics and Philosophy* 9 (1986), pp. 5–16.

[Backus 1959] Backus, John W.: The syntax and semantics of the proposed international algebraic language of the Zurch ACM-GAMM Conference. In: *Information Processing: Proceedings of the International Conference on Information Processing, Paris*, UNESCO, June 1959, pp. 125–132.

[Baker et al. 2007] Baker, Collin; Ellsworth, Michael; Erk, Katrin: SemEval-2007 Task 19: Frame Semantic Structure Extraction. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 99–104. – URL `http://www.aclweb.org/anthology/S/S07/S07-1018`.

[Baker et al. 2003] Baker, Collin F.; Fillmore, Charles J.; Cronin, Beau: The Structure of the FrameNet Database. In: *International Journal of Lexicography* 16 (2003), No. 3, pp. 281–296.

[Baumgartner et al. 2007] Baumgartner, William A.; Cohen, Bretonnel K.; Fox, Lynne M.; Acquaah-Mensah, George; Hunter, Lawrence: Manual curation is not sufficient for annotation of genomic databases. In: *ISMB/ECCB (Supplement of Bioinformatics)*, 2007, pp. 41–48.

[Beisswanger et al. 2008] Beisswanger, Elena; Lee, Vivian; Kim, Jung-jae; Rebholz-Schuhmann, Dietrich; Splendiani, Andrea; Dameron, Olivier; Schulz, Stefan; Hahn, Udo: Gene Regulation Ontology (GRO): Design principles and use cases. In: *Studies in Health Technology and Informatics* 136 (2008), pp. 9–14.

[Bellman 1958] Bellman, Richard: On a Routing Problem. In: *Quarterly of Applied Mathematics* 16 (1958), pp. 87–90.

[Bennett 1988] Bennett, Jonathan: *Events and Their Names*. Oxford University Press, 1988.

[Bennett 1996] BENNETT, Jonathan: What Events Are. In: *Events* (1996), pp. 137–151.

[Berger et al. 1996] BERGER, Adam; PIETRA, Stephen D.; PIETRA, Vincent D.: A maximum entropy approach to natural language processing. In: *Computational Linguistics* 22 (1996), No. 1, pp. 39–71.

[Bethard et al. 2008] BETHARD, Steven; LU, Zhiyong; MARTIN, James H.; HUNTER, Lawrence: Semantic role labeling for protein transport predicates. In: *BMC Bioinformatics* 9:277 (2008).

[Bies et al. 2005] BIES, Ann; KULICK, Seth; MANDEL, Mark: Parallel Entity and Treebank Annotation. In: *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky.* Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 21–28. – URL `http://www.aclweb.org/anthology/W/W05/W05-0304`.

[Bikel 2004] BIKEL, Daniel M.: Intricacies of Collins' parsing model. In: *Computational Linguistics* 30 (2004), No. 4, pp. 479–511.

[Björne et al. 2009] BJÖRNE, Jari; HEIMONEN, Juho; GINTER, Filip; AIROLA, Antti; PAHIKKALA, Tapio; SALAKOSKI, Tapio: Extracting Complex Biological Events with Rich Graph-Based Feature Sets. In: *Proceedings BioNLP 2009. Companion Volume: Shared Task on Event Extraction.* Boulder, Colorado, USA: Association for Computational Linguistics, June 2009, pp. 10–18. – URL `http://www.aclweb.org/anthology/W09-1402`.

[Blanco and Moldovan 2011] BLANCO, Eduardo; MOLDOVAN, Dan: Unsupervised Learning of Semantic Relation Composition. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 1456–1465. – URL `http://www.aclweb.org/anthology/P11-1146`.

[Blaschke et al. 1999] BLASCHKE, Christian; ANDRADE, Miguel A.; OUZOUNIS, Christos A.; VALENCIA, Alfonso: Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions. In: *ISMB 1999 – Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology.* Heidelberg, Germany, August 1999, pp. 60–67.

[Bloomfield 1914] BLOOMFIELD, Leonard: *An Introduction to the Study of Language.* New York: Henry Holt and Company, 1914.

[Bos et al. 2008] BOS, Johan; BRISCOE, Edward; CAHILL, Aoife; CARROLL, John; CLARK, Stephen; COPESTAKE, Ann; FLICKINGER, Dan; GENABITH, Josef van;

Hockenmaier, Julia; Joshi, Aravind; Kaplan, Ronald; King, Tracy H.; Kuebler, Sandra; Lin, Dekang; Loenning, Jan T.; Manning, Christopher; Miyao, Yusuke; Nivre, Joakim; Oepen, Stephan; Sagae, Kenji; Xue, Nianwen; Zhang, Yi (Editors.): *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation.* Manchester, UK: Coling 2008 Organizing Committee, August 2008. – URL `http://www.aclweb.org/anthology/W08-13`.

[Brill 1995] Brill, Eric: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. In: *Computational Linguistics* 21 (1995), No. 4, pp. 543–565.

[Brinker 1977] Brinker, Klaus: *Modelle und Methoden der strukturalistischen Syntax.* Kohlhammer, 1977.

[Browne et al. 1998] Browne, Allen C.; Divita, Guy; Nguyen, Van; Cheng, Vincent C.: Modular text processing system based on the Specialist lexicon and lexical tools. In: Chute, C. G. (Editors.): *AMIA'98 – Proceedings of the 1998 AMIA Annual Fall Symposium. A Paradigm Shift in Health Care Information Systems: Clinical Infrastructures for the 21st Century.* Orlando, Fl: Philadelphia, PA: Hanley & Belfus, November 1998, pp. 982.

[Buchholz and Marsi 2006] Buchholz, Sabine; Marsi, Erwin: CoNLL-X Shared Task on Multilingual Dependency Parsing. In: *CoNLL-X – Proceedings of the 10th Conference on Computational Natural Language Learning.* New York City, N.Y.: Association for Computational Linguistics, June 2006, pp. 149–164.

[Bunescu et al. 2005] Bunescu, Razvan; Ge, Ruifang; Kate, Rohit J.; Marcotte, Edward M.; Mooney, Raymond J.; Ramani, Arun K.; Wong, Yuk W.: Comparative experiments on learning information extractors for proteins and their interactions. In: *Artificial Intelligence in Medicine* 33 (2005), No. 2, pp. 139–155.

[Bunescu and Mooney 2005] Bunescu, Razvan; Mooney, Raymond J.: A Shortest Path Dependency Kernel for Relation Extraction. In: *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.* Vancouver, British Columbia, Canada: Association for Computational Linguistics, October 2005, pp. 724–731. – URL `http://www.aclweb.org/anthology/H/H05/H05-1091`.

[Bunescu and Mooney 2007] Bunescu, Razvan; Mooney, Raymond J.: Extracting Relations from Text. From Word Sequences to Dependency Paths. In: Kao, A.; Poteet, S. (Editors.): *Natural Language Processing and Text Mining.* Berlin: Springer Verlag, 2007, pp. 29–44.

[Buyko et al. 2008] Buyko, Ekaterina; Beisswanger, Elena; Hahn, Udo: Testing Different Ace-Style Feature Sets for the Extraction of Gene Regulation Relations from Medline Abstracts. In: *SMBM 2008 – Proceedings of the 3rd International Symposium on Semantic Mining in Biomedicine.* Turku, Finland, September 2008, pp. 21–28.

[Buyko et al. 2012] Buyko, Ekaterina; Beisswanger, Elena; Hahn, Udo: The Extraction of Pharamacogenetic and Pharmacogenomic Relations – A Case Study Using PharmGKB. In: *PSB 2012 – Proceedings of the Pacific Conference on Biocomputing*, 2012.

[Buyko et al. 2009] Buyko, Ekaterina; Faessler, Erik; Wermter, Joachim; Hahn, Udo: Event Extraction from Trimmed Dependency Graphs. In: *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 19–27. – URL http://www.aclweb.org/anthology/W/W09/W09-1403.

[Buyko et al. 2011a] Buyko, Ekaterina; Faessler, Erik; Wermter, Joachim; Hahn, Udo: Syntactic Simplification and Semantic Enrichment - Trimming Dependency Graphs for Event Extraction. In: *Computational Intelligence* 27 (2011), No. 4, pp. 610–644.

[Buyko and Hahn 2008] Buyko, Ekaterina; Hahn, Udo: Fully embedded type systems for the semantic annotation layer. In: *ICGL 2008 – Proceedings of the 1st International Conference on Global Interoperability for Language Resources*, Hong Kong, SAR, January 9-11, 2008. City University of Hong Kong, 2008, pp. 26–33.

[Buyko and Hahn 2010] Buyko, Ekaterina; Hahn, Udo: Evaluating the Impact of Alternative Dependency Graph Encodings on Solving Event Extraction Tasks. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.* Cambridge, MA: Association for Computational Linguistics, October 2010, pp. 982–992. – URL http://www.aclweb.org/anthology/D/D10/D10-1096.

[Buyko and Hahn 2011] Buyko, Ekaterina; Hahn, Udo: Generating Semantics for the Life Sciences via Text Analytics. In: *ICSC 2011 – Procseedings of the 5th IEEE International Conference on Semantic Computing*, Stanford University, Palo Alto, CA, USA, September 19-21, 2011. IEEE Computer Society Press, 2011, pp. 193–196.

[Buyko et al. 2011b] Buyko, Ekaterina; Linde, Jörg; Priebe, Steffen; Hahn, Udo: Towards Automatic Pathway Generation from Biological Full-Text Publications. In: Gama, João; Bradley, Elizabeth; Hollmén, Jaakko (Editors.): *IDA* Vol. 7014, Springer, 2011, pp. 67–79.

[Buyko et al. 2007] Buyko, Ekaterina; Tomanek, Katrin; Hahn, Udo: Resolution of coordination ellipses in biological named entities using conditional random fields. In: *PACLING 2007 – Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics.* Melbourne, Australia: Pacific Association for Computational Linguistics, 2007, pp. 163–171.

[Buyko et al. 2006] Buyko, Ekaterina; Wermter, Joachim; Poprat, Michael; Hahn, Udo: Automatically Adapting an NLP Core Engine to the Biology Domain. In: *Proceedings of the Joint BioLINK-Bio-Ontologies Meeting. A Joint Meeting of the ISMB Special Interest Group on Bio-Ontologies and the BioLINK Special Interest Group on Text Data Mining in Association with ISMB 2006.* Fortaleza, Brazil, August 2006, pp. 65–68.

[Carreras and Màrquez 2004] Carreras, Xavier; Màrquez, Lluís: Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In: *CoNLL-2004 – Proceedings of the 8th Conference on Computational Natural Language Learning in association with HLT/NAACL 2004.* Boston, MA, USA, May 2004, pp. 89–97.

[Carreras and Màrquez 2005] Carreras, Xavier; Màrquez, Lluís: Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In: *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005).* Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 152–164. – URL `http://www.aclweb.org/anthology/W/W05/W05-0620`.

[Casati and Varzi 2010] Casati, Roberto; Varzi, Achille: Events. In: Zalta, Edward N. (Editors.): *The Stanford Encyclopedia of Philosophy.* Spring 2010. 2010.

[Castaño et al. 2002] Castaño, Josè; Zhang, Jason; Pustejovsky, James D.: Anaphora Resolution in Biomedical Literature. In: *Proceedings of The International Symposium on Reference Resolution for Natural Language Processing.* Alicante, Spain, June 2002.

[Ceol et al. 2008] Ceol, Arnaud; Chatr-Aryomontri, Andrew; Licata, Luana; Cesareni, Gianni: Linking Entries in Protein Interaction Database to Structured Text: The FEBS Letters Experiment. In: *FEBS Letters* 582 (2008), April, pp. 1171–1177.

[Cer et al. 2010] Cer, Daniel; de Marneffe, Marie-Catherine; Jurafsky, Dan; Manning, Chris: Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy. In: *LREC'2010 – Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta, May 2010, pp. 1628–1632.

[Chang and Lin 2001] Chang, Chih-Chung; Lin, Chih-Jen: *LIBSVM: a library for support vector machines*, 2001. – Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Charniak and Johnson 2005] Charniak, Eugene; Johnson, Mark: Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In: *ACL'05 – Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI, USA, June 2005, pp. 173–180.

[Chen and Sharp 2004] Chen, Hao; Sharp, Burt M.: Content-rich biological network constructed by mining PubMed abstracts. In: *BMC Bioinformatics* 5:147 (2004).

[Chinchor 1998] Chinchor, Nancy: Overview of MUC-7. In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference held in Fairfax, VA*, May 1998.

[Chomsky 1956] Chomsky, Noam: Three models for the description of language. In: *IRE Transactions on Information Theory* 2 (1956), No. 3, pp. 113–124.

[Clegg and Shepherd 2005] Clegg, Andrew B.; Shepherd, Adrian J.: Evaluating and integrating Treebank parsers on a biomedical corpus. In: *Proceedings of the ACL 2005 Workshop on Software*. Ann Arbor, MI, USA, June 2005, pp. 14–33.

[Cohen and Hunter 2008] Cohen, Bretonnel K.; Hunter, Lawrence: Getting started in text mining. In: *PLoS Computational Biology* 4 (2008), January, No. 1, pp. e20.

[Cohen 2010] Cohen, Kevin B.: BioNLP: Biomedical Text Mining. In: Nitin Indurkhya, Fred J. D. (Editors.): *Handbook of Natural Language Processing*. 2010, pp. 605–626.

[Cohen and Hunter 2006] Cohen, Kevin B.; Hunter, Lawrence: A critical review of PASBio's argument structures for biomedical verbs. In: *BMC Bioinformatics* 7 (Suppl 3):S5 (2006), pp. S5.

[Cohn and Lapata 2008] Cohn, Trevor; Lapata, Mirella: Sentence Compression Beyond Word Deletion. In: *COLING 2008 – Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, UK: Coling 2008 Organizing Committee, August 2008, pp. 137–144. – URL `http://www.aclweb.org/anthology/C08-1018`.

[Cohn and Lapata 2009] Cohn, Trevor; Lapata, Mirella: Sentence Compression as Tree Transduction. In: *Journal of Artificial Intelligence Research* 34 (2009), pp. 637–674.

[Collins et al. 2003] Collins, Francis; Green, Eric; Guttmacher, Alan; Guyer, Mark: A vision for the future of genomics research. In: *Nature* 422 (2003), No. 6934 (24 Feb), pp. 835–847.

[Collins 2003] Collins, Michael: Head-Driven Statistical Models for Natural Language Parsing. In: *Computational Linguistics* 29 (2003), No. 4, pp. 589–637.

[Croft 1990] Croft, William: Possible Verbs and the Structure of Events. In: Tsohatzidis, S. L. (Editors.): *Meanings and Prototypes: Studies in Linguistic Categorization*. Routledge, London, 1990, pp. 48–73.

[Culotta and Sorensen 2004] Culotta, Aron; Sorensen, Jeffrey: Dependency Tree Kernels for Relation Extraction. In: *ACL'04 – Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Main Volume*. Barcelona, Spain, July 2004, pp. 423–429. – URL http://www.aclweb.org/anthology/P04-1054.

[Curran et al. 2007] Curran, James; Clark, Stephen; Bos, Johan: Linguistically Motivated Large-Scale NLP with C&C and Boxer. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 33–36. – URL http://www.aclweb.org/anthology/P07-2009.

[Davidson 1967] Davidson, Donald: The logical form of action sentences. In: Rescher, Nicolas (Editors.): *The Logic of Decision and Action*. Pittsburgh, PA: University of Pittsburgh Press, 1967, pp. 81–95.

[Davidson 1980] Davidson, Donald: *Essays on Actions and Events*. Oxford University Press, 1980.

[de Marneffe et al. 2006] de Marneffe, Marie-Catherine; MacCartney, Bill; Manning, Christopher D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: *LREC'2006 – Proceedings of the 5th International Conference on Language Resources and Evaluation*. Genoa, Italy, May 2006, pp. 449–454.

[Dijkstra 1959] Dijkstra, Edsger W.: A note on two problems in connexion with graphs. In: *Numerische Mathematik* 1 (1959), pp. 269–271.

[Ding et al. 2002] Ding, Jing; Berleant, Daniel; Nettleton, Dan; Wurtele, Eve S.: Mining MEDLINE: Abstracts, Sentences, or Phrases? In: *Pacific Symposium on Biocomputing*, 2002, pp. 326–337.

[Doddington et al. 2004] Doddington, George; Mitchell, Alexis; Przybocki, Mark; Ramshaw, Lance; Strassel, Stephanie; Weischedel, Ralph M.: The Automatic Content Extraction (Ace) Program: Tasks, data and evaluation. In: *LREC 2004 – Proceedings of the 4th International Conference on Language Resources and Evaluation. In Memory of Antonio Zampolli. Vol. 3.* Lisbon, Portugal, May 2004, pp. 837–840.

[Dolbey 2009] Dolbey, Andrew: *BioFrameNet: a FrameNet Extension to the Domain of Molecular Biology*, UC Berkeley, phdDissertation, 2009.

[Donaldson et al. 2003] Donaldson, Ian; Martin, Joel D.; Bruijn, Berry de; Wolting, Cheryl; Lay, Vicki; Tuekam, Brigitte; Zhang, Shudong; Baskin, Berivan; Bader, Gary D.; Michalickova, Katerina; Pawson, Tony; Hogue, Christopher W. V.: PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. In: *BMC Bioinformatics* 4:11 (2003).

[Dowty 1979] Dowty, David R.: *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ.* Springer, 1979.

[Dowty 1991] Dowty, David R.: Thematic proto-roles and argument selection. In: *Language* 67 (1991), September, No. 3, pp. 547–619.

[Faessler 2009] Faessler, Erik: *Automatische Extraktion von Protein-Protein-Interaktionen in biomedizinischen Texten unter Verwendung von Supportvektormaschinen.* 2009. – URL `http://www.julielab.de/coling_multimedia/de/downloads/Papers/diploma_thesis_faessler.pdf`.

[Fanselow and Felix 1990] Fanselow, Gisbert; Felix, Sascha W.: *Sprachtheorie. Eine Einführung in die Generative Grammatik. Band 1: Grundlagen und Zielsetzungen.* Francke, 1990.

[Fellbaum 1998] Fellbaum, Christiane (Editors.): *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press, 1998.

[Filippova and Strube 2008] Filippova, Katja; Strube, Michael: Sentence Fusion via Dependency Graph Compression. In: *EMNLP'08 – Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing.* Honolulu, Hawaii: Association for Computational Linguistics, October 2008, pp. 177–185. – URL `http://www.aclweb.org/anthology/D08-1019`.

[Fillmore 1968] Fillmore, Charles J.: The Case for Case. In: Bach, Emmon W.; Harms, Robert T. (Editors.): *Universals in Linguistic Theory.* New York: Holt, Rinehart & Winston, 1968, pp. 1–88.

[Fillmore and Atkins 1992] Fillmore, Charles J.; Atkins, Beryl T.: Towards a frame-based lexicon: the case of RISK. In: Lehrer, A.; Kitta, E. (Editors.): *Frames and Fields.* New York: Erlbaum Publishers, 1992, pp. 75–102.

[Finkel and Manning 2009] Finkel, Jenny R.; Manning, Christopher D.: Joint Parsing and Named Entity Recognition. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 326–334. – URL `http://www.aclweb.org/anthology/N/N09/N09-1037`.

[Francis and Kučera 1979] Francis, Nelson W.; Kučera, Henry: Brown Corpus Manual / Department of Linguistics, Brown University, Providence, Rhode Island, US. URL `http://icame.uib.no/brown/bcm.html`, 1979. – Technical Report.

[Friedman et al. 2002] Friedman, Carol; Kra, Pauline; Rzhetsky, Andrey: Two biomedical sublanguages: a description based on the theories of Zellig Harris. In: *Journal of Biomedical Informatics* 35 (2002), No. 4, pp. 222–235.

[Fundel et al. 2007] Fundel, Katrin; Küffner, Robert; Zimmer, Ralf: RelEx – Relation extraction using dependency parse trees. In: *Bioinformatics* 23 (2007), No. 3, pp. 365–371.

[Fürstenau and Lapata 2009] Fürstenau, Hagen; Lapata, Mirella: Semi-Supervised Semantic Role Labeling. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009).* Athens, Greece: Association for Computational Linguistics, March 2009, pp. 220–228. – URL `http://www.aclweb.org/anthology/E09-1026`.

[Galley and McKeown 2007] Galley, Michel; McKeown, Kathleen: Lexicalized Markov Grammars for Sentence Compression. In: *HLT-NAACL 2007 – Proceedings of the 7th International Conference on Human Language Technology Research and the 8th Conference of the North American Chapter of the Association for Computational Linguistics.* Rochester, NY, USA, April 2007, pp. 180–187.

[Gama-Castro et al. 2011] Gama-Castro, Socorro; Salgado, Heladia; Peralta-Gil, Martín; Santos-Zavaleta, Alberto; Muñiz-Rascado, Luis; Solano-Lira, Hilda; Jiménez-Jacinto, Verónica; Weiss, Verena; García-Sotelo, Jair S.; López-Fuentes, Alejandra; Porrón-Sotelo, Liliana; Alquicira-Hernández, Shirley; Medina-Rivera, Alejandra; Martínez-Flores, Irma; Alquicira-Hernández, Kevin; Martínez-Adame, Ruth; Bonavides-Martínez, César; Miranda-Ríos, Juan; Huerta, Araceli M.; Mendoza-Vargas, Alfredo; Collado-Torres, Leonardo; Taboada, Blanca;

VEGA-ALVARADO, Leticia; OLVERA, Maricela; OLVERA, Leticia; GRANDE, Ricardo; MORETT, Enrique; COLLADO-VIDES, Julio: RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units). In: *Nucleic Acids Research* 39 (2011), No. Database-Issue, pp. 98–105.

[Gasperin 2006] GASPERIN, Caroline: Semi-supervised anaphora resolution in biomedical texts. In: *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*. New York, New York: Association for Computational Linguistics, June 2006, pp. 96–103. – URL `http://www.aclweb.org/anthology/W/W06/W06-3316`.

[Girju et al. 2006] GIRJU, Roxana; BADULESCU, Adriana; MOLDOVAN, Dan: Automatic Discovery of Part-Whole Relations. In: *Computational Linguistics* 32 (2006), No. 1, pp. 83–135. – ISSN 0891-2017.

[Girju et al. 2007] GIRJU, Roxana; NAKOV, Preslav; NASTASE, Vivi; SZPAKOWICZ, Stan; TURNEY, Peter; YURET, Deniz: SemEval-2007 Task 04: Classification of Semantic Relations between Nominals. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 13–18. – URL `http://www.aclweb.org/anthology/S/S07/S07-1003`.

[Götz and Suhre 2004] GÖTZ, Thilo; SUHRE, Oliver: Design and implementation of the UIMA Common Analysis System. In: *IBM Systems Journal* 43 (2004), No. 3, pp. 476–489.

[Grenon 2006] GRENON, Pierre: Temporal Qualification and Change with First-Order Binary Predicates. In: *Formal Ontologies in Infromation Systems – FOIS* Vol. 150, IOS Press, 2006, pp. 155–166.

[Grewendorf et al. 1990] GREWENDORF, Günther; HAMM, Fritz; STERNEFELD, Wolfgang: *Sprachliches Wissen. Eine Einführung in moderne Theorien der grammatischen Beschreibung*. 4th Edition. Frankfurt am Main: Suhrkamp, 1990.

[Grishman and Sundheim 1996] GRISHMAN, Ralph; SUNDHEIM, Beth: Message Understanding Conference – 6: A brief history. In: *COLING'96 – Proceedings of the 16th International Conference on Computational Linguistics* Vol. 1, Copenhagen, Denmark, August 1996, pp. 466–471.

[Hacioglu et al. 2004] HACIOGLU, Kadri; PRADHAN, Sameer; WARD, Wayne; MARTIN, James H.; JURAFSKY, Daniel: Semantic Role Labeling by Tagging Syntactic Chunks. In: NG, Hwee T.; RILOFF, Ellen (Editors.): *HLT-NAACL*

2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004). Boston, Massachusetts, USA: Association for Computational Linguistics, May 6 - May 7 2004, pp. 110–113.

[Hahn et al. 2008] Hahn, Udo; Buyko, Ekaterina; Landefeld, Rico; Mühlhausen, Matthias; Poprat, Michael; Tomanek, Katrin; Wermter, Joachim: An Overview of JCoRe, the JULIE Lab UIMA Component Repository. In: *Proceedings of the LREC'08 Workshop Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP.* Marrakech, Morocco, May 2008, pp. 1–7.

[Hahn et al. 2007a] Hahn, Udo; Buyko, Ekaterina; Tomanek, Katrin; Piao, Scott; McNaught, John; Tsuruoka, Yoshimasa; Ananiadou, Sophia: An Annotation Type System for a Data-Driven NLP Pipeline. In: *The LAW at ACL 2007 – Proceedings of the Linguistic Annotation Workshop*, Prague, Czech Republic, June 28-29, 2007. Stroudsburg, PA: Association for Computational Linguistics, 2007, pp. 33–40.

[Hahn et al. 2009] Hahn, Udo; Tomanek, Katrin; Buyko, Ekaterina; Kim, Jung J.; Rebholz-Schuhmann, Dietrich: How Feasible and Robust is the Automatic Extraction of Gene Regulation Events? A Cross-Method Evaluation under Lab and Real-Life Conditions. In: *Proceedings of the NAACL workshop on BioNLP 2009*, Association for Computational Linguistics, 2009, pp. 37–45.

[Hahn and Wermter 2004] Hahn, Udo; Wermter, Joachim: High-performance tagging on medical texts. In: *COLING Geneva 2004 – Proceedings of the 20th International Conference on Computational Linguistics*, Association for Computational Linguistics, August 2004, pp. 973–979.

[Hahn and Wermter 2006] Hahn, Udo; Wermter, Joachim: Levels of natural language processing for text mining. In: Ananiadou, Sophia; McNaught, John (Editors.): *Text Mining for Biology and Biomedicine.* Norwood, MA: Artech House, 2006, pp. 13–41.

[Hahn et al. 2007b] Hahn, Udo; Wermter, Joachim; Blasczyk, Rainer; Horn, Peter A.: Text Mining: Powering the Database Revolution (Correspondence). In: *Nature* 448 (2007), No. 7150, pp. 130.

[Hajič et al. 2009] Hajič, Jan; Ciaramita, Massimiliano; Johansson, Richard; Kawahara, Daisuke; Martí, Maria A.; Màrquez, Lluís; Meyers, Adam; Nivre, Joakim; Padó, Sebastian; Štěpánek, Jan; Straňák, Pavel; Surdeanu, Mihai; Xue, Nianwen; Zhang, Yi: The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared*

Task. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 1–18. – URL http://www.aclweb.org/anthology/W09-1201.

[Hajǐc 1998] Hajǐc, Jan: Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In: *Issues of Valency and Meaning.* Prague/Praha: Karolinum, 1998, pp. 106–132.

[Hakenberg et al. 2005] Hakenberg, Jörg; Leser, Ulf; Plake, Conrad; Kirsch, Harald; Rebholz-Schuhmann, Dietrich: LLL'05 Challenge: Genic Interaction Extraction – Identification of Language Patterns Based on Alignment and Finite State Automata. In: *LLL-2005 – Proceedings of the 4th Learning Language in Logic Workshop in association with ICML 2005.* Bonn, Germany, August 2005, pp. 38–45.

[Hakenberg et al. 2008] Hakenberg, Jörg; Plake, Conrad; Royer, Loic; Strobelt, Hendrik; Leser, Ulf; Schroeder, Michael: Gene mention normalization and interaction extraction with context models and sentence motifs. In: *Genome Biology* 9(Sippl 2):S14 (2008).

[Harris 1991] Harris, Zellig: *A theory of language and infromation: a mathematical approach.* Oxford: Clarendon Press, 1991.

[Harris 2002] Harris, Zellig: The structure of Science Infromation. In: *Journal of Biomedical Informatics* 35 (2002), No. 4, pp. 215–221.

[Hearst 1992] Hearst, Marti A.: Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th conference on Computational Linguistics.* Morristown, NJ, USA: Association for Computational Linguistics, 1992, pp. 539–545.

[Hearst 1999] Hearst, Marti A.: Untangling text data mining. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics.* College Park, MD, USA, June 1999, pp. 3–10.

[Hendrickx et al. 2010] Hendrickx, Iris; Kim, Su N.; Kozareva, Zornitsa; Nakov, Preslav; Ó Séaghdha, Diarmuid; Padó, Sebastian; Pennacchiotti, Marco; Romano, Lorenza; Szpakowicz, Stan: SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In: *Proceedings of the 5th International Workshop on Semantic Evaluation.* Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 33–38. – URL http://www.aclweb.org/anthology/S10-1006.

[Higginbotham 1985] Higginbotham, James: On semantics. In: *Linguistic Inquiry* 16 (1985), No. 4, pp. 547–593.

[Hirschman et al. 2007] Hirschman, Lynette; Krallinger, Martin; Valencia, Alfonso (Editors.): *Proceedings of the 2nd* BioCreative *Challenge Evaluation Workshop.* Madrid: CNIO Centro Nacional de Investigaciones Oncológicas, 2007.

[Hirschman et al. 2005] Hirschman, Lynette; Yeh, Alexander S.; Blaschke, Christian; Valencia, Alfonso: Overview of BioCreative: Critical assessment of information extraction for biology. In: *BMC Bioinformatics* 6(Suppl 1):S1 (2005).

[Hoffmann and Valencia 2004] Hoffmann, Robert; Valencia, Alfonso: A Gene Network for Navigating the Literature. In: *Nature Genetics* 36 (2004), No. 7, pp. 664. – URL `http://www.ihop-net.org/`.

[Howe et al. 2008] Howe, Doug; Costanzo, Maria; Fey, Petra; Gojobori, Takashi; Hannick, Linda; Hide, Winston; Hill, David; Kania, Renate; Schaeffer, Mary; St Pierre, Susan; Twigger, Simon; White, Owen; Yon Rhee, Seung: Big data: The future of biocuration. In: *Nature* 455 (2008), September, No. 7209, pp. 47–50.

[Huang et al. 2010] Huang, Cuili; Wang, Yaqiang; Zhang, Yongmei; Jin, Yu; Yu, Zhonghua: Coreference resolution in biomedical full-text articles with domain dependent features. In: *ICCTD 2010 – Proceedings of the 2nd International Conference on Computer Technology and Development*, 2010, pp. 616–620.

[Huang et al. 2005] Huang, Minlie; Zhu, Xiaoyan; Li, Ming: A Hybrid Method for Relation Extraction from Biomedical Literature. In: *International Journal of Medical Informatics* 75 (2005), August, No. 6, pp. 443–455. – ISSN 1386-5056.

[Huang et al. 2004] Huang, Minlie; Zhu, Xiaoyan; Payan, Donald G.; Qu, Kunbin; Li, Ming: Discovering patterns to extract protein-protein interactions from full texts. In: *Bioinformatics* 20 (2004), No. 18, pp. 3604–3612.

[Hunter and Cohen 2006] Hunter, Lawrence; Cohen, Bretonnel K.: Biomedical Language Processing: What's beyond PubMed? In: *Mol Cell* 21 (2006), March, No. 5, pp. 589–594.

[International Human Genome Sequencing Consortium 2001] International Human Genome Sequencing Consortium: Initial sequencing and analysis of the human genome. In: *Nature* 409 (2001), No. 6822, pp. 860–921.

[Jackendoff 1972] Jackendoff, Ray: *Semantic Interpretation in Generative Grammar.* Cambridge, MA: MIT Press, 1972.

[Jackendoff 1990] Jackendoff, Ray: *Semantic Structures.* Cambridge, MA: MIT Press, 1990.

[Jenssen et al. 2001] Jenssen, Tor-Kristian; Lægreid, Astrid; Komorowski, Jan; Hovig, Eivind: A literature network of human genes for high-throughput analysis of gene expression. In: *Nature Genetics* 28 (2001), No. 1, pp. 21–28.

[Jiang and Zhai 2007] Jiang, Jing; Zhai, ChengXiang: A Systematic Exploration of the Feature Space for Relation Extraction. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 113–120. – URL `http://www.aclweb.org/anthology/N/N07/N07-1015`.

[Johansson and Nugues 2007a] Johansson, Richard; Nugues, Pierre: Extended Constituent-to-dependency Conversion for English. In: *NODALIDA 2007 – Proceedings of the 16th Nordic Conference of Computational Linguistics*. Tartu, Estonia, May 2007, pp. 105–112.

[Johansson and Nugues 2007b] Johansson, Richard; Nugues, Pierre: LTH: Semantic Structure Extraction using Nonprojective Dependency Trees. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 227–230. – URL `http://www.aclweb.org/anthology/S/S07/S07-1048`.

[Jonnalagadda et al. 2009] Jonnalagadda, Siddhartha; Tari, Luis; Hakenberg, Jörg; Baral, Chitta; Gonzalez, Graciela: Towards Effective Sentence Simplification for Automatic Processing of Biomedical Text. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 177–180. – URL `http://www.aclweb.org/anthology/N/N09/N09-2045`.

[Jurafsky and Martin 2009] Jurafsky, Daniel; Martin, James H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. 2nd Edition. Prentice Hall, 2009.

[Kambhatla 2004] Kambhatla, Nanda: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 2004, pp. 22.

[Kang et al. 2010] Kang, Ning; Mulligen, Erik M. van; Kors, Jan: Comparing and combining chunkers of biomedical text. In: *Journal of Biomedical Informatics* 44 (2010), No. 2, pp. 354–360.

[Kano et al. 2011] Kano, Yoshinobu; Bjorne, Jari; Ginter, Filip; Salakoski, Tapio; Buyko, Ekaterina; Hahn, Udo; K Bretonnel, Cohen; Verspoor, Karin; Roeder, Christophe; Lawrence E, Hunter; Kilicoglu, Halil; Bergler, Sabine; Landeghem, Sofie V.; Parys, Thomas V.; Peer, Yves VAN D.; Miwa, Makoto; Ananiadou, Sophia; Neves, Mariana; Pascual-Montano, Alberto; Ozgur, Arzucan; Radev, Dragomir R.; Riedel, Sebastian; Saetre, Rune; Chun, Hong-Woo; Kim, Jin-Dong; Pyysalo, Sampo; Ohta, Tomoko; Tsujii, Jun'ichi: U-Compare bio-event meta-service: compatible BioNLP event extraction services. In: *BMC Bioinformatics* 12:481 (2011).

[Karamanis et al. 2008] Karamanis, Nikiforos; Seal, Ruth; Lewin, Ian; McQuilton, Peter; Vlachos, Andreas; Gasperin, Caroline; Drysdale, Rachel; Briscoe, Ted: Natural Language Processing in aid of FlyBase curators. In: *BMC Bioinformatics* 9:193 (2008).

[Karlsson et al. 1995] Karlsson, Fred; Voutilainen, Atro; Heikkil¨a, Juha; Antilla, Arto: *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text.* Berlin, New York: Mouton de Gruyter, 1995.

[Katrenko and Adriaans 2006] Katrenko, Sophia; Adriaans, Pieter W.: Learning relations from biomedical corpora using dependency trees. In: Tuyls, Karl; Westra, Ronald L.; Saeys, Yvan; Nowé, Ann (Editors.): *KDECB 2006 – Knowledge Discovery and Emergent Complexity in Bioinformatics. Revised Selected Papers of the 1st International Workshop.* Vol. 4366, Ghent, Belgium, May 10, 2006. Berlin: Springer, 2006, pp. 61–80.

[Kilicoglu and Bergler 2009] Kilicoglu, Halil; Bergler, Sabine: Syntactic Dependency Based Heuristics for Biological Event Extraction. In: *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 119–127. – URL http://www.aclweb.org/anthology/W09-1418.

[Kim 1976] Kim, Jaegwon: In: Brand, Myles; Walton, Douglas (Editors.): *Action Theory.* Dordrecht:Reidel, 1976, Chap. Events as Property Exemplifications, pp. 159–177.

[Kim et al. 2009] Kim, Jin-Dong; Ohta, Tomoko; Pyysalo, Sampo; Kano, Yoshinobu; Tsujii, Jun'ichi: Overview of BioNLP'09 Shared Task on Event Extraction. In: *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 1–9. – URL http://www.aclweb.org/anthology/W09-1401.

[Kim et al. 2008a] Kim, Jin-Dong; Ohta, Tomoko; Tsujii, Jun'ichi: Corpus annotation for mining biomedical events from literature. In: *BMC Bioinformatics* 9:10 (2008).

[Kim et al. 2004] Kim, Jin-Dong; Ohta, Tomoko; Tsuruoka, Yoshimasa; Tateisi, Yuka; Collier, Nigel: Introduction to the bio-entity recognition task at JNLPBA. In: *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, pp. 70–75.

[Kim et al. 2011] Kim, Jin-Dong; Pyysalo, Sampo; Ohta, Tomoko; Bossy, Robert; Nguyen, Ngan; Tsujii, Jun'ichi: Overview of BioNLP Shared Task 2011. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 1–6. – URL `http://www.aclweb.org/anthology/W11-1801`.

[Kim and Rebholz-Schuhmann 2010] Kim, Jung-jae; Rebholz-Schuhmann, Dietrich: Improving the extraction of complex regulatory events from scientific text by using ontology-based inference. In: *Journal of Biomedical Semantics* 2(Suppl 5):S3 (2010).

[Kim et al. 2008b] Kim, Seon-Ho; Yoon, Juntae; Yang, Jihoon: Kernel approaches for genic interaction extraction. In: *Bioinformatics* 24 (2008), No. 1, pp. 118–126.

[Kipper et al. 2000] Kipper, Karin; Dang, Hoa T.; Palmer, Martha S.: Class-Based Construction of a Verb Lexicon. In: *AAAI/IAAI 2000 – Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*. Austin, TX, USA: AAAI Press / The MIT Press, 2000, pp. 691–696. – ISBN 0-262-51112-6.

[Klein et al. 2001] Klein, T. E.; Chang, J. T.; Cho, M. K.; Easton, K. L.; Fergerson, R.; Hewett, M.; Lin, Z.; Liu, Y.; Liu, S.; Oliver, D. E.; Rubin, D. L.; Shafa, F.; Stuart, J. M.; Altman, R. B.: Integrating genotype and phenotype information: An overview of the PharmGKB project. In: *Pharmacogenomics Journal* 1 (2001), No. 3, pp. 167–170.

[Knight and Marcu 2002] Knight, Kevin; Marcu, Daniel: Summarization beyond sentence extraction: A probabilistic approach to sentence compression. In: *Artificial Intelligence* 139 (2002), No. 1, pp. 91–107.

[Kohonen 2000] Kohonen, Teuvo: *Self-Organizing Maps*. Springer, December 2000.

[Koomen et al. 2005] Koomen, Peter; Punyakanok, Vasin; Roth, Dan; Yih, Wen-tau: Generalized Inference with Multiple Semantic Role Labeling Systems. In: *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005).* Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 181–184. – URL `http://www.aclweb.org/anthology/W/W05/W05-0625`.

[Kratzer 1995] Kratzer, Angelika: Stage-level and individual-level predicates. In: Carlson, Gregory N.; Pelletier, Francis J. (Editors.): *The Generic Book.* Chicago, London: University of Chicago Press, 1995, pp. 125–175.

[Krötzsch et al. 2009] Krötzsch, Markus; Patel-Schneider, Peter F.; Rudolph, Sebastian; Hitzler, Pascal; Parsia, Bijan: OWL 2 Web Ontology Language Primer / W3C. 2009. – Technical Report.

[Kudoh and Matsumoto 2000] Kudoh, Taku; Matsumoto, Yuji: Use of support vector learning for chunk identification. In: *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 142–144.

[Lafferty et al. 2001] Lafferty, John D.; McCallum, Andrew; Pereira, Fernando C. N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[Lease and Charniak 2005] Lease, Matthew; Charniak, Eugene: Parsing Biomedical Literature. In: *IJCNLP-05 – Proceedings of the 2nd International Joint Conference on Natural Language Processing.* Jeju Island, Republic of Korea, October 2005, pp. 58–69.

[Leitner et al. 2010] Leitner, Florian; Mardis, Scott A.; Krallinger, Martin; Cesareni, Gianni; Hirschman, Lynette; Valencia, Alfonso: An Overview of BioCreative II.5. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 7 (2010), No. 3, pp. 385–399.

[Leroy et al. 2003] Leroy, Gondy; Chen, Hsinchun; Martinez, Jesse D.: A shallow parser based on closed-class words to capture relations in biomedical text. In: *Journal of Biomedical Informatics* 36 (2003), No. 3, pp. 145–158.

[Levi 1978] Levi, Judith N.: *The Syntax and Semantics of Complex Nominals.* New York: Academic Press, 1978.

[Levin 1993] Levin, Beth: *English Verb Classes and Alternations A Preliminary Investigation.* Chicago and London: University of Chicago Press, 1993.

[Lin 1998] Lin, Dekang: Dependency-based Evaluation of MiniPar. In: *Proceedings of the LREC'98 Workshop on the Evaluation of Parsing Systems.* Granada, Spain, May 1998, pp. 48–56.

[Linde et al. 2011] Linde, Jörg; Buyko, Ekaterina; Hahn, Udo; Guthke, Reinhard: Full-genomic network inference for non-model organisms: A case study for the fungal pathogen Candida albicans. In: *Proceedings of the World Academy of Science, Engineering and Technology (WASET) International Conference on Bioinformatics Computational Biology and Biomedical Engineering* (2011), pp. 224–228.

[Llorens et al. 2010] Llorens, Hector; Saquete, Estela; Navarro, Borja: TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In: *Proceedings of the 5th International Workshop on Semantic Evaluation.* Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 284–291. – URL `http://www.aclweb.org/anthology/S10-1063`.

[Lu and Wilbur 2010] Lu, Zhiyong; Wilbur, W. J.: Overview of BioCreative III Gene Normalization. In: *Proceedings of BioCreative III workshop*, 2010, pp. 24–38.

[Maienborn 2011] Maienborn, Claudia: Event Semantics. In: Maienborn, Claudia; Heusinger, Klaus von; Portner, Paul (Editors.): *Semantics. An international handbook of natural language meaning* Vol. 1. Berlin, New York: Mouton de Gruyter, 2011.

[Manning and Schütze 1999] Manning, Chrsitopher D.; Schütze, Hinrich: *Foundations of statistical natural language processing.* Cambridge, MA: The MIT Press, 1999.

[Marcus et al. 1994] Marcus, Mitchell P.; Santorini, Beatrice; Marcinkiewicz, Mary A.: Building a Large Annotated Corpus of English: The Penn Treebank. In: *Computational Linguistics* 19 (1994).

[Marcus et al. 1993] Marcus, Mitchell P.; Santorini, Beatrice; Marcinkiewicz, Mary A.: Building a large annotated corpus of English: The Penn Treebank. In: *Computational Linguistics* 19 (1993), No. 2, pp. 313–330.

[Marsh and Perzanowski 1998] Marsh, Elaine; Perzanowski, Dennis: MUC-7 evaluation of IE technology: Overview of results. In: *MUC-7 – Proceedings of the Seventh Message Understanding Conference.* Fairfax, Virginia, USA, April 1998.

[Martins and Smith 2009] Martins, André F.; Smith, Noah A.: Summarization with a joint model for sentence extraction and summarization. In: *ILP for NLP – Proceedings of the NAACL/HLT 2009 Workshop on Integer Linear Programming for Natural Language Processing.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 1–9. – URL `http://www.aclweb.org/anthology/W09-1801`.

[Mathivanan et al. 2006] Mathivanan, Suresh; Periaswamy, Balamurugan; Gandhi, T. K. B.; Kandasamy, Kumaran; Suresh, Shubha; Mohmood, Riaz; Ramachandra, Y. L.; Pandey, Akhilesh: An evaluation of human protein-protein interaction data in the public domain. In: *BMC Bioinformatics* 7(Suppl 5):S19 (2006).

[McCallum 2002] McCallum, Andrew K.: MALLET: A Machine Learning for Language Toolkit. URL `http://mallet.cs.umass.edu/`, 2002. – Technical Report.

[McClosky 2010] McClosky, David: *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*, Department of Computer Science, Brown University, Dissertation, 2010.

[McClosky and Charniak 2008] McClosky, David; Charniak, Eugene: Self-Training for Biomedical Parsing. In: *Proceedings ACL-08/HLT-08.* Columbus, Ohio, USA, June 2008, pp. 101–104.

[McClosky et al. 2010] McClosky, David; Charniak, Eugene; Johnson, Mark: Automatic Domain Adaptation for Parsing. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Los Angeles, California: Association for Computational Linguistics, June 2010, pp. 28–36. – URL `http://www.aclweb.org/anthology/N10-1004`.

[McClosky et al. 2011] McClosky, David; Surdeanu, Mihai; Manning, Christopher: Event Extraction as Dependency Parsing for BioNLP 2011. In: *Proceedings of BioNLP Shared Task 2011 Workshop.* Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 41–45. – URL `http://www.aclweb.org/anthology/W11-1806`.

[McDonald 2006] McDonald, Ryan T.: Discriminative Sentence Compression with Soft Syntactic Evidence. In: *EACL'06 – Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics.* Trento, Italy, April 2006, pp. 297–304.

[McDonald et al. 2005] McDonald, Ryan T.; Pereira, Fernando; Ribarov, Kiril; Hajic, Jan: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: *HLT/EMNLP 2005 – Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, October 2005, pp. 523–530. – URL `http://www.aclweb.org/anthology/H/H05/H05-1066`.

[Meyers et al. 2004] Meyers, Adam; Reeves, Ruth; Macleod, Catherine; Szekely, Rachel; Zielinska, Veronika; Young, Brian; Grishman, Ralph: The NomBank Project: An Interim Report. In: Meyers, A. (Editors.): *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2004, pp. 24–31.

[Miller et al. 2007] Miller, John E.; Torii, Manabu; Vijay-Shanker, K.: Adaptation of POS Tagging for Multiple BioMedical Domains. In: *Biological, translational, and clinical language processing*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 179–180. – URL `http://www.aclweb.org/anthology/W/W07/W07-1024`.

[Mintz et al. 2009] Mintz, Mike; Bills, Steven; Snow, Rion; Jurafsky, Daniel: Distant supervision for relation extraction without labeled data. In: *Proceedings of the 2009 Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 1003–1011.

[Mitchell 1997] Mitchell, Tom M.; Munson, Eric M. (Editors.): *Machine Learning*. Boston, MA: WCB/McGraw-Hill, 1997 (McGraw-Hill Series in Computer Science). – 414 S.

[Miwa et al. 2010] Miwa, Makoto; Sætre, Rune; Kim, Jin-Dong; Tsujii, Jun'ichi: Event Extraction with complex event classification using rich features. In: *Journal of Bioinformatics and Computational Biology* 8 (2010), No. 1, pp. 131–146.

[Miyao et al. 2008] Miyao, Yusuke; Sætre, Rune; Sagae, Kenji; Matsuzaki, Takuya; Tsujii, Jun'ichi: Task-oriented Evaluation of Syntactic Parsers and Their Representations. In: *ACL 2008 – Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 46–54. – URL `http://www.aclweb.org/anthology/P/P08/P08-1006`.

[Miyao and Tsujii 2002] Miyao, Yusuke; Tsujii, Jun'ichi: Maximum entropy estimation for feature forests. In: *HLT 2002 – Proceedings of the 2nd International*

Conference on Human Language Technology Research. San Diego, CA, USA, March 2002, pp. 292–297.

[Montague 1969] Montague, Richard: On the Nature of Certain Philosophical Entities. In: *The Monist* 53 (1969), No. 2, pp. 159–194.

[Morgan et al. 2008] Morgan, Alexander A.; Lu, Zhiyong; Wang, Xinglong; Cohen, Aaron; Fluck, Juliane; Ruch, Patrick; Divoli, Anna; Fundel, Katrin; Leaman, Robert; Hakenberg, Jörg; Sun, Chengjie; Liu, Heng-hui; Torres, Rafael; Krauthammer, Michael; Lau, William W.; Liu, Hongfang; Hsu, Chun-Nan; Schuemie, Martijn; Cohen, Kevin B.; Hirschman, Lynette: Overview of BioCreative II gene normalization. In: *Genome Biology* 9(Suppl 2):S3 (2008).

[Müller et al. 2004] Müller, Hans-Michael; Kenny, Eimear E.; Sternberg, Paul W.: Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. In: *PLoS Biology* 2 (2004), November, No. 11, pp. 1984–1998.

[Naur et al. 1960] Naur, P.; Backus, J. W.; Bauer, F. L.; Green, J.; Katz, C.; McCarthy, J.; Perlis, A. J.; Rutishauser, H.; Samelson, K.; Vauquois, B.; Wegstein, J. H.; Wijnagaarden, A. van; Woodger, M.: Report on the Algorithmic Language ALGOL 60. In: *Communications of the ACM* 3 (1960), No. 5, pp. 299–314. – Revised in CACM 6:1, 1-17, 1963.

[Nédellec 2005] Nédellec, Claire: Learning Language in Logic: Genic interaction extraction challenge. In: *Proceedings LLL-2005 – 4th Learning Language in Logic Workshop*, Bonn, Germany, August 2005, pp. 31–37.

[Nivre et al. 2007] Nivre, Joakim; Hall, Johan; Nilsson, Jens: MaltParser: A language-independent system for data-driven dependency parsing. In: *Natural Language Engineering* 13 (2007), No. 2, pp. 95–135.

[Nivre et al. 2010] Nivre, Joakim; Rimell, Laura; McDonald, Ryan; Gómez Rodríguez, Carlos: Evaluation of Dependency Parsers on Unbounded Dependencies. In: *COLING 2010 – Proceedings of the 23rd International Conference on Computational Linguistics.* Beijing, China: Coling 2010 Organizing Committee, August 2010, pp. 833–841. – URL `http://www.aclweb.org/anthology/C10-1094`.

[Noy et al. 2009] Noy, Natalya F.; Shah, Nigam H.; Whetzel, Patricia L.; Dai, Benjamin; Dorf, Michael; Griffith, Nicholas; Jonquet, Clement; Rubin, Daniel L.; Storey, Margaret-Anne D.; Chute, Christopher G.; Musen, Mark A.: BioPortal: ontologies and integrated data resources at the click of a mouse. In: *Nucleic Acids Research* 37(Web-Server-Issue) (2009), pp. 170–173.

[Oda et al. 2008] Oda, Kanae; Kim, Jin-Dong; Ohta, Tomoko; Okanohara, Daisuke; Matsuzaki, Takuya; Tateisi, Yuka; Tsujii, Jun'ichi: New challenges for text mining: Mapping between text and manually curated pathways. In: *BMC Bioinformatics* 9(Suppl 3):S5 (2008).

[Odds 1988] Odds, F C.; Tindall, Baillière (Editors.): *Candida and Candidosis.* 2nd. London: W.B. Saunders Company, 1988.

[Ohta et al. 2002] Ohta, Tomoko; Tateisi, Yuka; Kim, Jin-Dong: The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In: Marcus, Mitchell P. (Editors.): *HLT 2002 – Human Language Technology Conference. Proceedings of the 2nd International Conference on Human Language Technology Research.* San Diego, Cal., USA: San Francisco, CA: Morgan Kaufmann, March 2002, pp. 82–86.

[Palmer et al. 2007] Palmer, Alexis; Ponvert, Elias; Baldridge, Jason; Smith, Carlota: A Sequencing Model for Situation Entity Classification. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.* Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 896–903. – URL `http://www.aclweb.org/anthology/P07-1113`.

[Palmer et al. 2005] Palmer, Martha S.; Gildea, Daniel; Kingsbury, Paul: The Proposition Bank: An annotated corpus of semantic roles. In: *Computational Linguistics* 31 (2005), No. 1, pp. 71–106.

[Park 2001] Park, Jong C.: Using Combinatory Categorial Grammar to Extract Biomedical Information. In: *IEEE Intelligent Systems* 16 (2001), No. 6, pp. 62–67.

[Pollard and Sag 1994] Pollard, Carl J.; Sag, Ivan A.: *Head-driven Phrase Structure Grammar.* Chicago, IL: University of Chicago Press, 1994.

[Porter 1980] Porter, M. F.: An algorithm for suffix stripping. In: *Program* 14 (1980), No. 3, pp. 130–137.

[Pustejovsky et al. 2003] Pustejovsky, James; Hanks, Patrick; Saurí, Roser; See, Andrew; Gaizauskas, Robert; Setzer, Andrea; Radev, Dragomir; Sundheim, Beth; Day, David; Ferro, Lisa; Lazo, Marcia: The TimeBank corpus. In: Archer, Dawn; Rayson, Paul; Wilson, Andrew; McEnery, Tony (Editors.): *Proceedings of the Corpus Linguistics 2003 Conference*, Lancaster University, U.K., 2003 (UCREL Technical Paper 16), pp. 647–656.

[Pustejovsky and Verhagen 2009] Pustejovsky, James; Verhagen, Marc: SemEval-2010 Task 13: Evaluating Events, Time Expressions, and Temporal

Relations (TempEval-2). In: *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 112–116. – URL `http://www.aclweb.org/anthology/W09-2418`.

[Pyysalo et al. 2008] Pyysalo, Sampo; Airola, Antti; Heimonen, Juho; Björne, Jari; Ginter, Filip; Salakoski, Tapio: Comparative analysis of five protein-protein interaction corpora. In: *BMC Bioinformatics* 9(Suppl 3):S6 (2008).

[Pyysalo et al. 2007] Pyysalo, Sampo; Ginter, Filip; Heimonen, Juho; Björne, Jari; Boberg, Jorma; Jarvinen, Jouni; Salakoski, Tapio: BioInfer: A corpus for information extraction in the biomedical domain. In: *BMC Bioinformatics* 8:50 (2007).

[Pyysalo et al. 2006a] Pyysalo, Sampo; Ginter, Filip; Pahikkala, Tapio; Boberg, Jorma; Jarvinen, Jouni; Salakoski, Tapio: Evaluation of two dependency parsers on biomedical corpus targeted at protein-protein interactions. In: *International Journal of Medical Informatics* 75 (2006), June, No. 6, pp. 430–442.

[Pyysalo et al. 2009] Pyysalo, Sampo; Ohta, Tomoko; Kim, Jin-Dong; Tsujii, Jun'ichi: Static Relations: a Piece in the Biomedical Information Extraction Puzzle. In: *Proceedings of the BioNLP 2009 Workshop*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 1–9. – URL `http://www.aclweb.org/anthology/W09-1301`.

[Pyysalo et al. 2006b] Pyysalo, Sampo; Salakoski, Tapio; Aubin, Sophie; Nazarenko, Adeline: Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. In: *Proceedings of Second International Symposium on Semantic Mining in Biomedicine (SMBM)* Vol. 7, November 2006, pp. S2.

[Qian et al. 2008] Qian, Longhua; Zhou, GuoDong; Kong, Fang; Zhu, QiaoMing; Qian, Peide: Exploiting Constituent Dependencies for Tree Kernel-Based Semantic Relation Extraction. In: *COLING 2008 – Proceedings of the 22nd International Conference on Computational Linguistics*. Manchester, UK, August 2008, pp. 697–704. – URL `http://www.aclweb.org/anthology/C08-1088`.

[Quine 1960] Quine, Willard V.: *Word and Object*. Cambridge, MA: MIT Press, 1960.

[Ramani et al. 2005] Ramani, Arun K.; Bunescu, Razvan; Mooney, Raymond J.; Marcotte, Edward M.: Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. In: *Genome Biology* 6:R40 (2005).

[Ratnaparkhi 1996] Ratnaparkhi, Adwait: A Maximum Entropy Model for Part-of-Speech Tagging. In: Brill, Eric; Church, Kenneth (Editors.): *EMNLP'96 – Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing*. Philadelphia, PA: Association for Computational Linguistics, May 1996, pp. 133–142.

[Ratnaparkni 1998] Ratnaparkni, Adwait: *Maximum Entropy Models for Natural Language Ambiguity Resolution*, University of Pennsylvania, Dissertation, 1998.

[Riedel and McCallum 2011] Riedel, Sebastian; McCallum, Andrew: Robust Biomedical Event Extraction with Dual Decomposition and Minimal Domain Adaptation. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 46–50. – URL http://www.aclweb.org/anthology/W11-1807.

[Riedel et al. 2011] Riedel, Sebastian; McClosky, David; Surdeanu, Mihai; McCallum, Andrew; D. Manning, Christopher: Model Combination for Event Extraction in BioNLP 2011. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 51–55. – URL http://www.aclweb.org/anthology/W11-1808.

[Rimell and Clark 2009] Rimell, Laura; Clark, Stephen: Porting a lexicalized-grammar parser to the biomedical domain. In: *Journal of Biomedical Informatics* 42 (2009), No. 5, pp. 852–865.

[Rimell et al. 2009] Rimell, Laura; Clark, Stephen; Steedman, Mark: Unbounded Dependency Recovery for Parser Evaluation. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, August 2009, pp. 813–821. – URL http://www.aclweb.org/anthology/D/D09/D09-1085.

[Rodríguez-Penagos et al. 2007] Rodríguez-Penagos, Carlos; Salgado, Heladia; Martínez-Flores, Irma; Collado-Vides, Julio: Automatic reconstruction of a bacterial regulatory network using Natural Language Processing. In: *BMC Bioinformatics* 8:293 (2007).

[Rosario and Hearst 2001] Rosario, Barbara; Hearst, Marti A.: Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy.

In: *EMNLP'01 – Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, URL http://www.aclweb.org/anthology-new/W/W01/W01-0511, 2001, pp. 82–90.

[Sætre et al. 2007] Sætre, Rune; Sagae, Kenji; Tsujii, Jun'ichi: Syntactic Features for Protein-Protein Interaction Extraction. In: *LBM 2007 – Short Paper Proceedings of the 2nd International Symposium on Languages in Biology and Medicine.* Singapore, December 2007, pp. 6.1–6.14.

[Sagae and Tsujii 2007] Sagae, Kenji; Tsujii, Jun'ichi: Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In: *EMNLP-CoNLL 2007 – Proceedings of the Conference on Empirical Methods in Natural Language Processing and the Conference on Computational Natural Language Learning.* Prague, Czech Republic, June 2007, pp. 1044–1050.

[Sager et al. 1987] Sager, Naomi; Friedman, Carol; Lyman, Margaret S.: *Medical Language Processing: Computer Management of Narrative Data.* Addison-Wesley, Reading, MA, 1987.

[Sampson 1994] Sampson, Geoffrey: *English for the Computer.* Oxford University Press, 1994.

[Sáric et al. 2004] Sáric, Jasmin; Jensen, Lars J.; Bork, Peer; Ouzounova, Rossitza; Rojas, Isabel: Extracting Regulatory Gene Expression Networks From Pubmed. In: *ACL '04 – Proceedings of the 42nd Meeting of the Association for Computational Linguistics.* Barcelona, Spain, July 2004, pp. 191–198. – URL http://www.aclweb.org/anthology/P04-1025.

[Sasaki et al. 2008a] Sasaki, Yutaka; Montemagni, Simonetta; Pezik, Piotr; Rebholz-Schuhmann, Dietrich; McNaught, John; Ananiadou, Sophia: BioLexicon: A Lexical Resource for the Biology Domain. In: *SMBM 2008 – Proceedings of the 3rd International Symposium on Semantic Mining in Biomedicine*, Turku, Finland, 2008, pp. 109–116.

[Sasaki et al. 2008b] Sasaki, Yutaka; Thompson, Paul; Cotter, Philip; McNaught, John; Ananiadou, Sophia: Event Frame Extraction Based on a Gene Regulation Corpus. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008).* Manchester, UK: Coling 2008 Organizing Committee, August 2008, pp. 761–768. – URL http://www.aclweb.org/anthology/C08-1096.

[Schölkopf and Smola 2001] Schölkopf, Bernhard; Smola, Alexander J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and*

Beyond (Adaptive Computation and Machine Learning). Cambridge, MA: The MIT Press, 12 2001.

[Sekine 1997] SEKINE, Satoshi: The Domain Dependence of Parsing. In: *Proceedings of the Fifth Conference on Applied Natural Language Processing.* Washington, DC, USA: Association for Computational Linguistics, March 1997, pp. 96–102. – URL `http://www.aclweb.org/anthology/A97-1015`.

[Seringhaus and Gerstein 2007] SERINGHAUS, Michael; GERSTEIN, Mark: Publishing perishing? Towards tomorrow's information architecture. In: *BMC Bioinformatics* 8:17 (2007).

[Sha and Pereira 2003] SHA, Fei; PEREIRA, Fernando C. N.: Shallow parsing with conditional random fields. In: *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.* Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 134–141. – URL `http://aclweb.org/anthology-new/N/N03/N03-1028`.

[Siegel and McKeown 1996] SIEGEL, Eric V.; MCKEOWN, Kathleen R.: Gathering statistics to aspectually classify sentences with a genetic algorithm. In: *Proceedings of the Second International Conference on New Methods in Language Processing.* Ankara, Turkey, September 1996.

[Skut et al. 1997] SKUT, Wojciech; KRENN, Brigitte; BRANTS, Thorsten; USZKOREIT, Hans: An Annotation Scheme for Free Word Order Languages. In: *Proceedings of the Fifth Conference on Applied Natural Language Processing.* Washington, DC, USA: Association for Computational Linguistics, March 1997, pp. 88–95. – URL `http://www.aclweb.org/anthology/A97-1014`.

[Sleator and Temperley 1991a] SLEATOR, Daniel; TEMPERLEY, Davy: Parsing English with a Link Grammar / Carnegie-Mellon University. URL `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/link/pub/www/papers/ps/tr91-196.pdf`, 10 1991 (CMU-CS-91-196). – Technical Report.

[Sleator and Temperley 1991b] SLEATOR, Daniel; TEMPERLEY, Davy: Parsing English with a link grammar / Department of Computer Science, CMU. 1991. – Technical Report.

[Sleator and Temperley 1993] SLEATOR, Daniel D.; TEMPERLEY, Davy: Parsing English with a Link Grammar. In: *Proceedings of the 3rd International Workshop on Parsing Technologies*, 1993, pp. 277–292.

[Smith et al. 2005] Smith, Barry; Ceusters, Werner; Klagges, Bert; Köhler, Jacob; Kumar, Anand; Lomax, Jane; Mungall, Chris; Neuhaus, Fabian; Rector, Alan L.; Rosse, Cornelius: Relations in biomedical ontologies. In: *Genome Biology* 6:R46 (2005).

[Smith et al. 2004] Smith, Lawrence H.; Rindflesch, Thomas; Wilbur, W. J.: MedPost: A part-of-speech tagger for bioMedical text (Applications Note). In: *Bioinformatics* 20 (2004), No. 14, pp. 2320–2321.

[Snow et al. 2005] Snow, Rion; Jurafsky, Daniel; Ng, Andrew Y.: Learning Syntactic Patterns for Automatic Hypernym Discovery. In: Saul, Lawrence K.; Weiss, Yair; Bottou, Leon (Editors.): *Advances in Neural Information Processing Systems (NIPS 2004).* Cambridge, MA: MIT Press, 2005, pp. 1297–1304.

[Snow et al. 2006] Snow, Rion; Jurafsky, Daniel; Ng, Andrew Y.: Semantic taxonomy induction from heterogenous evidence. In: *ACL'06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* Morristown, NJ, USA: Association for Computational Linguistics, July 2006, pp. 801–808. – URL `http://www.aclweb.org/anthology/P06-1101`.

[Steedman 1987] Steedman, Mark J.: Combinatory grammars and human language processing. In: *Modularity in Knowledge Representation and Natural-Language Understanding* (1987), pp. 187–205.

[Steedman 1996] Steedman, Mark J.: Surface Structure and Interpretation. (1996).

[Stevenson and Greenwood 2009] Stevenson, Mark; Greenwood, Mark A.: Dependency Pattern Models for Information Extraction. In: *Research on Language and Computation* 7 (2009), No. 1, pp. 13–39.

[Sun et al. 2007] Sun, Chengjie; Guan, Yi; Wang, Xiaolong; Lin, Lei: Rich features based Conditional Random Fields for biological named entities recognition. In: *Computers in Biology and Medicine* 37 (2007), No. 9, pp. 1327–1333.

[Sundheim 1995] Sundheim, Beth: Overview of Results of the MUC-6 evaluation. In: *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Columbia, MD, 1995, pp. 13–31.

[Surdeanu et al. 2008] Surdeanu, Mihai; Johansson, Richard; Meyers, Adam; Màrquez, Lluís; Nivre, Joakim: The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In: *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning.* Manchester, England: Coling 2008 Organizing Committee, August 2008, pp. 159–177. – URL http://www.aclweb.org/anthology/W08-2121.

[Swanson 1986] SWANSON, Don R.: Fish oil, Raynaud's syndrome, and undiscovered public knowledge. In: *Perspectives in Biology and Medicine* 20 (1986), No. 1, pp. 7–18.

[Swanson 1988] SWANSON, Don R.: Migraine and magnesium: Eleven neglected connections. In: *Perspectives in Biology and Medicine* 31 (1988), No. 4, pp. 526–557.

[Szolovitz 2003] SZOLOVITZ, Peter: Adding a medical lexicon to an English parser. In: *Proceeedings of the 2003 AMIA Annual Symposium.* Bethesda, MD: Americal Medical Informatics Association, 2003, pp. 639–643.

[Tam et al. 2008] TAM, Wai L.; SATO, Yo; MIYAO, Yusuke; TSUJII, Junichi: Parser Evaluation Across Frameworks without Format Conversion. In: *COLING 2008 – Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation.* Manchester, UK: Coling 2008 Organizing Committee, August 2008, pp. 29–35. – URL http://www.aclweb.org/anthology/W08-1305.

[Tapanainen and Jarvinen 1997] TAPANAINEN, Pasi; JARVINEN, Timo: A non-projective dependency parser. In: *Proceedings of the Fifth Conference on Applied Natural Language Processing.* Washington, DC, USA: Association for Computational Linguistics, March 1997, pp. 64–71. – URL http://www.aclweb.org/anthology/A97-1011.

[Tateisi et al. 2005] TATEISI, Yuka; YAKUSHIJI, Akane; TSUJII, Jun'ichi: Syntax Annotation for the GENIA corpus. In: *IJCNLP 2005 – Proceedings of the 2nd International Joint Conference on Natural Language Processing.* Jeju Island, Korea, October 2005, pp. 222–227.

[Tesnière 1959] TESNIÈRE, Lucien: *Elements de syntaxe structurale.* Paris: Klincksieck, 1959.

[Thompson et al. 2009] THOMPSON, Paul; IQBA, Syed A.; MCNAUGHT, John; ANANIADOU, Sophia: Construction of an annotated corpus to support biomedical information extraction. In: *BMC Bioinformatics* 10:349 (2009).

[Tjong Kim Sang and Buchholz 2000] TJONG KIM SANG, Erik F.; BUCHHOLZ, Sabine: Introduction to the CoNLL-2000 Shared Task: Chunking. In: CARDIE, Claire; DAELEMANS, Walter; NÃ©DELLEC, Claire; TJONG KIM SANG, Erik F.

(Editors.): *Proceedings of the 4th Conference on Computational Language Learning (CoNLL-2000) and the 2nd Learning Language in Logic Workshop (LLL-2000).* Lisbon, Portugal: Association for Computational Linguistics, September 2000, pp. 127–132.

[Tomanek 2010] Tomanek, Katrin: *Resource-Aware Annotation through Active Learning*, Technische Universität Dortmund, phdDissertation, 2010.

[Tomanek and Hahn 2009] Tomanek, Katrin; Hahn, Udo: Semi-Supervised Active Learning for Sequence Labeling. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP.* Suntec, Singapore: Association for Computational Linguistics, August 2009, pp. 1039–1047. – URL `http://www.aclweb.org/anthology/P/P09/P09-1117`.

[Tomanek et al. 2007a] Tomanek, Katrin; Wermter, Joachim; Hahn, Udo: A reappraisal of sentence and token splitting for Life Sciences Documents. In: Kuhn, K. A.; Warren, J. R.; Leong, T. Y. (Editors.): *MEDINFO'07 – Proceedings of the 12th World Congress on Medical Informatics. Building Sustainable Health Systems*, IOS Press, 2007 (Studies in Health Technology and Informatics 129), pp. 524–528.

[Tomanek et al. 2007b] Tomanek, Katrin; Wermter, Joachim; Hahn, Udo: Sentence and Token Splitting Based on Conditional Random Fields. In: *PACLING 2007 – Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics.* Melbourne, Australia: Melbourne: Pacific Association for Computational Linguistics, September 2007, pp. 49–57.

[Tsai et al. 2007] Tsai, Richard; Chou, Wen-Chi; Su, Ying-Shan; Lin, Yu-Chun; Sung, Cheng-Lung; Dai, Hong-Jie; Yeh, Irene; Ku, Wei; Sung, Ting-Yi; Hsu, Wen-Lian: BIOSMILE: A semantic role labeling system for biomedical verbs using a maximum-entropy model with automatically generated template features. In: *BMC Bioinformatics* 8:325 (2007).

[Tsuruoka et al. 2007] Tsuruoka, Yoshimasa; McNaught, John; Tsujii, Jun'ichi; Ananiadou, Sophia: Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. In: *Bioinformatics* 23 (2007), No. 20, pp. 2768–2774.

[Tsuruoka et al. 2005] Tsuruoka, Yoshimasa; Tateishi, Yuka; Kim, Jin-Dong; Ohta, Tomoko; McNaught, John; Ananiadou, Sophia; Tsujii, Jun ichi: Developing a Robust Part-of-Speech Tagger for Biomedical Text. In: Bozanis, Panayiotis; Houstis, Elias N. (Editors.): *Advances in Informatics In Advances in Informatics* Vol. 3476, 2005, pp. 382–392.

[Turner and Charniak 2005] TURNER, Jenine; CHARNIAK, Eugene: Supervised and Unsupervised Learning for Sentence Compression. In: *ACL 2005 – Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, Michigan, June 25-30, 2005, June 2005, pp. 290–297. – URL `http://www.aclweb.org/anthology/P05-1036`.

[UniProt Consortium 2008] UNIPROT CONSORTIUM, The: The Universal Protein Resource (UniProt). In: *Nucleic Acids Research* 36 (2008), January, No. Database issue. – ISSN 1362-4962.

[Vanderwende et al. 2007] VANDERWENDE, Lucy; SUZUKI, Hisami; BROCKETT, Chris; NENKOVA, Ani: Beyond SUMBASIC: Task-focused summarization with sentence simplification and lexical expansion. In: *Information Processing and Management* 43 (2007), No. 6, pp. 1606–1618.

[Vendler 1967] VENDLER, Zeno: *Linguistics and Philosophy*. Cornell University Press, Ithaca NY, 1967.

[Venter et al. 2001] VENTER, Craig J. et al.: The Sequence of the Human Genome. In: *Science* 291 (2001), No. 5507, pp. 1304–1351.

[Verhagen et al. 2007] VERHAGEN, Marc; GAIZAUSKAS, Robert; SCHILDER, Frank; HEPPLE, Mark; KATZ, Graham; PUSTEJOVSKY, James: SemEval-2007 Task 15: TempEval Temporal Relation Identification. In: *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 75–80. – URL `http://www.aclweb.org/anthology/S/S07/S07-1014`.

[Vickrey and Koller 2008] VICKREY, David; KOLLER, Daphne: Sentence Simplification for Semantic Role Labeling. In: *ACL 2008: HLT – Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 344–352. – URL `http://www.aclweb.org/anthology/P/P08/P08-1040`.

[Vlachos 2009] VLACHOS, Andreas: *Semi-supervised learning for biomedical information extraction*, University of Cambridge, phdDissertation, 2009.

[Wattarujeekrit et al. 2004] WATTARUJEEKRIT, Tuangthong; SHAH, Parantu; COLLIER, Nigel: PASBio: predicate-argument structures for event extraction in molecular biology. In: *BMC Bioinformatics* 5:155 (2004).

[Wehrens and Buydens 2007] WEHRENS, Ron; BUYDENS, Lutgarde M. C.: Self- and Super-organising Maps in R: the Kohonen package. In: *Journal of Statistical Software* 21 (2007), No. 5, pp. 1–19.

[Wermter et al. 2005] Wermter, Joachim; Fluck, Juliane; Stroetgen, Jannik; GeiÃŸler, Stefan; Hahn, Udo: Recognizing noun phrases in biomedical text: An evaluation of lab prototypes and commercial chunkers. In: Hahn, Udo; Valencia, Alfonso (Editors.): *SMBM 2005 – Proceedings of the 1st International Symposium on Semantic Mining in Biomedicine.* Hinxton, England, U.K., April 2005, pp. 25–33.

[Wermter et al. 2009] Wermter, Joachim; Tomanek, Katrin; Hahn, Udo: High-Performance Gene Name Normalization with GeNo. In: *Bioinformatics* 25 (2009), No. 6, pp. 815–821.

[Wundt 1900] Wundt, Wilhelm: *Völkerpsychologie: eine Untersuchung der Entwicklungsgesetze von Sprache, Mythus und Sitte.* Band II: Die Sprache, Zweiter Teil. W. Engelmann, Leipzig, 1900.

[Xia and Palmer 2001] Xia, Fei; Palmer, Marta: Converting Dependency Structures to Phrase Structures. In: *Proceedings of the HLT'01 – First International Conference on Human Language Technology Research.* San Francisco: Association for Computational Linguistics, 2001. – URL `http://aclweb.org/anthology-new/H/H01/H01-1014.pdf`.

[Xuan et al. 2007] Xuan, Weijian; Watson, Stanley J.; Meng, Fan: Tagging Sentence Boundaries in Biomedical Literature. In: Gelbukh, Alexander F. (Editors.): *Proceedings of the Eighth International Conference on Intelligent Text Processing and Computational Linguistics – CICLing'07* Vol. 4394, Springer, 2007, pp. 186–195.

[Yakushiji et al. 2001] Yakushiji, Akane; Tateisi, Yuka; Miyao, Yusuke; Tsujii, Jun'ichi: Event extraction from biomedical papers using a full parser. In: Altman, Russ B.; Dunker, A. K.; Hunter, Lawrence; Lauderdale, Kevin; Klein, Teri E. (Editors.): *PSB 2001 – Proceedings of the 6th Pacific Symposium on Biocomputing.* Maui, Hawaii, USA, January 2001, pp. 408–419.

[Yamagata et al. 2006] Yamagata, Kiwamu; Fukutomi, Satoshi; Takagi, Kazuyuki; Ozeki, Kazuhiko: Sentence Compression Using Statistical Information About Dependency Path Length. In: Sojka, Petr; Kopecek, Ivan; Pala, Karel (Editors.): *Text, Speech and Dialogue. TSD 2006 – Proceedings of the 9th International Conference* Vol. 4188. Brno, Czech Republic, September 2006, pp. 127–134.

[Yang et al. 2008] Yang, Hui; Nenadic, Goran; Keane, John: Identification of transcription factor contexts in literature using machine learning approaches. In: *BMC Bioinformatics* 9(Suppl 3):S11 (2008).

[Yang et al. 2004] Yang, Xiaofeng; Su, Jian; Zhou, GuoDong; Tan, Chew L.: An NP-Cluster Based Approach to Coreference Resolution. In: *COLING Geneva 2004 – Proceedings of the 20th International Conference on Computational Linguistics* Vol. 2. Geneva, Switzerland: Association for Computational Linguistics, August 2004, pp. 226–232.

[Yeh et al. 2005] Yeh, Alexander; Morgan, Alexander; Colosimo, Marc; Hirschman, Lynette: BioCreAtIvE task 1A: gene mention finding evaluation. In: *BMC Bioinformatics* 6(Suppl 1):S2 (2005).

[Yousfi-Monod and Prince 2008] Yousfi-Monod, Mehdi; Prince, Violaine: Sentence Compression as a Step in Summarization or an Alternative Path in Text Shortening. In: *Coling 2008: Companion volume: Posters.* Manchester, UK: Coling 2008 Organizing Committee, August 2008, pp. 139–142. – URL `http://www.aclweb.org/anthology/C08-2035`.

[Zelenko et al. 2003] Zelenko, Dmitry; Aone, Ch.; Richardella, Anthony: Kernel Methods for Relation Extraction. In: *Journal of Machine Learning Research* 3 (2003), pp. 1083–1106.

[Zhao et al. 2009] Zhao, Hai; Chen, Wenliang; Kity, Chunyu; Zhou, Guodong: Multilingual Dependency Learning: A Huge Feature Engineering Method to Semantic Dependency Parsing. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task.* Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 55–60. – URL `http://www.aclweb.org/anthology/W09-1208`.

[Zhou and Su 2000] Zhou, GuoDong; Su, Jian: Error-driven HMM-based Chunk Tagger with Context-dependent Lexicon. In: *EMNLP/VLC'2000 – Proceedings of Joint Sigdat Confierence on Empirical Methods in Natural Langauge Processing and Very Large Corpora*, 2000, pp. 71–79.

[Zhou and Zhang 2007] Zhou, Guodong; Zhang, Min: Extracting relation information from text documents by exploring various types of knowledge. In: *Information Processing & Management* 43 (2007), No. 4, pp. 969–982. – ISSN 0306-4573.