Universidade de Aveiro
2013

Departamento de
Electrónica, Telecomunicações e Informática
Programa de Doutoramento em Informática das
Universidades do Minho, Aveiro e do Porto

David Emmanuel Marques Campos

Mineração de informação biomédica a partir de literatura científica

Mining biomedical information from scientific literature

Universidade de Aveiro
2013

Departamento de
Electrónica, Telecomunicações e Informática
Programa de Doutoramento em Informática das
Universidades do Minho, Aveiro e do Porto

David Emmanuel Marques Campos

Mineração de informação biomédica a partir de literatura científica

Mining biomedical information from scientific literature

Tese apresentada às Universidades do Minho, Aveiro e do Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor no âmbito do doutoramento conjunto MAP-i, realizada sob a orientação científica do Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Sérgio Guilherme Aleixo de Matos, Investigador Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

Doutor Amadeu Mortágua Velho da Maia Soares
Professor Catedrático do Departamento de Biologia da Universidade de Aveiro, Portugal

vogais / examiners committee

Doutor Fernando Manuel Augusto da Silva
Professor Catedrático do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto, Portugal

Doutor Francisco José Moreira Couto
Professor Associado do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, Portugal

Doutor Erik M. van Mulligen
Professor Assistente do Departamento de Informática Médica do Erasmus Medical Centre de Rotterdam, Holanda

Doutor Paulo Jorge Sousa Gomes
Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Portugal

Doutor Sérgio Guilherme Aleixo de Matos
Investigador Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (co-orientador), Portugal

agradecimentos / acknowledgements

Gostava de agradecer aos meus orientadores, José Luís Oliveira e Sérgio Matos, pela oportunidade, orientação e apoio incondicional durante o doutoramento. Acredito profundamente que as suas visões, recomendações, discussões e amizade foram um contributo fundamental para a execução deste trabalho. Também gostava de agradecer aos membros do Grupo de Bioinformática pelos seus conselhos e conversas divertidas, que o tornaram num local divertido para estar e trabalhar. Agradeço também aos membros dos grupos de investigação em que estive durante os estágios internacionais, pelo seu apoio e conselhos. Em particular, Dietrich Rebholz-Schuhmann pela orientação e conversas interessantes durante a minha estadia em Cambridge, e Erik van Mulligen e Jan Kors pelas recomendações e interessantes trocas de ideias enquanto estive em Roterdão. Também quero agradecer aos meus amigos, Luis Ribeiro, Pedro Alves e David Ferreira, pelas conversas idiotas e rejuvenescedoras sobre o trabalho e a vida, acompanhadas de umas igualmente refrescantes cervejas. Finalmente, um agradecimento especial à minha família, aos meus pais Luís e Graça e às minhas irmãs Jenny e Isis, não só pelo apoio genuíno e ilimitado durante este período, mas também pelas bases sólidas que tornaram isto possível. Por último, mas não menos importante, quero agradecer à minha namorada Raquel pela amizade, apoio e paciência infinita.

I would like to thank my supervisors, José Luís Oliveira and Sérgio Matos, for the opportunity, guidance and unconditional support through the PhD. I truly believe that their vision, recommendations, discussions and friendship were an important contribution to the execution of this work. I also like to thank the members of the Bioinformatics Group for their advices and entertaining conversations, which made it a fun place to be and work. I am also grateful to the members of the research groups that I have been involved with during the international internships, for their support and advices. In particular, Dietrich Rebholz-Schuhmann for the supervision and interesting conversations during my stay in Cambridge, and Erik van Mulligen and Jan Kors for their recommendations and engaging exchange of ideas while in Rotterdam. I also want to thank my friends, Luis Ribeiro, Pedro Alves and David Ferreira, for the idiotic and refreshing talks about work and life, accompanied by some equally refreshing beers. Finally, a special thanks to my family, my parents Luís and Graça and my sisters Jenny and Isis, not only for their genuine and unlimited support during this period, but also for the solid foundations that made this possible. Last but not least, I want to thank my girlfriend Raquel for the friendship, support and endless patience.

Palavras-chave

Bioinformática, mineração de texto, extracção de informação, reconhecimento de conceitos, mineração de relações, mineração interactiva.

Resumo

A rápida evolução e proliferação de uma rede mundial de computadores, a Internet, resultou num esmagador e constante crescimento na quantidade de dados e informação publicamente disponíveis, o que também se verificou na biomedicina. No entanto, a inexistência de estrutura em dados textuais inibe o seu processamento direto por parte de soluções informatizadas. Extração de informação é a tarefa de mineração de texto que pretende extrair automaticamente informação de fontes de dados de texto não estruturados. O objetivo do trabalho descrito nesta tese foi essencialmente focado em construir soluções inovadoras para extração de informação biomédica a partir da literatura científica, através do desenvolvimento de aplicações simples de usar por programadores e bio-curadores, capazes de fornecer resultados mais precisos, usáveis e de forma mais rápida. Começámos por abordar o reconhecimento de nomes de conceitos - uma tarefa inicial e fundamental - com o desenvolvimento de Gimli, uma solução baseada em inteligência artificial que aplica uma estratégia incremental para otimizar as características linguísticas extraídas do texto para cada tipo de conceito. Posteriormente, Totum foi implementado para harmonizar nomes de conceitos provenientes de sistemas heterogéneos, oferecendo uma solução mais robusta e com melhores resultados. Esta aproximação recorre a informação contida em corpora heterogéneos para disponibilizar uma solução não restrita às característica de um único corpus. Uma vez que as soluções anteriores não oferecem ligação dos nomes a bases de conhecimento, Neji foi construído para facilitar o desenvolvimento de soluções complexas e personalizadas para o reconhecimento de conceitos nomeados e respectiva normalização. Isto foi conseguido através de uma plataforma modular e flexível focada em rapidez e desempenho, integrando um vasto conjunto de módulos de processamento optimizados para o domínio biomédico. De forma a disponibilizar identificação de conceitos biomédicos em tempo real, BeCAS foi desenvolvido para oferecer um serviço, aplicação e *widget* Web. A extracção de relações entre conceitos também foi abordada através do desenvolvimento de TrigNER, uma solução baseada em inteligência artificial para o reconhecimento de palavras que desencadeiam a ocorrência de eventos biomédicos. Esta ferramenta aplica um algoritmo automático para encontrar as melhores características linguísticas e parâmetros para cada tipo de evento. Finalmente, de forma a auxiliar o trabalho de bio-curadores, Egas foi desenvolvido para suportar a anotação rápida, interactiva e colaborativa em tempo real de documentos biomédicos, através da anotação manual e automática de conceitos e relações de forma contextualizada. Resumindo, este trabalho contribuiu para a actualização mais precisa das actuais bases de conhecimento, auxiliando a formulação de hipóteses e a descoberta de novo conhecimento.

Keywords

Bioinformatics, text mining, information extraction, concept recognition, relation mining, interactive mining.

Abstract

The rapid evolution and proliferation of a world-wide computerized network, the Internet, resulted in an overwhelming and constantly growing amount of publicly available data and information, a fact that was also verified in biomedicine. However, the lack of structure of textual data inhibits its direct processing by computational solutions. Information extraction is the task of text mining that intends to automatically collect information from unstructured text data sources. The goal of the work described in this thesis was to build innovative solutions for biomedical information extraction from scientific literature, through the development of simple software artifacts for developers and biocurators, delivering more accurate, usable and faster results. We started by tackling named entity recognition - a crucial initial task - with the development of Gimli, a machine-learning-based solution that follows an incremental approach to optimize extracted linguistic characteristics for each concept type. Afterwards, Totum was built to harmonize concept names provided by heterogeneous systems, delivering a robust solution with improved performance results. Such approach takes advantage of heterogenous corpora to deliver cross-corpus harmonization that is not constrained to specific characteristics. Since previous solutions do not provide links to knowledge bases, Neji was built to streamline the development of complex and custom solutions for biomedical concept name recognition and normalization. This was achieved through a modular and flexible framework focused on speed and performance, integrating a large amount of processing modules optimized for the biomedical domain. To offer on-demand heterogenous biomedical concept identification, we developed BeCAS, a web application, service and widget. We also tackled relation mining by developing TrigNER, a machine-learning-based solution for biomedical event trigger recognition, which applies an automatic algorithm to obtain the best linguistic features and model parameters for each event type. Finally, in order to assist biocurators, Egas was developed to support rapid, interactive and real-time collaborative curation of biomedical documents, through manual and automatic in-line annotation of concepts and relations. Overall, the research work presented in this thesis contributed to a more accurate update of current biomedical knowledge bases, towards improved hypothesis generation and knowledge discovery.

Contents

List of Figures

List of Tables

Acronyms

AL Active Learning. 92
API Application Programming Interface. 87, 132–135
ASO Alternating Structure Optimization. 44
ATC Anatomical Therapeutic Chemical. 24

BoW Bag-of-Words. 57

CARO Common Anatomy Reference Ontology. 26
ChEBI Chemical Entities of Biological Interest. 26
CL Cell Ontology. 26
CLI Command Line Interface. 7, 8, 87, 117, 135, 136, 174
CRF Conditional Random Field. 7–9, 41–44, 59, 68, 73–75, 77, 78, 81, 85, 86, 88, 90, 91, 94, 96, 102, 114, 142, 146, 150, 151, 153, 173, 174
CSS Cascading Style Sheets. 132, 163
CTD Comparative Toxicogenomics Database. 25
CUI Concept Unique Identifier. 46

DDI Drug-Drug interaction. 50, 52, 54, 55, 65, 155
DFA Deterministic Finite Automaton. 35, 113, 141
DM Data Mining. 2
DNA Deoxyribonucleic Acid. 4, 7, 55, 88, 92, 114, 173
DO Disease Ontology. 26

EM Expectation Maximization. 44
ExPASy Expert Protein Analysis System. 25

GeNS Genomic Name Server. 32
GO Gene Ontology. 17, 26
GSC Gold Standard Corpus. 18, 27

HGNC HUGO Gene Nomenclature Committee. 24
HGP Human Genome Project. 3

HMDB Human Metabolome Database. 25
HMM Hidden Markov Model. 41, 42, 45
HTML HyperText Markup Language. 132, 163
HTTP Hypertext Transfer Protocol. 132, 133
HTTPS Hypertext Transfer Protocol Secure. 164, 171

IAA Inter Annotator Agreement. 18, 169, 171
ICD International Classification of Diseases. 24
IDF Inverse Document Frequency. 97
IE Information Extraction. 2, 3, 5–7, 15–17, 19, 20, 106, 176
IR Information Retrieval. 2

JSON JavaScript Object Notation. 115, 116, 132, 133, 146

KEGG Kyoto Encyclopedia of Genes and Genomes. 25, 54

LVG Lexical Variant Generation. 32

MEDLINE Medical Literature Analysis and Retrieval System Online. 4, 8, 28, 46, 58, 65, 74, 84, 86, 90, 92, 97, 103, 121, 131, 132, 153
MedRA Medical Dictionary for Regulatory Activities. 26
MEMM Maximum Entropy Markov Model. 42, 68
MeSH Medical Subject Headings. 25, 47, 58
ML Machine Learning. 7–9, 12, 28, 33–37, 41, 45–49, 57–59, 61, 63, 64, 66, 68, 73, 81, 94, 95, 105, 115–119, 125, 132, 138, 142, 145, 146, 149, 152, 167, 174, 176, 177
MRD Machine Readable Dictionary. 48, 49
MUC Message Understanding Conference. 2, 3

NCBI National Center for Biotechnology Information. 26
NER Named Entity Recognition. 2, 6–8, 12, 18, 24, 28, 29, 39, 41, 42, 44, 46, 57, 68, 73, 80, 87, 89–92, 97, 102, 105–107, 113, 114, 122, 123, 136, 152, 173, 174, 176
NLP Natural Language Processing. 8, 20, 22, 28, 113, 139

OMIM Online Mendelian Inheritance in Man. 25

PDB Protein Data Bank. 25
PharmGKB Pharmacogenomics Knowledge Base. 25
POS Part-of-Speech. 20, 21, 38, 44, 47, 57, 58, 61, 71, 77, 81, 84, 96, 113, 119, 133, 136, 139, 142
PPI Protein-Protein interaction. 49, 51, 52, 54, 55, 57, 58, 60–62, 64, 65, 155–157, 166–169
PRO Protein Ontology. 26

RAM Random Access Memory. 86, 131, 152

RDF Resource Description Framework. 17
REST Representational State Transfer. 132, 134, 164, 166
RNA Ribonucleic Acid. 7, 55, 88, 92, 114, 173
ROI Region of Interest. 112

SNOMED Systematized Nomenclature of Medicine. 25
SO Sequence Ontology. 26
SOAP Simple Object Access Protocol. 166
SSC Silver Standard Corpus. 18
SVD Singular Value Decomposition. 57
SVG Scalable Vector Graphics. 164
SVM Support Vector Machine. 9, 41, 42, 44, 45, 47, 57, 59, 63, 68, 91, 153, 168

TM Text Mining. 2, 177
TSV Tab-separated Values. 114, 141

UMLS Unified Medical Language System. 26, 32, 46, 48, 65

VSM Vector Space Model. 47
VTT Variable Trigonometric Threshold. 57

W3C World Wide Web Consortium. 17
WHO World Health Organization. 24
WSD Word Sense Disambiguation. 46–49, 59

XML Extensible Markup Language. 86, 91, 112, 115, 117, 133, 136, 146

Chapter 1

Introduction

For thousands of years, human beings felt the need to share and store information in some way, taking advantage of physical materials for "permanent" storage, from stones to silicon. In the Neolithic period, petroglyphs were used to represent and share popular thoughts through symbols. Nowadays, at the information age, there was a shift to a society based on informatization, creating a knowledge-based world surrounded by high-technology devices where information storage and transmission around the globe became simple routine tasks. A recent study presented by Hilbert and López [1] shows that in 1986 the World had the capacity to store 539MB of data per person, which grew to more than 43GB in 2007. Regarding the World's capacity to exchange information, it grew from 281 petabytes (1000 terabytes) in 1986 to 65 exabytes (1000 petabytes) in 2007. In summary, an overwhelming and constantly growing amount of publicly available data and information was observed, a direct consequence of the rapid evolution and proliferation of a world-wide computerized network, the Internet.

Most public data is available in digital format in the form of texts, videos, sounds and images, which facilitates the access by the widespread availability of internet-capable devices, but the lack of structure inhibits its direct usage by computerized solutions, making its processing and interpretation much more difficult. Typically, in order to access information, one must search for the appropriate data, extract information, and understand its meaning. This type of data is known as unstructured, since it is not organized in a pre-defined way or does not follow a specific data model. Blumberg and Atre [2] estimated that more than 85 percent of all business information exists as unstructured data. Overall, this information explosion consumed huge amounts of expensive and valuable resources, both human and technical, demanding the creation of tools to properly manage unstructured of data.

In order to understand the role of computer-based solutions, it is important to acknowledge the content of the human mind, which can be classified into five different categories, according to Ackoff [3]:

- Data: symbols;

- Information: data that are processed to be useful, answering questions such as “who”, “what”, “where” and “when”;
- Knowledge: application of data and information, answering questions such as “how”;
- Understanding: appreciation of “why”;
- Wisdom: evaluated understanding.

Following the same path, the main goal of computational solutions for information processing is to extract information from data and then induce knowledge from the combination of several sources. To accomplish this, it is necessary to convert unstructured data into a structured form, targeting a specific goal and domain. Going further, by associating the information extracted from large amounts of data, computerized solutions may also contribute to discover hidden relations that enable gaining new knowledge that would otherwise remain undiscovered.

With an overwhelming amount of information recorded in texts, there is high research interest in new techniques that can identify, extract, manage, integrate, and exploit it. Text Mining (TM) is the field of Data Mining (DM) that deals with those requirements, aiming to derive high-quality information from text. To achieve this objective, two main directions can be followed in TM research:

- Information Extraction (IE): aims to extract specific information from unstructured data, building a structured and unambiguous representation of concepts and relations between them [4];
- Information Retrieval (IR): its goal is the representation, storage, organization and access to information items, providing easy access to the information in which the final user is interested [5].

Figure 1.1 illustrates the global TM pipeline, presenting the results provided by each task and the relations between them. As stated before, the main input of these systems is natural language text, which is processed to extract specific information in a structured manner, enabling its readability by computers. In order to streamline the process of filtering the data, information retrieval techniques are applied, to provide only the information that the final user is searching for. In the end, the information from unstructured data is filtered and presented in a simple and structured form, focusing on the information requested by final users.

IE and its several methods were introduced by the Message Understanding Conferences (MUCs) [6], which defined the requirements, evaluation strategies and the several tasks that need to be performed in order to accomplish the IE idea and goals successfully. These tasks include:

- Named Entity Recognition (NER): identify chunks of text as specific entity names, such as people and organizations;

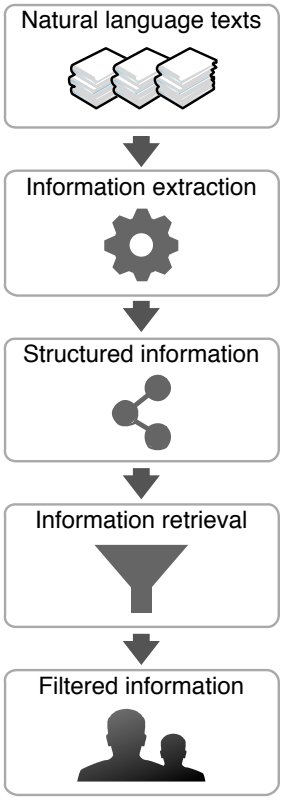


Figure 1.1: Global pipeline of text mining.

- Normalization and disambiguation: associate an unique meaning to a concept name (e.g., "June" could refer to a person's name, a calendar month or a gene);
- Coreference: identify when two different expressions refer to the same concept (e.g., "David" and "he" in the same sentence may refer to the same person);
- Relation mining: extract relations between concepts (e.g., considering the entities "Barack Obama" and "USA", the relation "Barack Obama → President → USA" should be extracted if it is present in the text in some manner);
- Summarization: extract and compile main ideas of a text;
- Classification: identify prime themes of a specific text (e.g., sports, politics, and arts);

In the seven editions of MUC they applied the previously described tasks on people and organizations information mining, leading to state-of-the-art solutions and baseline results for following IE research.

1.1 Biomedical information extraction

The Human Genome Project (HGP) [7] dramatically changed the landscape of biomedical research, by sequencing the human genome and making it publicly available. Such detailed

genetic data provided enormous challenges to disclose knowledge not previously explored, and affecting almost all areas of health, since many connections between different fields are based on related genomic mechanisms. Over the years, genome sequencing became considerably cheaper. The 1000 Genomes Project [8] was the first to sequence genomes of a large number of people and to provide a more comprehensive resource on human genetic variation. Such significant progress resulted in an improved and faster biomedical research, contributing to an increasing amount of published scientific articles, reporting new and focused findings. Considering the wide impact of genetic information, establishing connections between discoveries from different research specialties is fundamental. However, communication between highly specialized fields is poor [9], which hinders progress and respective finding of new knowledge.

The overwhelming amount of textual information in biomedicine was verified with the rapid growth in the number of published documents, such as articles, books and technical reports. In fact, Lawrence Hunter [10] demonstrated an exponential growth of the biomedical literature. Medical Literature Analysis and Retrieval System Online (MEDLINE), which is the United States of America National Library of Medicine (NLM) premier bibliographic database, in 2013 contained over 21 million references to journal articles in life sciences. It continues to be daily updated, and, since 2005, between 2000-4000 completed references are added each day[1]. At this rate of publication, it is difficult for researchers to keep up with current knowledge and relevant publications from different fields of expertise, limiting the practical application of biomedical science in the form of diagnosis, prevention and treatments.

To properly extract relevant information and maintain existing knowledge resources updated, MEDLINE and other biomedical resources started to manually curate scientific articles, collecting information about biomedical concepts, such as genes, proteins and chemicals. However, with the increasing amounts of data, this became a hard and very expensive task. For instance, Baumgartner et al. [11] argues that manually curating and completing some genomic resources may take decades to be completed. This situation naturally led to the development of computational solutions to extract specific biomedical information from scientific articles, in order to "shift the burden of information overload from the researcher to the computer" [12]. However, the biomedical domain presents various complex challenges that hinder the application of text mining techniques. Primarily, the complex biomedical knowledge network covers a wide range of interrelated concepts (e.g., from Deoxyribonucleic Acid (DNA) to body parts) and processes (e.g., from DNA transcription to hormone regulation). Thus, linking concepts and/or processes of the two edges of the knowledge spectrum is considered fundamental to properly understand "how" and "why" things happen. As a consequence of this complexity, the biomedical domain is composed of many fields and sub-fields of expertise with restricted communication, which limits the overall perspective of current knowledge. Moreover, since biomedicine is not an exact science, it is in constant evolution,

[1] `http://www.nlm.nih.gov/bsd/medline_lang_distr.html`

with new knowledge and principles emerging constantly. On the other hand, the textual representation of biomedical information also presents various characteristics that hinder its precise processing and extraction. For instance, the use of specialized non-standardized terminology results in highly descriptive texts with high levels of ambiguity, which is a direct consequence of the domain complexity. As a result of those levels of complexity, biomedical text mining is considered a proving ground for the application and development of innovative IE solutions, since it is assumed that a technique that performs well in the biomedical domain will perform equally well in a different and simpler domain [13]. Moreover, biomedical researchers are highly interested in the results of biomedical text mining research, in order to advance their own work and sub-field of interest.

Due to the aforementioned challenges, the result of biomedical text mining solutions contain mistakes that should be processed carefully. Nonetheless, even different expert annotators have different interpretations of the same data, which results in active and complex discussions with different opinions. For instance, considering the manual annotation of a set of documents, if curators do not follow precisely defined guidelines, the resulting annotations may be considerably different, presenting low agreement between the various curators [14]. Thus, the quality of the information provided by automatic solutions may be directly limited by the quality of the information in the manually annotated data, since learn-by-example solutions try to reproduce behaviors contained in the training data. Additionally, current automatic solutions are not able to perform as well as expert annotators, since it is difficult to convert a curator's domain knowledge into a structured representation.

Overall, by collecting information from scientific articles, biomedical text mining contributes to: *1*) keep current knowledge bases updated; and *2*) generate hypothesis for knowledge discovery. Current genomic resources already use automatic text mining solutions to keep databases updated, such as BioLexicon [15] for chemicals, genes and disorders and STRING [16] for protein-protein interactions. On the other hand, biomedical text mining is also applied to infer hidden relationships between concepts, generating hypothesis and discovering new knowledge. Such task is typically performed by applying the ABC model, which was proposed by Swanson [17] and intends to extract concept relations from different articles (A is related with B, and B is related with C) to infer new and indirect relationships (A is related with C). Thus, such automatic solutions help ranking target articles by automatically extracting concept relationships, also suggesting likely related concepts regarding a specific field. For instance, iHop [18] applies this idea to find hidden protein-protein interactions, and FACTA+ [19, 20] supports finding indirectly associated proteins, drugs and disorders.

There are various examples of success regarding the applicability of text mining solutions on biomedical real-life problems, namely in pharmacovigilance, drug discovery and drug repurposing. Swanson [21] was the first one to show the benefits of scientific literature mining in drug discovery and repurposing, presenting two scientific hypothesis, namely the beneficial

effects of fish oil to patients with Raynaud's disease, and the potential of magnesium to treat migraines. Both connections were validated in clinical trials [22, 23] and became well established in nowadays clinical practices. On the other hand, the EU-ADR European project [24] developed an innovative computerized system [25] for pharmacovigilance, detecting adverse drug reactions to supplement spontaneous reporting systems and translate scientific and clinical evidences into patient safety and health benefit. To achieve this, the authors applied text mining techniques to analyze electronic health records of 30 million patients in order to detect "signals", i.e., combinations of drugs and suspected adverse events that warrant further investigation. In the end, the authors confirmed the association between the use of non-steroidal anti-inflammatory drugs and upper gastrointestinal bleeding. Such examples show the success of applying text mining techniques on real-life problems, which provide several benefits, namely the need for less money and time for drug discovery, testing and vigilance, resulting in improved healthcare services.

Despite many biomedical IE solutions have already been developed and applied successfully in real-life scenarios, there is still an enormous lack of consolidated tools and algorithms that can be widely used by researchers of the biomedical community.

1.2 Research goals

The main goal of this doctorate was to investigate new and improved methods to streamline the development of high performance biomedical IE solutions, tackling the tasks of concept recognition (NER and normalization), relation mining and interactive curation. To attain this major objective, various goals were carefully defined:

- Architect software solutions to minimize and/or solve identified problems by taking advantage of the best techniques for each issue, through a detailed analysis of the existing limitations, required features, data models and implementation details;
- Develop new and improved solutions to support and perform some biomedical information extraction tasks, facilitating their completion and making them simple routine tasks for developers and/or end-users;
- Participate in domain challenges to evaluate the performance and/or behavior of developed solutions, also keeping updated with the last efforts on various mining tasks;
- Contribute to national and international research projects by applying the developed solutions and/or developing or adapting solutions focused on specific high-end goals;
- Collaborate with international research groups to promote knowledge sharing and partnerships.

1.3 Contributions

The work presented in this thesis contributed positively to the biomedical text mining community in various aspects:

- Reviews: provided easy to understand and overall overviews and analysis of specific tasks of the biomedical IE domain;
- Methods: applied new techniques and followed innovative approaches to solve or minimize existing problems;
- Software: developed new open and closed-source tools, libraries and frameworks to make specific IE tasks accessible, providing detailed documentation for users and developers;
- Analyses: detailed and new analyses to confirm known facts or to sustain new conclusions;
- Personal: developed a personal insight of the domain, regarding its problems and possible paths to follow.

The first contribution resulted in detailed surveys and critical analysis of specific tasks of the biomedical information extraction domain. The first review [26] presents a careful analysis of distinct biomedical NER approaches, describing applied techniques, presenting practical implementations, and evaluating and comparing the achieved performance results. The second work [27] surveys Machine Learning (ML) based solutions for heterogenous biomedical concept recognition, with an in-depth analysis of existing solutions regarding techniques and achieved performance results and considering multiple biomedical concept types.

The second contribution is a NER system that outperforms existing open-source solutions. Gimli [28] is a Command Line Interface (CLI) tool to perform ML-based recognition of biomedical concept names. It is open source and publicly available at `http://bioinformatics.ua.pt/gimli`, providing detailed documentation for users and developers. Gimli uses Conditional Random Fields (CRFs) with a rich set of features, combining annotations of heterogenous models with a simple confidence-based harmonization technique. Moreover, two post-processing methods are applied to improve annotations quality. The recognition of different biomedical concepts is performed using different CRF models, whose feature set is optimized through an incremental approach. Such technique allowed an in-depth analysis of the best features required to recognize different concept types, providing a better understanding of their linguistic and complexity characteristics. Through Gimli, we confirm the importance of tokenization, showing that our tokenization delivers improved results for biomedical NER. Moreover, we provide a comprehensive analyses of local context features, comparing windows with conjunctions, and discussing the added value of building new features by taking advantage of conjunctions. Finally, we also demonstrate the advantages of using dependency parsing features in different concept types. Gimli was applied in two corpora, identifying gene/protein, DNA, Ribonucleic Acid (RNA), cell type and cell line concept

names, outperforming previously available open source tools. Gimli was further applied in the recognition of chemical compound and drug names [29], also achieving encouraging and positive performance results.

Considering the distinct approaches followed by other NER systems, and the positive outcomes provided by ensemble solutions, the next challenge was to investigate a better strategy for named entity harmonization. Totum [30] is a ML-based solution that resulted from such efforts, targeting the harmonization of gene and protein names provided by multiple heterogenous NER systems. It uses CRFs to harmonize heterogenous annotations provided by four systems applying different recognition techniques and using four corpora with different characteristics to perform training and evaluation. By doing so, Totum delivers an innovative cross-corpus solution to perform gene and protein names harmonization, which is not constrained to a specific corpus as the original systems are. Moreover, the provided annotations take advantage of a rich knowledge base, creating unique and reasoned guidelines that respect as much as possible the heterogeneity of the various corpora. When evaluated on each corpus and in a merged corpus simulating the MEDLINE, Totum delivered significant improvements in comparison with state-of-the-art solutions and each individual system. By analyzing the provided annotations, we concluded that improved results were achieved due to the deletion of incorrect annotations, the recognition of annotations discarded by other approaches, and the creation of new entity names. Moreover, we also emphasized the differences between the various corpora and respective annotation guidelines. This research work was developed during an internship in the European Bioinformatics Institute (EBI) with the supervision of Dietrich Rebholz-Schumman, as a contribution to the European CALBC project.

After exploring the most different and advanced approaches for NER, the next research goal was to tackle heterogeneous and large-scale biomedical concept recognition, providing identifiers from databases to every extracted concept name. Neji [31, 32] is a modular framework and tool to support the development of concept recognition solutions, being specifically optimized for the biomedical domain through dedicated and optimized modules and supported standards. It is open source and publicly available at `http://bioinformatics.ua.pt/neji`, providing detailed documentation for users and developers. Neji delivers high modularity together with fast processing speeds and high performance results, integrating modules for Natural Language Processing (NLP), concept recognition, disambiguation and post-processing. The extracted information is stored in an innovative concept tree, supporting structured ambiguity and multiple identifiers per concept. Moreover, the framework also integrates multi-threaded documents processing together with standard input and output formats. In order to make Neji accessible, it also integrates a CLI annotation tool, which allows users to easily perform offline annotation of large amounts of documents with custom dictionaries and ML models with normalization resources. The reliability of Neji was confirmed by annotating three different corpora with a total of nine concept types, outperforming exist-

ing solutions for heterogenous concept recognition. Such throughout performance analysis was the first to consider both names and identifiers with a large amount of heterogenous biomedical concepts. In order to streamline the application of heterogeneous biomedical concept recognition, we took advantage of Neji and developed BeCAS [33], a web application, web-service and widget for on-demand biomedical concept identification. It is available at `http://bioinformatics.ua.pt/becas` with detailed documentation for users and developers. Concept recognition features are provided through web-services, supporting selective annotation of eleven biomedical concepts. The web application applies innovative annotations visualization and filtering interfaces, supporting inline nested concept names and providing link-outs to reference curated databases. The widget allows easy integration of BeCAS features in any web page. BeCAS was developed in collaboration with Tiago Nunes.

The third contribution of this doctorate is on event mining. Taking advantage of previous results, namely the performance of Gimli and the flexibility and speed of Neji, we tackled a challenging and more linguistic processing and knowledge intensive task: trigger recognition for biomedical event mining. TrigNER [34] is a ML-based solution to perform automatic and optimized recognition of biomedical event triggers. It is open source and publicly available at `http://bioinformatics.ua.pt/trigner`, providing detailed documentation for users. TrigNER applies CRFs with a rich feature-set and post-processing modules, applying an innovative automatic optimization method to obtain the best feature-set and model parameters for each event trigger. Such technique removes the hard task of manually optimizing ML models. Besides simplifying the optimization process, such technique also allows to easily identify the complexity and linguistic characteristics of different triggers. When evaluated against manually annotated corpora with nine gene-centric event triggers, TrigNER outperformed existing solutions on various event trigger types. Moreover, we also showed that CRFs are able to perform as well as Support Vector Machines (SVMs) in the recognition of event triggers. This research work was conducted as a result of an internship at the Erasmus Medical Center (EMC), in collaboration with Quoc-Chin Bui.

Finally, we worked in a comprehensive platform for interactive curation, in order to assist biocurators in their daily tasks, and take advantage of the previously developed automatic solutions. Egas [35] is an innovative web-based platform for biomedical collaborative curation-as-a-service, supporting manual and automatic annotation of concepts and relations. It is available at `http://bioinformatics.ua.pt/egas` with detailed usage documentation. Egas user interface was developed targeting simplicity and intuitive interactions, through inline document visualization, filtering, insertion and deletion of annotations and relations. Moreover, it provides a rich set of features to support the complete workflow of knowledge curation, such as integrated project management, import and export features to/from local and remote servers, automatic and state-of-the-art annotation services, and innovative real-time collaboration. Egas was developed on top of standard web technologies, in order to enable

fast processing and visualization of documents in modern web-browsers. When evaluated by expert annotators at the BioCreative IV interactive annotation task [36], Egas obtained remarkable results in terms of usability, reliability and performance. Egas was developed in collaboration with Jóni Lourenço.

Overall, this research work contributed with innovative applications, frameworks and libraries for the biomedical text mining community. By improving the application of biomedical information extraction methods on scientific literature, the work on this thesis further contributed to keep current knowledge bases updated, generating new hypothesis towards knowledge discovery.

1.4 Results

This doctorate generated a number of outcomes in terms of publications and software solutions, which are summarized as follows:

- Publications:
  - Book chapters:
    * D. Campos, S. Matos, and J. L. Oliveira, "Current methodologies for biomedical Named Entity Recognition," in *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, M. Elloumi and A. Y. Zomaya, Eds. John Wiley & Sons, Inc., Dec. 2013, pp. 839–868;
    * D. Campos, S. Matos, and J. L. Oliveira, "Biomedical Named Entity Recognition: A Survey of Machine-Learning Tools," in *Theory and Applications for Advanced Text Mining*, S. Sakurai, Ed. InTech, 2012, pp. 175–195.
  - International journals:
    * D. Campos, J. Lourenço, S. Matos, and J. L. Oliveira, "Egas: a web-based document curation platform," *Database (Oxford), Under Review*
    * D. Campos, Q.-C. Bui, S. Matos, and J. L. Oliveira, "TrigNER: automatically optimized biomedical event trigger recognition on scientific documents," *Source code for biology and medicine*, vol. 9, no. 1, Jan. 2014
    * D. Campos, S. Matos, and J. L. Oliveira, "A modular framework for biomedical concept recognition." *BMC bioinformatics*, vol. 14, no. 1, p. 281, Sep. 2013;
    * T. Nunes, D. Campos, S. Matos, and J. L. Oliveira, "BeCAS: biomedical concept recognition services and visualization." *Bioinformatics (Oxford, England)*, vol. 29, no. 15, pp. 1915–1916, Jun. 2013;
    * D. Campos, S. Matos, and J. L. Oliveira, "Gimli: open source and high-performance biomedical name recognition." *BMC bioinformatics*, vol. 14, p. 54, 2013;

  * D. Campos, S. Matos, I. Lewin, J. L. Oliveira, and D. Rebholz-Schuhmann, "Harmonization of gene/protein annotations: towards a gold standard MEDLINE." *Bioinformatics (Oxford, England)*, vol. 28, no. 9, pp. 1253–1261, May 2012;
  * Z. Lu, H.-Y. Kao, C.-H. Wei, M. Huang, J. Liu, C.-J. Kuo, C.-N. Hsu, R. T.-H. Tsai, H.-J. Dai, N. Okazaki, H.-C. Cho, M. Gerner, I. Solt, S. Agarwal, F. Liu, D. Vishnyakova, P. Ruch, M. Romacker, F. Rinaldi, S. Bhattacharya, P. Srinivasan, H. Liu, M. Torii, S. Matos, D. Campos, K. Verspoor, K. M. Livingston, and W. J. Wilbur, "The gene normalization task in BioCreative III." *BMC bioinformatics*, vol. 12 Suppl 8, p. S2, 2011.

– International conferences and workshops:

  * D. Campos, S. Matos, and J. L. Oliveira, "Chemical name recognition with harmonized feature-rich conditional random fields," *Fourth BioCreative Challenge Evaluation Workshop*, vol. 2, pp. 82–87, 2013;
  * D. Campos, J. Lourenço, T. Nunes, R. Vitorino, P. Domingues, S. Matos, and J. L. Oliveira, "Egas-Collaborative Biomedical Annotation as a Service," in *Fourth BioCreative Challenge Evaluation Workshop*, Bethesda, Maryland, USA, Oct. 2013, pp. 254–259;
  * Q. C. Bui, E. M. Van Mulligen, D. Campos, and J. A. Kors, "A fast rule-based approach for biomedical event extraction," in *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, Aug. 2013, pp. 104–108;
  * D. Campos, S. Matos, and J. L. Oliveira, "Neji: a tool for heterogeneous biomedical concept identification," in *BioLINK SIG, ISMB/ECCB*, Berlin, Germany, Jul. 2013, pp. 28–31;
  * D. Campos, D. Rebholz-Schuhmann, S. Matos, and J. L. Oliveira, "A CRF-based approach to harmonize heterogeneous gene/protein annotations," in *Second CALBC Workshop*, Cambridge, UK, Mar. 2011, pp. 17–18;
  * D. Campos, S. Matos, and J. L. Oliveira, "Annotating the CALBC corpus with a machine learning harmonization approach," in *Second CALBC Workshop*, Cambridge, UK, Mar. 2011, pp. 43–45;
  * P. Lopes, D. Campos, and J. L. Oliveira, "A tagging system for bioinformatics resources," *2010 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB 2010)*, pp. 1–4, 2010;
  * D. Campos, S. Matos, and J. L. Oliveira, "Recognition of Gene/Protein names using Conditional Random Fields," in *International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, Valencia, Spain, Oct. 2010, pp. 275–280;
  * S. Matos, D. Campos, and J. L. Oliveira, "Vector-space models and terminolo-

gies in gene normalization and document classification," in *Proceedings of the Third BioCreative Challenge Workshop*, Bethesda, Maryland, USA, 2010, pp. 119–124.

- Software:
  - Open-source:
    * Gimli[2] (`http://bioinformatics.ua.pt/gimli`);
    * Neji[2] (`http://bioinformatics.ua.pt/neji`);
    * TrigNER (`http://bioinformatics.ua.pt/trigner`).
  - Application service:
    * Becas[2] (`http://bioinformatics.ua.pt/becas`);
    * Egas (`http://bioinformatics.ua.pt/egas`).

1.5 Thesis outline

The outline of the rest of this thesis is as follows (Figure 1.2):

- Chapter 2 presents an in-depth analysis of the current work in biomedical information extraction, focusing on concept recognition, relation mining, event mining and interactive curation, and describing each task's challenges, available resources, applied techniques, existing solutions and overall achieved performance results;
- Chapter 3 describes Gimli, an open source and ML-based solution for biomedical named entity recognition, performing automatic identification of gene/protein, DNA, RNA, cell type and cell line names;
- Chapter 4 introduces Totum, a ML-based solution that harmonizes gene/protein annotations provided by heterogeneous NER systems, which was optimized and evaluated against a combination of manually curated corpora;
- Chapter 5 presents Neji, an open source framework optimized for biomedical concept recognition built around four key characteristics: modularity, scalability, speed, and usability. It integrates modules for biomedical natural language processing and concept recognition, which is provided through dictionary matching and ML with normalization. We also describe BeCAS, a web application, web-service and widget that allows identifying more than 1.2 million concepts;
- Chapter 6 describes TrigNER, an open source and ML-based solution for biomedical event trigger recognition, automatically optimizing its characteristics for the recognition of specific event triggers;
- Chapter 7 introduces Egas, a web-based platform for biomedical text mining and collaborative curation, supporting manual and automatic annotation of concepts and relations

[2]Registered software.

through simplistic interfaces built with standard web technologies. It also provides integrated project administration that allows managing annotation guidelines, users, target concepts and relations;

- Chapter 8 presents some concluding remarks of this thesis and highlights directions for future work.

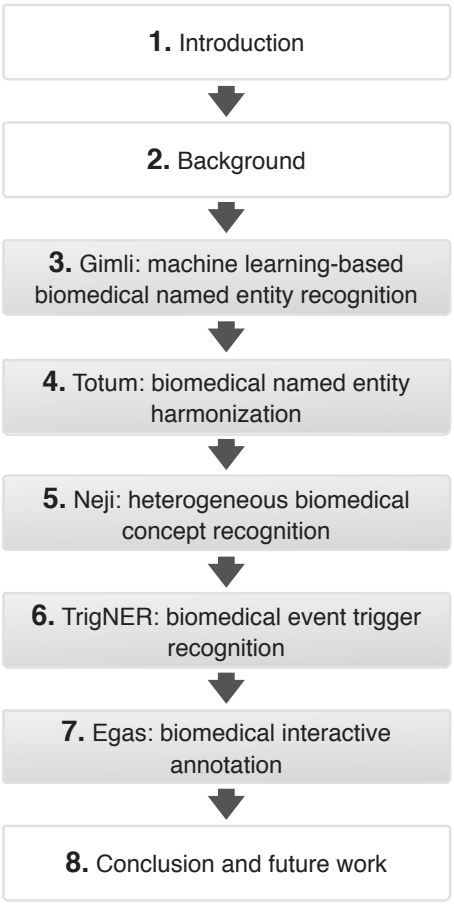


Figure 1.2: Thesis outline with chapters and respective research work dependencies.

Chapter 2

Biomedical information extraction

This chapter is based on:

- D. Campos, S. Matos, and J. L. Oliveira, "Biomedical Named Entity Recognition: A Survey of Machine-Learning Tools," in *Theory and Applications for Advanced Text Mining*, S. Sakurai, Ed. InTech, 2012, pp. 175–195
- D. Campos, S. Matos, and J. L. Oliveira, "Current methodologies for biomedical Named Entity Recognition," in *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, M. Elloumi and A. Y. Zomaya, Eds. John Wiley & Sons, Inc., Dec. 2013, pp. 839–868

The research work presented in this thesis was mainly focused on two essential tasks of biomedical information extraction, namely concept recognition and relation mining. This chapter presents the aforementioned tasks by providing an in-depth description of their goals, associated challenges, applied approaches and techniques, existing solutions and achieved performance results. Such analyses provide a rich and detailed description of current state-of-the-art work of biomedical IE, defining the platform for further improvements and research. Figure 2.1 presents the resources and processing steps considered in this analysis:

- Preliminaries: resources and processing tasks fundamental to the development and evaluation of biomedical IE solutions;
  - Resources: available resources to support the development and evaluation of IE solutions;
  - Evaluation: metrics to understand the behavior of IE solutions and compare to different approaches;
  - Pre-processing: process input documents to simplify information extraction tasks;

- Concept recognition: identify concept names and associate unique identifiers from knowledge-bases;
  - Named entity recognition: identify concept names;
  - Normalization and disambiguation: associate unique identifiers to previously recognized names;
- Relation mining: extract relations between previously collected concepts;

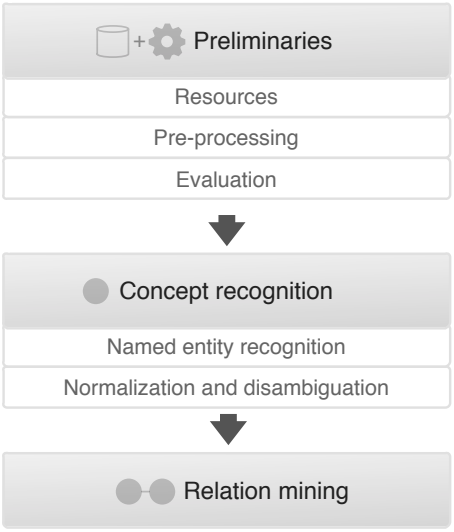


Figure 2.1: Dependencies of processing steps and resources currently applied on biomedical information extraction.

2.1 Preliminaries

Since several IE tasks have many processing steps and/or resources in common, we will describe them first in this section, in order to better understand the global workflow and the most basic methods. Preliminary tasks include collecting resources to support the development of IE solutions, define evaluation metrics to understand the behavior of the developed approaches, and use text pre-processing techniques to enable the automatic and computerized application of IE techniques.

2.1.1 Resources

Resources are used to support the development of biomedical IE solutions, providing domain knowledge input towards the development of automatic solutions with reasoned decisions, or as platforms to evaluate and compare different approaches.

Knowledge bases

There are various types of knowledge bases that are relevant for biomedical IE, namely databases and ontologies. As knowledge bases, the main goal of such resources is to model aspects of reality, presenting different characteristics that make each one more appropriate for different problems. Overall, ontologies provide an excellent way to represent reality, but databases are better to store and search when data is of considerable size [45]. Due to the complexity of the biomedical domain, there are hundreds of databases[1] and ontologies[2] modeling biomedical knowledge. Most resources are focused on collecting detailed data regarding specific concepts, such as gene and protein, drug, chemical and species. For instance, Uniprot [46] is a database that provides a centralized and authoritative resource for protein sequences and functional information. On the other hand, Gene Ontology (GO) [47] provides a set of structured vocabularies for specific biological domains, which can be used to describe gene products in any organism, organized in three ontologies to describe molecular function, biological process, and cellular component.

Considering the complexity of the biomedical domain, there is no resource that integrates all required information. Thus, researchers started working on techniques to integrate different knowledge bases. Resource Description Framework (RDF) [48] is the World Wide Web Consortium (W3C) specification for conceptual description and modeling of information. By taking advantage of RDF, researchers are able to integrate heterogenous sources of information in an unique resource, maintaining existing links between concepts from different knowledge bases. For instance, Bio2RDF [49] was one the first projects to successfully "rdfize" heterogenous resources in biomedicine, building a unique endpoint to access semantic enabled information from KEGG, PDB, MGI, HGNC and NCBI's databases. Moreover, the European Bioinformatics Institute (EBI) also released an RDF platform[3] to enable easy access and integration of gene expression data.

Knowledge bases containing relevant information for each task will be presented throughout this chapter.

Corpora

A corpus is a set of text documents that usually contain annotations focused on specific tasks and domains. Such annotations are used to develop and evaluate implemented solutions. The development of IE solutions is highly dependent on the quality of the information provided in the corpus. A corpus is also used to obtain performance results, allowing understanding the behavior of the system on real-life problems. Such evaluation enables the comparison of distinct solutions to the same problem.

[1] `http://library.buffalo.edu/hsl/biomed/`
[2] `http://bioportal.bioontology.org/ontologies`
[3] `http://www.ebi.ac.uk/rdf`

There are two types of annotated corpora, varying with the source of the annotations:

- Gold Standard Corpus (GSC): annotations are performed manually by expert annotators, following specific and detailed guidelines;
- Silver Standard Corpus (SSC): annotations are automatically generated by computerized solutions.

Manually annotated corpora typically provide valuable and high-quality information curated by human experts. One important factor of such corpora is the Inter Annotator Agreement (IAA), which evaluates the quality and agreement of the information provided by different experts. Thus, a low IAA reflects a disagreement between the annotators, consequently providing inconsistent information that hinders the development and evaluation. Due to the effort and associated costs required to build such corpora, only small amounts of documents and information instances are typically provided. On the other hand, the advantage of automatically generated corpora is the amount of provided information, offering thousands of documents and information instances. However, there is still an active discussion regarding the usage of such information, since a large amount of mistakes is present. Thus, some researchers argue that such information may not be used as primary targets of development and evaluation. Nonetheless, it may be considered in the development of algorithms to provide additional data not available in GSCs.

Corpora also vary in the granularity, considering full-text documents, just their abstracts or selected sentences. Sentence-based corpora are typically targeted at tasks that do not require information context, such as NER. For instance, when performing disambiguation, the context provided in a sentence may not be sufficient, requiring the complete paragraph, section or document. Nonetheless, such corpora provide a good heterogeneous sample of the target domains. On the other hand, full-text documents potentially hold more information than just their abstracts, but require more time and computational resources to be processed. Schuemie et al. [50] considered almost four thousand documents to evaluate information content of abstracts and full-text documents, concluding that the information coverage in full texts is much greater than in abstracts, even though abstracts have higher information density. Moreover, evaluating different sections, the authors concluded that the results section is the one that provides the highest information coverage.

Corpora specifically developed for each task will be presented throughout this chapter.

2.1.2 Evaluation

In order to understand the behavior of the developed system, it is important to measure the accuracy of the generated annotations. This can be performed by annotating a corpus and then comparing the automatic annotations with the ones provided by expert curators. Thus, each automatic annotation must be classified as being a:

- True Positive (TP): the system provides an annotation that exists in the curated corpus;
- True Negative (TN): the non existence of an annotation is correct according to the curated corpus;
- False Positive (FP): the system provides an annotation that does not exist in the curated corpus;
- False Negative (FN): the system does not provide an annotation that is present in the curated corpus.

Exact and approximate matching can be used to obtain performance results and to better understand the behavior of the system. With approximate matching we can understand the performance when minor and non-informative mistakes are discarded.

Performance results are obtained using three important measures: precision, recall and F-measure. Those measures assume values between 0 (worst) and 1 (best). Precision measures the ability of a system to present only relevant items, and is formulated as:

$$Precision = \frac{\text{relevant items retrieved}}{\text{total items retrieved}} = \frac{TP}{TP + FP}. \tag{2.1}$$

On the other hand, recall or sensitivity measures the ability of a system to present all relevant items, and is formulated as:

$$Recall = \frac{\text{relevant items retrieved}}{\text{relevant items in collection}} = \frac{TP}{TP + FN} \tag{2.2}$$

Finally, F-measure is the harmonic mean of precision and recall. The balanced F-measure is most commonly used, and is formulated as:

$$F\text{-}measure = 2\frac{Precision.Recall}{Precision + Recall} \tag{2.3}$$

Other measures are relevant to evaluate binary classification problems, such as accuracy, sensitivity and specificity. Accuracy measures the ability of a system to provide correct predictions (including positive and negative):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

Finally, specificity measures the ability of the system to identify negative results:

$$Specificity = \frac{TN}{FP + TN} \tag{2.5}$$

2.1.3 Pre-processing

Various pre-processing steps are generally applied before performing any biomedical IE task, such as natural language processing and stopword removal.

Natural language processing

NLP solutions can be accomplished by computerized systems in an effective manner. However, it is necessary firstly to properly delimit the documents into meaningful units. Most NLP solutions expect their input to be segmented into sentences, which are the basic units of information and knowledge exchange. Moreover, each sentence should be split in tokens, which are the basic and meaningful units of data processing. Since real-world documents lack such well-defined structure, it is necessary to implement various methods to perform such tasks.

Due to the specificities of the biomedical domain, methods developed for common English may not provide the best outcomes when used on scientific documents. For instance, He and Kayaalp [51] analyzed the application of various tokenizers on biomedical documents, concluding that most solutions are too simplistic for real-life biomedical applications. Similarly, Verspoor et al. [52] compared the performance of various sentence tagging, tokenization, Part-of-Speech (POS) tagging and dependency parsing tools on biomedical full-text documents, showing that domain optimization is fundamental in most tasks. Thus, it is important to develop and use methods optimized to deal with the special linguistic characteristics of biomedical terms.

Figure 2.2 presents the various linguistic processing tasks and their dependencies, illustrating the provided output for the example sentence "Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells.". The dependencies between tasks mean that, for instance, POS tagging should not be performed before tokenization, since the tokens are fundamental to assign the linguistic role tags.

Sentence splitting Sentence splitting is the process of breaking a text document into its respective sentences. In the end, each sentence should provide a specific local, logical and meaningful context for future tasks. Various solutions were developed to perform sentence splitting on biomedical documents, such as Lingpipe[4], GENIA SS [53], JSBD [54], OpenNLP[5] and SPECIALIST NLP[6]. The best performing solutions can achieve an F-measure of 99%.

Tokenization Tokenization is the process of breaking a sentence into its constituent meaningful units, called tokens. It is one of the most important tasks of the IE workflow, since all the following tasks will be based on the tokens resulting from this process. Consequently, various tools were developed specifically for the biomedical domain, such as GENIA Tagger [55], JTBD ,[54] and SPECIALIST NLP. He and Kayaalp [51] present a detailed comparison of various biomedical tokenizers. The best performing solutions can achieve an F-measure of 96%.

[4] `http://alias-i.com/lingpipe`
[5] `http://opennlp.apache.org`
[6] `http://lexsrv3.nlm.nih.gov/Specialist`

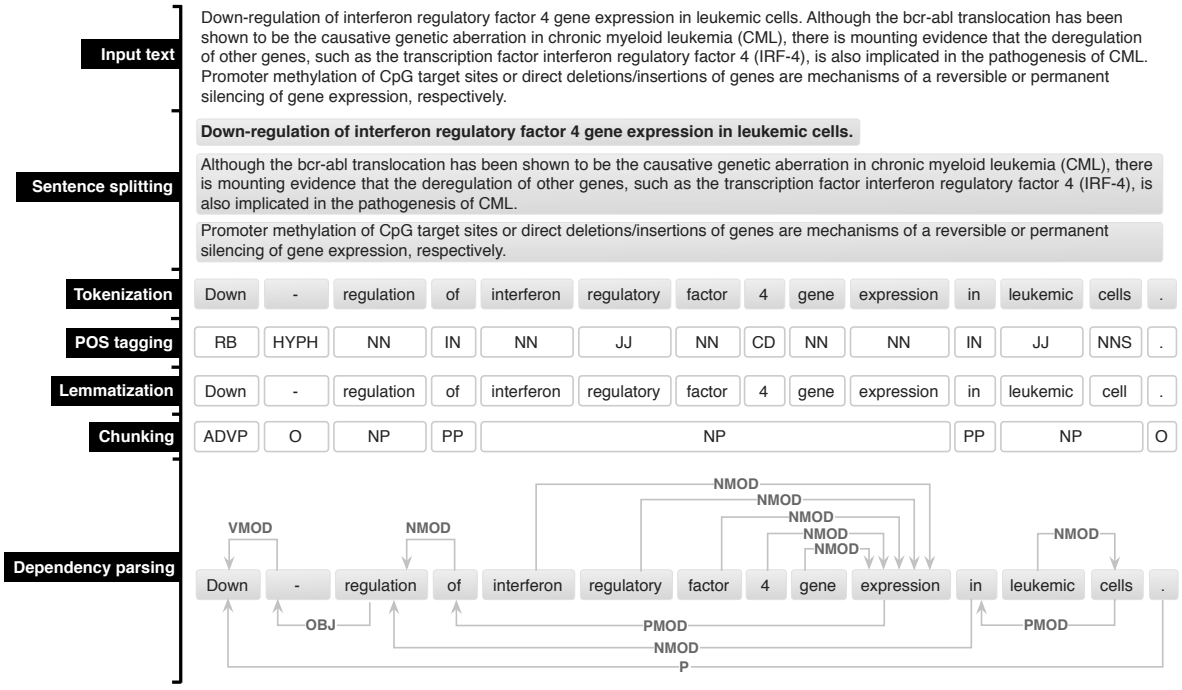


Figure 2.2: Illustration of NLP tasks and their dependencies, presenting the obtained outputs of sentence splitting, tokenization, POS tagging, chunking and dependency parsing considering the sentence "Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells.".

Lemmatization Since morphological variants of words have similar semantic interpretations, they can be considered as equivalent. For this reason, lemmatization can be used to group together inflected forms of a word, in order to process them as a single item. Thus, lemmatization is a robust technique that finds the root form of each word. For instance, the lemma of "was" is "be". There are various solutions for biomedical lemmatization, such as GENIA Tagger and BioLemmatizer [56]. The best performing solutions can achieve an F-measure of 97%.

Part-of-speech tagging To understand the linguistic role of each token in a sentence, it is also possible to associate each token with a particular grammatical category based on its definition and context, a procedure called POS tagging. In the end, each token is tagged as providing a specific linguistic contribution, such as Noun (NN), Adjective (JJ) or Adverb (RB). GENIA Tagger, Lingpipe and OpenNLP are examples of solutions supporting biomedical POS tagging. The best performing solutions achieve an F-measure of 90%.

Chunking Chunking intends to provide an understanding of the structure of a sentence, grouping together tokens with similar syntactic roles. Thus, it splits a sentence into groups

of tokens that constitute a grammatical unit, like noun phrase (NP), verb phrase (VP) or preposition phrase (PP). GENIA Tagger, Lingpipe and OpenNLP are examples of solutions that support biomedical chunking. Kang et al. [57] presents a detailed comparison of chunkers for the biomedical domain. The best performing solutions achieve a top of 95% of F-measure.

Dependency parsing Previous NLP tasks provide a local analysis of the sentence. On the other hand, dependency parsing allows to understand in detail how tokens and chunk phrases are related in a sentence, providing an in-depth syntactic analysis of the sentence. The relations are also categorized representing specific grammatical roles, such as noun modifier (NMOD), verb modifier (VMOD) or preposition modifier (PMOD). For instance, considering Figure 2.2, the dependency parser output indicates that "leukemic" is a noun modifier of "cells". GDep [58], Stanford Parser [59], Enju [60], Berkeley [61] and Charniak-Lease [62] are examples of solutions for biomedical dependency parsing. Verspoor et al. [52] presents a detailed comparison of dependency parsers for the biomedical domain, where the best solutions achieve a top F-measure performance of 65%.

Stopwords

One of the most commonly used techniques, is to discard words that are already known to be non-informative and that produce a large amount of mistakes. This filtering contributes to improved performance results and consequently reduces the amount of data to be processed. Pubmed provides a list of stopwords especially obtained for the biomedical domain[7]. Examples of such words in English are "be", "can", "therefore" and "which".

2.2 Concept recognition

A concept corresponds to a biomedical entity present on a curated resource, and it is used to represent current knowledge. Typically, a resource is a database or ontology that contains and relates information regarding a specific knowledge sub-field, where each concept has an unique identifier. For instance, "Cellular tumor antigen p53" is a protein (concept type) that is present on the Uniprot database with the unique identifier "P04637". Thus, concept recognition is the task that intends to automatically extract names of concepts and relate them with unique identifiers from curated resources (Figure 2.3). Considering dozens of concept types, applying this technique allows to automatically extract names of various biomedical concepts from millions of documents.

Concept recognition is a crucial initial step in information extraction, since next steps rely on its output to be performed successfully. However, biomedical documents present several challenges that make the application of these techniques even harder. The main challenge is

[7] `http://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T43`

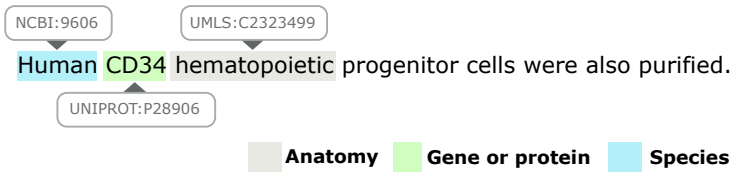


Figure 2.3: Illustration of the biomedical concept recognition task, where each recognized concept name is associated to a unique identifier from a curated resource.

related with terminology, due to the complexity of the used terms for biomedical concepts and processes [63, 64]:

- Non-standardized naming convention: a concept name could be found in various spelling forms (e.g., "N-acetylcysteine", "N-acetyl-cysteine", and "NAcetylCysteine");
- Ambiguous names: one name could be related with more than one concept, depending on the text context;
- Abbreviations: abbreviations are frequently used (e.g., "TCF" may refer to "T cell factor" or to "Tissue Culture Fluid");
- Descriptive naming convention: many concept names are descriptive, which makes their recognition a complex task (e.g., "normal thymic epithelial cells");
- Conjunction and disjunction: two or more concept names sharing one head noun (e.g., "91 and 84 kDa proteins" refers to "91 kDa protein" and "84 kDa protein");
- Nested names: one name may occur within a longer name, as well as occur independently (e.g., "T cell" is nested within "nuclear factor of activated T cells family protein");
- Names of newly discovered concepts: there is an overwhelming growth rate and constant discovery of novel biomedical concepts, which takes time to register in curated nomenclatures.

Biomedical concepts from the whole spectrum of biomedical knowledge are of interest for being automatically extracted from scientific articles, in order to build a rich and reliable information profile. The following biomedical concepts are typically the ones of more interest, due to their implications and inherent interactions and relations:

- Species or Organism: e.g., "mouse" and "human";
- Gene or protein: e.g., "BRCA1" and "breast cancer type 1 susceptibility protein";
- Enzyme: e.g., "lactase", "catalase" and "amylase";
- Mutation: e.g., "c.1517A>G" and "Asp506Gly";
- Drug: e.g., "phenformin" and "methazolamide";
- Chemical: e.g., "chow" and "water";
- Anatomy: e.g., "cervix" and "endothelium";
- Disorder: e.g., "alzheimer's disease" and "parkinson's disease";

- Pathway: e.g., “photosynthesis” and “histidine metabolism”;
- Biological process: e.g., “aging” and “circadian regulation”;
- Molecular function: e.g., “carbonic anhydrase” and “annealing”.

Biomedical concept recognition can be decomposed in two different steps: NER, and normalization and disambiguation, in order to recognize the names and associate the correct unique identifiers, respectively. Thus, different techniques may be applied to perform each step, which we will describe and analyze in detail.

2.2.1 Resources

Knowledge bases

Various agencies created standards for concept names definition and applicability on real life tasks, in order to provide unique and centralized resources and promote their linkage with patient health records and research laboratory resources. For instance, the International Classification of Diseases (ICD) is used to classify diseases and other health problems recorded on many types of health and vital records, namely death certificates and health records. On the other hand, Anatomical Therapeutic Chemical (ATC) is the World Health Organization (WHO) pharmaceutical coding system that divides drugs into different groups, according to the organ or system on which they act and their therapeutic and chemical characteristics. Despite the fact that standardization processes have been successfully applied on some concept types, most key biological concepts still lack standards for careful names definition, centralized storage and integration on daily tasks. Moreover, there is no single resource that includes all variant names of a specific concept. Thus, it is fundamental to combine available knowledge bases to collect as much information as possible regarding a specific concept type. For instance, Tsuruoka et al. [65] concluded that gene and protein databases contain on average 5-14 names for each identifier, which reflects the importance of aggregating as much information in a single resource.

Table 2.1 presents databases and ontologies publicly available, which may contain relevant data for biomedical concept recognition.

Table 2.1: Biomedical databases and ontologies.

	Name	Concept(s)
Databases	Entrez Gene [66]	• Gene
	HUGO Gene Nomenclature Committee (HGNC) [67]	• Gene
	GenBank [66]	• Sequence
	dbSNP [68]	• Genetic variation

	Uniprot [46]	• Protein
	Protein Data Bank (PDB) [69]	• Protein
	Expert Protein Analysis System (ExPASy) [70]	• Enzyme
	ChemIDplus [71]	• Chemical
	Human Metabolome Database (HMDB) [72]	• Small molecules
	DrugBank [73]	• Drug
	Pharmacogenomics Knowledge Base (PharmGKB) [74]	• Gene • Drug • Disease
	RxNorm [75]	• Drug
	Kyoto Encyclopedia of Genes and Genomes (KEGG) [76]	• Pathway
	BioSystems [77]	• Pathway
	Online Mendelian Inheritance in Man (OMIM) [78]	• Disease • Variation • Gene
	Systematized Nomenclature of Medicine (SNOMED) [79]	• Anatomy • Morphology • Species • Chemical • Drug • Disease • Diagnosis • Procedure • Physical agents, forces, activities • Social context
	Medical Subject Headings (MeSH) [80]	• Protein • Chemical • Disease
	Comparative Toxicogenomics Database (CTD) [81]	• Gene • Chemical • Disease • Pathway

	Medical Dictionary for Regulatory Activities (MedRA) [82]	• Disease
Ontologies	Chemical Entities of Biological Interest (ChEBI) [83]	• Chemical
	Cell Ontology (CL) [84]	• Cell
	GO [47]	• Gene
	Protein Ontology (PRO) [85]	• Protein
	Sequence Ontology (SO) [86]	• Sequence
	Disease Ontology (DO)	• Disease
	National Center for Biotechnology Information (NCBI) taxonomy [87]	• Species
	Common Anatomy Reference Ontology (CARO) [88]	• Anatomy
	Unified Medical Language System (UMLS) semantic network [89]	• Species • Anatomy • Chemical • Biological function • Physical object • Idea or concept

Corpora

Table 2.2 presents a list of relevant corpora for biomedical concept recognition, considering the source of annotations, target concepts and availability of unique identifiers from known knowledge bases. As we can see, most of the research efforts have been on the recognition of gene and protein names, with various corpora containing several thousands of annotated sentences. Such effort is a consequence of two different factors: the importance of genes and proteins on the biomedical domain, and the high variability and lack of standardization of names. Various challenges were organized for the recognition of gene and protein names, such as BioCreative [90] and JNLPBA [91]. The SCAI IUPAC corpus is also a good example of a specific sub-entity type corpus, containing only annotations of chemicals that follow the IUPAC nomenclature. Finally, BioCreative CHEMDNER, AnEM and CellFinder are very recent corpora, showing that the development of manually annotated corpora for the various entity types is still an ongoing work. Overall, we can see that there is a significant difference on the amount of available gold and silver standard corpora. Only the CALBC corpus provides silver standard annotations, with more than 700 thousand abstracts with more than 10 million annotations obtained through a majority voting strategy to harmonize

annotations from systems that contributed to the project. Moreover, as expected, there is a significant difference in the amount of provided information between silver and gold standard corpora. The largest GSC provides only around 90 thousand sentences. This difference is also observed when comparing corpora with and without identifiers information, where typically corpora without identifiers provide a higher amount of annotated sentences. Moreover, there is a clear recent trend on corpora with full-text articles and heterogenous concept types, which reflects the progress of the field, with powerful solutions capable to process large amounts of documents annotating multiple concept types.

Table 2.2: List or relevant corpora for biomedical concept recognition.

Corpus	Year	Type	Concepts	IDs	Granularity	Size*
GENETAG [92]	2005	Gold	• Gene and protein	✗	Sentences	20000
JNLPBA [91]	2004	Gold	• Gene and protein	✗	Abstracts	22402
FSUPRGE [93]	2008	Gold	• Gene and protein	✗	Abstracts	≈29447
PennBioIE [94]	2004	Gold	• Gene and protein	✗	Abstracts	≈22877
BioCreative II GN [95]	2008	Gold	• Gene and protein	✓	Abstracts	≈2529
BioCreative III GN [38]	2011	Gold	• Gene and protein	✓	Full texts	≈272439
OrganismTagger [96]	2011	Gold	• Species	✓	Full texts	9863
Linnaeus [97]	2010	Gold	• Species	✓	Full texts	19491
SCAI Disease [98]	2010	Gold	• Disorders	✗	Abstracts	≈3640
EBI Disease [99]	2008	Gold	• Disorders	✓	Sentences	600
Arizona Disease [100]	2009	Gold	• Disorders	✓	Sentences	2500
BioText [101]	2004	Gold	• Disorders	✗	Abstracts	3655
SCAI IUPAC [102]	2008	Gold	• Chemical	✗	Sentences	20300
SCAI General [103]	2008	Gold	• Chemical	✗	Sentences	914
BioCreative CHEMDNER [104]	2013	Gold	• Chemical	✗	Abstracts	≈90000
AnEM [105]	2012	Gold	• Anatomy	✗	Sentences	4700
CellFinder [106]	2012	Gold	• Gene and protein • Species • Anatomy • Cell components • Cell line • Cell type	✗	Full texts	2100

CRAFT [107]	2012	Gold	• Gene and Protein • Species • Chemical • Cell • Biological processes • Molecular functions • Cellular components	✓	Full texts	21000
CALBC [108]	2010	Silver	• Gene and Protein • Species • Disorders	✓	Abstracts	≈6428547

*Size is provided considering the number of sentences. Approximate values assume that each MEDLINE abstract contains on average 7.2±1.9 sentences [109]. We considered the best-case scenario with ≈9 sentences.

2.2.2 Named entity recognition

The goal of NER is to identify chunks of text that refer to names of specific concepts of interest. Such recognition can be performed following different approaches, which can be categorized as being based on rules, dictionary matching or ML. The development of such divergent solutions is composed of various complex steps that are part of different processing pipelines, but can be generalized in a common workflow. Figure 2.4 presents that general processing pipeline, which is composed by the following resources and steps:

- Corpus: collection of related text documents;
- Pre-processing: perform NLP to simplify and enable automatic recognition process;
- NER: automatically recognize specific concept names;
- Post-processing: refinement of already recognized concept names;
- Annotated corpus: input documents containing recognized concept names.

Even though different approaches follow a similar processing pipeline, each one fulfills different requirements, depending on the linguistic characteristics of the concepts being identified. Such heterogeneity is a consequence of the predefined naming standards and how faithfully the biomedical community followed them. Thus, it is recommended to take advantage of the approaches that better fulfill the requirements of each concept type:

- Rule-based: names with a strongly defined orthographic and morphological structure (e.g., gene variants);
- Dictionary-based: closely defined vocabulary of names (e.g., species);
- ML-based: strong variability and highly dynamic vocabulary of names (e.g., genes, proteins and chemicals).

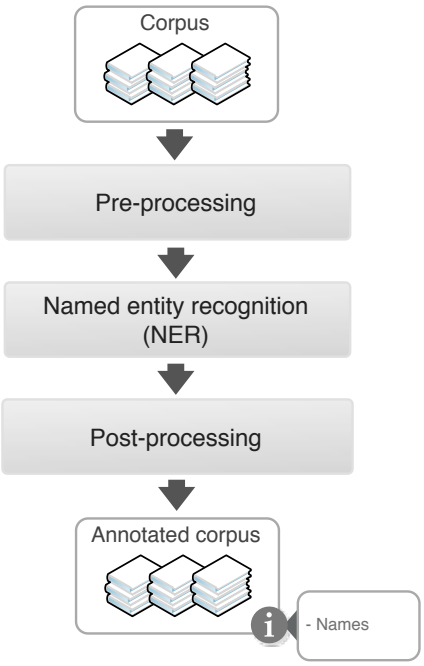


Figure 2.4: General processing pipeline of biomedical NER solutions.

Applying the best approaches is not possible in all cases, since each approach presents different technical requirements.

Rule-based

Rule-based systems rely on a set of rules specified by experts, combining orthographic characteristics with word syntactic and semantic properties. In addition to the required human efforts and resources, the generated rules are too specific, being focused on recognizing entity names on a specific corpus. Generally, when these rules are used in a different context, the overall performance falls and fewer concepts are correctly recognized. Thus, rule-based approaches are recommended to the recognition of strictly defined and standardized concept names. Nevertheless, as one of the first approaches for NER, various researchers explored the potential of this approach to recognized biomedical concept names from scientific documents. For instance, Fukuda et al. [110] presents PROPER, a system that uses surface clues (e.g., capital letters, symbols, and digits) to extract candidates of protein names. On the other hand, PASTA [111] uses a mixture of manually generated and automatically generated rules to build twelve different orthographic templates, one per concept type.

Dictionary-based

Dictionaries are large collections of names that intend to include all names regarding a specific concept. In this approach, a match between the dictionary entries and the unstruc-

tured text is accomplished, in order to correctly recognize the concept names of a predefined context. However, dictionary-based approaches have two main limitations: *a*) large number of false positives caused by concepts with short names; and *b*) existence of spelling variations that are not available in curated resources. The first limitation can be minimized by removing short names from the dictionary, however such concepts will never be identified from texts. Since most times these short names are abbreviations, they can be recognized by applying abbreviation resolution techniques [112]. On the other hand, the second limitation can be slightly overcome by applying approximate string matching techniques (e.g., Levenshtein Distance [113], Jaro-Winkler Measure [114] and SoftTFIDF [115]), which should be applied carefully, since a too relaxed matching may also deliver a large amount of false positives.

In the development of dictionary-based approaches, one must perform the matching between the entries of a well curated and complete dictionary with chunks of text. Afterwards, the successful matches will be precisely related with unique identifiers from curated knowledge bases. Figure 2.5 presents the core components used and tasks performed on dictionary-based approaches, illustrating the relations between them:

- Terminology resources: domain knowledge;
- Dictionaries: a combination of several databases to collect the maximum number of entity names and identifiers as possible;
- Pre-processing: perform tasks on natural language texts and the dictionary to simplify the recognition process;
- String searching and matching: perform string matching between dictionary's entries and text;
- Post-processing: refinement of already matched names, resolution of abbreviations and exploitation of multiple occurrences of the same entity within the text;
- Annotated corpus: recognized names with respective identifiers.

Terminology resources The dictionary is the core component of these approaches, since the match with the text is performed using the concept names contained in the dictionary, which makes its creation one of the most important steps. Table 2.3 presents an example of a dictionary of Cell names, where each identifier has various corresponding names. Thus, both "acanthocytes" and "spur cell" are recognized with the identifier "UMLS:C0000886:T025:CELL", therefore pointing to the same concept.

Combining databases and ontologies in an unique dictionary is not a straightforward task, since each resource uses its own unique identifiers. However, various research works have already combined a wide set of resources to build comprehensive terminological resources (Table 2.4). For instance, BioThesaurus [116] maps a large collection of gene and protein names to protein entries in UniProt. It allows retrieving synonymous names of a specific

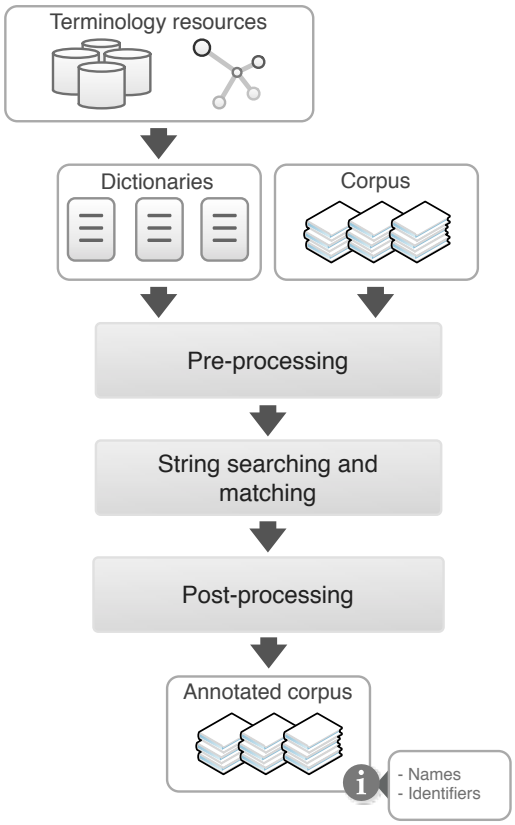


Figure 2.5: General processing pipeline of dictionary-based NER solutions.

Table 2.3: Sample of a dictionary of Cell concept names, using UMLS as the curated knowledge base.

Identifier	Names
UMLS:C0000886:T025:CELL	• acanthocytes • acanthocyte • acanthrocyte • spur cell • cells spur • crenated cell • spiny prickle cell
UMLS:C0001280:T025:CELL	• armed macrophage • activated macrophage • activating macrophage
UMLS:C0002449:T025:CELL	• ameloblast

Table 2.4: List of relevant biomedical terminology aggregators.

Name	Concept(s)
BioThesaurus [116]	• Gene • Protein
BioLexicon [117]	• Gene • Protein • Enzyme • Chemical • Disorder • Species
UMLS metathesaurus [118]	• Activities and behaviors • Anatomy • Chemicals and drugs • Concepts and ideas • Devices • Disorders • Genes • Geographic areas • Species • Objects • Occupations • Organizations • Phenomena • Physiology • Procedures
Genomic Name Server (GeNS) [119]	• Gene • Protein • Enzyme • Species • Drug • Disease • Pathway
Jochem [120]	• Chemical

protein and identify ambiguous names shared by multiple proteins. In 2013, BioThesaurus covered more than 2 million proteins, resulting in more than 2.8 million names extracted from multiple biological databases. On the other hand, BioLexicon [15] combines terminology focused on several biomedical concepts, namely gene and protein, enzyme, chemical, disorder and species. Besides combining several resources, it also augments the collected terms with new variants automatically extracted from biomedical literature. In 2013, BioLexicon covered 2.2 million biomedical concepts with 4 million names variants.

Despite the wide set of resources described above, there are variant names that will not be included in the combined dictionary. Cohen et al. [121] concluded that many names are simple orthographic variants of each other, and that most of these variants can be generated using simple rules, such as [122]:

- replace internal spaces by hyphens or vice-versa (e.g., "IL 10" to "IL-10");
- remove internal spaces or hyphens (e.g., "NF-kappa B" to "NFkappaB");
- add an hyphen between a letter and digit (e.g., "NFXL1" to "NFXL-1");
- if the term ends on hyphen and digit, replace the digit by the Roman equivalent (e.g., "NFXL-1" to "NFXL-I").

Schuemie et al. [123] presents a complete and detailed survey on spelling variation rules for gene and protein names, describing and studying the impact of almost 20 rules considering four different species. There are also tools developed specifically to generate names variants. For instance, Lexical Variant Generation (LVG) [124] is a complete and highly configurable solution that generates variant names based on more than 60 different rules. Since generating names variants based on general rules may result in a large amount of nonsensical or ambiguous names, it is important to apply a post-processing step to remove such problematic names. For instance, Hanisch et al. [125] removed names from the dictionary by applying regular expressions pre-defined by experts, which represent patterns of unspecific synonyms (e.g., only non-descriptive tokens).

String Matching The previously exposed problems related to the recognition of biomedical concept names demand the application of sophisticated methods to compare dictionary entries with natural language texts. Such solutions can be categorized as being: exact, approximate (also known as fuzzy) and ML-based. All approaches can be applied considering case sensitive or insensitive matching, in order to consider terms with the same letters in different case forms as equal. That way, "BRCA1" and "brca1" are considered the same term if case insensitive matching is applied.

Exact string matching finds names in text that are exactly the same as in the dictionary. On the other hand, approximate string matching approaches allow differences between the text and the entry in the dictionary. Three different approximate matching approaches can be applied, based on measures, rules and ML. Measure-based approaches calculate a value that reflects the similarity between two names (the name in the dictionary and the chunk of text), and only matches with a similarity value higher than a pre-defined threshold will be accepted as concept names. Levenshtein distance (also known as edit distance) [113], Jaro-Winkler [114] and SoftTFIDF [115] are examples of measure-based approaches. Tsuruoka et al. [65] shows that SoftTFIDF is the non-ML based solution that provides the best performing results, since it considers domain knowledge to calculate similarity measures. On the other hand, rule-based approaches define specific rules for accepting and rejecting each

concept name. For instance, Rebholz-Schuhmann et al. [126] successfully applied this idea using a regular expression for each concept name. For instante, the regular expression "BRCA[-][1I]" accepts "BRCA 1", "BRCA I", "BRCA-1" and "BRCA-I" as names. Finally, ML-based approaches are also used to induce string similarity from actual examples of string pairs. For instance, Tsuruoka et al. [65] used a logistic regression approach to learn string similarity measures from a dictionary of gene and protein names. In the end, the author showed that such approach outperforms previous approximate matching techniques. Cohen et al. [115] presents a throughout study regarding the applicability of string matching approaches, comparing exact and approximate matching solutions used ah-doc and combined with other measures. The authors also confirm SoftTFIDF as the best non-ML based solution. However, the best approach is a combination of Levenstein distance with Jaro-Winkler, with the respective scores being adaptively combined by a machine learning technique. In the end, the authors made their findings publicly available through the SecondString[8] tool, which provides methods to perform exact and approximate string matching. Overall, approximate string matching solutions have better performance results when compared against exact matching. However, with a robust dictionary and an effective spelling variation generation procedure, exact matching can also achieve excellent performance results, as shown by Fundel et al. [127].

String Searching During the matching process, it is necessary to perform a complete match between the text and the entries in the dictionary. Considering a dictionary with thousands of entries and a text with thousands of chunks to be matched, if this process is performed in a brute force manner, it would take a large amount of time to be excuted, becoming completely impracticable. In order to solve this problem, the basic idea is to organize the several strings on a structure that will streamline the searching process, establishing relations between sub-strings that are common to specific strings. Thus, to find a specific string, it becomes necessary to navigate through the several sub-strings to find the desired object. On each step of this navigation, the strings that will not match with the searched string are discarded, removing the need to perform comparisons with all entries in the dictionary. Trie [128], Suffix Array [129] and PATRICIA Tree [130] are examples of techniques that apply this idea. Other approaches use Hash functions to improve the searching procedure, for instance by producing a hash of the first word of each entry, and then perform the search using the generated hash. Those solutions were originally created for exact string matching, however they were extended to support approximate matching too. For instance Shang et al. [131] present a solution for approximate string matching with Tries.

Deterministic Finite Automaton (DFA) [132] is another method that can be applied to perform string matching and searching. In a simplistic way, DFAs are finite state machines that accept or reject finite strings of symbols. Thus, a DFA transits from one state to another,

[8] `http://secondstring.sourceforge.net`

depending on the sequence of input symbols, and a string is accepted if its parsing finishes in a state marked as final. Considering that each input string of symbols is a name from the dictionary, one can build a DFA to match all names in a dictionary. Since each entry of the DFA is a regular expression, approximate string matching can be also performed, by considering orthographic variants on input dictionary names. For instance, "BRCA1", "BRCA 1" and "BRCA-1" names variants can be recognized through the regular expression "BRCA[-]{0,1}1".

Post-Processing After performing the recognition step using dictionaries, there may be some entity names that have not been recognized with success. For instance, authors of scientific documents often introduce abbreviations of entities by using a format similar to "antilymphocyte globulin (ALG)" or "ALG (antilymphocyte globulin)". Due to the small number of characters, abbreviations are normally discarded during the matching process, otherwise a large amount of false recognitions would be generated. Thus, it is common to perform an abbreviation resolution step, which can be accomplished using a simple algorithm [112] with high degree of accuracy, followed by additional processing to ensure that both mentions are recognized.

During the matching process, it is normal that a match associates a name with multiple entity identifiers. This occurs when the dictionary contains very similar names that refer to distinct concept identifiers, or even when an unique name is related with multiple identifiers, varying with the context (e.g., different species). To solve this problem, a disambiguation step is performed, to resolve ambiguous names to the correct unique identifiers. This subject is further discussed in the normalization section.

Machine learning-based

ML-based approaches use methods to learn how to recognize specific entity names. The learning procedure uses texts that contain concept names annotated by experts. This approach tries to solve the dictionary-based problems, recognizing new spelling variations of a concept name. However, ML does not provide direct identifier information of recognized concepts from curated resources. This problem can be solved using a dictionary in an extra step, in order to link the recognized names to the corresponding entries in the dictionary.

With ML approaches, it is necessary to train a computational model to induce the characteristics of specific entity names. After training the model, the system is prepared to be applied to non-annotated texts, predicting the chunks of text that are entity names. Figure 2.6 presents the core components and tasks that are used and performed on machine learning approaches, illustrating the relations between them (the steps that are not described here have the same role as in other approaches):

- Pre-processing: input text processing and classes representation;

- Feature extraction and selection: extract, select and/or induce features from the tokens, in order to be used by the model to predict entity names;
- Model: induce a set of rules that describe and distinguish data classes or concepts;
- Post-processing: refinement of already matched names and resolution of abbreviations.

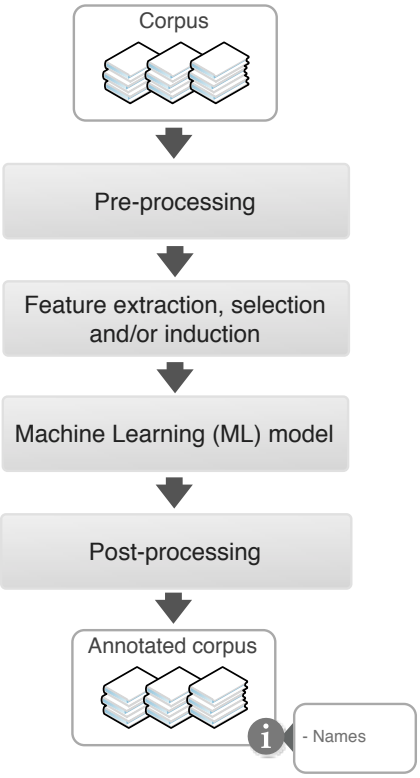


Figure 2.6: General processing pipeline of ML-based NER solutions.

Pre-processing Apart from the pre-processing steps previously described, in order to identify if each token is part or not of an entity name, it is necessary to use an encoding method that will give a tag to each token of the text. Such tags are used as classes of the ML classifiers. The simplest encoding is the IO encoding, which tags each token as either being in (tag "I") a particular named entity or outside any entity (tag "O"). This encoding is defective because it cannot represent two entities next to each other, since there is no boundary tag. The BIO encoding is the *de facto* standard, and it extends the IO encoding solving the boundary problem. It subdivides the "in" tags as either being the beginning of the entity (tag "B") or the continuation of the entity (tag "I"). The BMEWO encoding extends the BIO encoding by distinguishing the end of an entity (tag "E") tokens from the middle entity tokens (tag "M"), and adding a new tag ("W") for entities with only one token. Finally, the BMEWO+ encoding extends the BMEWO encoding by adding a local contextual behavior. Thus, if a

gene name is in the previous or following token, a string "GENE" should be concatenated to the tag of the current token. Table 2.5 presents an example of the application of the several encoding methods on a sample sentence.

Table 2.5: Class specification of the sentence "Gamma glutamyl transpeptidase (GGTP) activity in the seminal fluid".

Sentence	IO	BIO	BMEWO	BMEWO+
Gamma	I	B	B	B_GENE
glutamyl	I	I	M	M_GENE
transpeptidase	I	I	E	E_GENE
(	O	O	O	GENE_O_GENE
GGTP	I	B	W	W_GENE
)	O	O	O	GENE_O
activity	O	O	O	O
in	O	O	O	O
the	O	O	O	O
seminal	O	O	O	O
fluid	O	O	O	O

Feature extraction The features are the input of the ML model, which uses them to predict if a specific chunk of text is an entity name or not. In text mining, we need to extract these features from text, in order to precisely describe the input text and reflect the special phenomena and linguistic characteristics of the naming conventions. The final goal is to identify only the features that provide a positive contribution, contributing to an increase of performance. Thus, feature extraction intends to extract features from texts using previously defined rules and/or external resources. In the end of the feature extraction process, as the input to the classifier, the features must be represented in the form of a vector. So, each feature should assume the value "1" if it is present on the current token or "0" if it is not (Table 2.6).

In order to properly define the feature extraction rules, it is fundamental to understand how the chosen tokenizer works, since features are extracted considering the generated tokens. For instance, if the tokenizer splits words separated by hyphens (e.g., tokenization of "nf-kappa" results in three tokens: "nf", "-" and "kappa"), it does not make sense to define a feature that will tag words containing hyphens. The set of extracted features can be divided into three distinct groups, which contain categories of features:

- Internal features: use heuristics and methods to extract characteristics of the text (e.g.,

Table 2.6: Illustration of the matrix of features as the input of the machine learning technique. Each vector defines the features present on an instance.

	Feature 1	**Feature 2**	**...**	**Feature m**
Instance 1	1	0	...	0
Instance 2	0	1	...	0
⋮	⋮	⋮	⋱	⋮
Instance n	0	0	...	1

feature that describes if a word is capitalized or not);

  - Linguistic: based on linguistic parsing, such as lemmas, chunks and POS tags;
  - Orthographic: to capture word formation, such as "AllCaps" to indicate that all characters in a token are capitalized, or "InitCap" to indicate that a token starts with a capitalized character;
  - Morphological: capture common patterns between different tokens, such as suffixes and prefixes.

- External features: use external resources to provide domain knowledge to the recognition process.
  - Dictionaries: match the text with dictionaries and provide the result as features;
- Local context: establish relations between features and/or tokens to model local context.
  - Window: add all or filtered features of preceding and/or succeeding tokens as features of the current token;
  - Conjunctions: create new features by grouping together features of the surrounding tokens.

Detailed description of specific feature types is presented on further chapters, exposing its detailed application when appropriate.

Feature selection After the feature extraction process, a large amount of features may be generated, which may affect the model training speed and achieved performance results:

- Overfitting: the large amount of features may contribute to a model highly optimized to the training data. Thus, when the generated model is applied to a different data set, its performance drops dramatically;
- Curse of dimensionality: a large amount of features may generate a sparse feature space, which hinders obtaining a valid and optimized statistical model with high performance results.

In order to minimize those problems, it is important to select only the features that provide useful information for the recognition process, applying a filtering step. This technique is called Feature Selection, and it provides several advantages [133]: *a*) avoids overfitting, curse of dimensionality and improves the model's performance; *b*) provides faster and more cost-effective models; and *c*) provides a deeper insight into the underlying processes that generated the data. However, the advantages of feature selection come at a certain price, as the search for a subset of relevant features introduces an additional layer of complexity in the modeling task. There are works that already demonstrated the positive effect of feature selection. For instance, Hakenberg et al. [134] showed that after removing 95% of the features, the prediction quality of concept names was practically not affected, and the time necessary to train the model dropped dramatically.

Different methodologies to apply feature selection have been proposed. Those techniques can be organized into three categories, depending on how they combine the feature selection search with the construction of the model [133]:

- Filter: obtain the features' relevance by looking only at the intrinsic properties of the data. Those techniques treat the problem of finding a good feature subset independently of the model;
- Wrapper: the model is used to find the best feature subset, by training and testing the subset of features using the specific considered model;
- Embedded: the search for the optimal subset of features is embedded in the classifier.

Figure 2.7 lists the overall advantages and limitations of the various feature selection approaches, presenting examples of the most influential methods.

Feature induction The features described so far are based on manually defined rules, providing significant and valuable information to perform NER. However, it is possible that some informative features are not extracted by the defined rules. Feature induction intends to minimize this problem, by creating new features that provide information and characteristics not previously covered. New features are created by building conjunctions of previously defined atomic features. For instance, if a token contains the features "CHUNK=NP" and "LEMMA=regulation", a new conjunction feature "CHUNK=NP_@_LEMMA=regulation" may be created to reflect the relevant association between the two atomic features. Thus, feature induction works by iteratively considering sets of candidate atomic and conjunction features created from the initially defined set of atomic features. Only candidates that provide useful information are included in the final set of features. Intuitively, features with high gain provide strong evidence for many decisions. As a consequence of this process, feature induction also prevents over fitting by not considering features that do not provide useful information. This technique is deeply related to the used model, because the information provided by each feature varies from model to model. Della Pietra et al. [135] proposed an

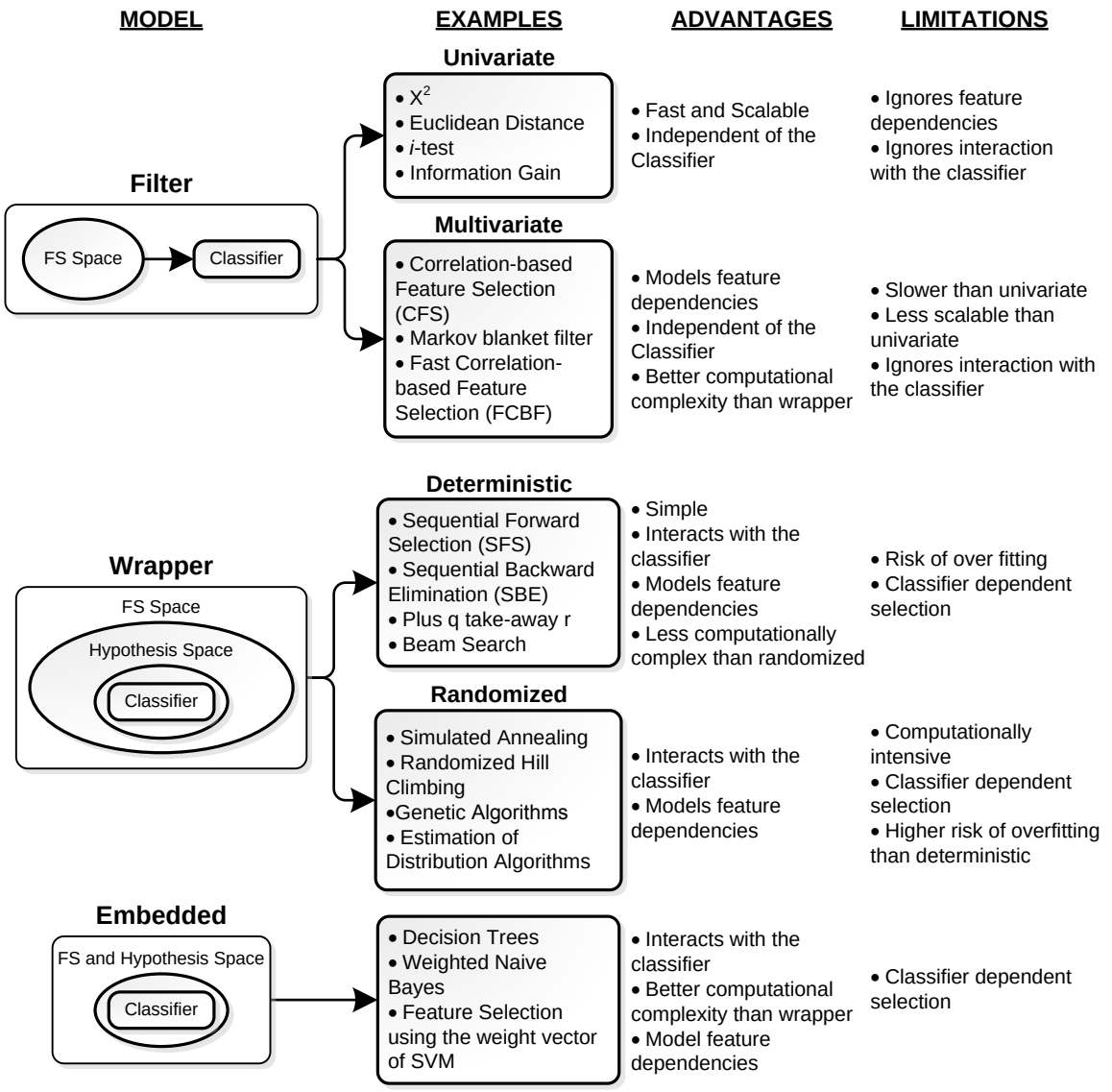


Figure 2.7: Different approaches to implement feature selection, presenting examples, advantages and limitations of each (based on [133]).

efficient algorithm to search for features that can effectively increase the models' performance. Various NER systems take advantage of feature induction [136–138]. McDonald and Pereira [138] also demonstrate the high positive contribution of using typical feature extraction and induction at the same time.

ML model ML methods work by building a feature-based statistical representation of target meaning from training data, in order to describe the seen information in a meaningful way and develop an appropriate response to unseen data. The structure that supports storing such decision framework is known as model, which can be represented and obtained in

several different manners. Overall, a ML model must be: *a*) Descriptive (capture information from training data); *b*) Predictive (generalize to unseen data); and *c*) Explanatory (provide plausible and informative description of the learned information). Research has demonstrated that it is extremely fruitful to model the behavior of complex systems by taking advantage of automatic and self-learning algorithms. Also, probabilistic models often show better performance and robustness against categorical models. Accordingly, several probabilistic models have been shown to be especially useful for extracting meaning from natural language texts, such models can be classified as being:

- Generative: model the distribution of individual classes. Naïve Bayes (NB) and Hidden Markov Models (HMMs) are examples of such models;
- Discriminative: learn the boundaries between classes. SVMs and CRFs are examples of such models.

For instance, considering the problem of identifying the language that someone is speaking, a generative model will learn each language and determine the language associated with the speech. On the other hand, discriminative models determine the linguistic differences without learning any language. Thus, assuming that we have an input sequence of observations (represented by X) and classes that need to be inferred from the given observations (represented by Y), generative models calculate $P(x|y)$, i.e., the probability of the observed data given the target classes, and discriminative models calculate $P(y|x)$, i.e., the probability of each class considering the observed data.

Depending on the used data, labelled and/or unlabeled, the learning process of ML models can be classified as being:

- Supervised learning: use labelled data to generate a function that maps inputs to desired outputs;
- Semi-supervised learning: combines both labelled and unlabeled data to generate an appropriate function that maps inputs to desired outputs;
- Unsupervised learning: apply appropriate functions to infer patterns from unlabeled data.

Several solutions were created to solve the challenges imposed by supervised learning. CRF is one of the models with more research interest for sequence labeling, since it presents several advantages over other methods. Firstly, CRFs avoid the label bias problem [139], a weakness of Maximum Entropy Markov Models (MEMMs). Additionally, the conditional nature of CRFs (a discriminative model) relaxes strong independence assumptions required to learn the parameters of a generative model, such as HMMs [140]. Moreover, CRFs outperformed both MEMMs and HMMs on a number of real-world sequence labeling tasks [139]. Finally, SVMs follow a different approach and have been shown to deliver comparable results to CRFs [141] using identical feature functions. However, training complex SVM models for NER may

take more time [142, 143]. Both SVMs and CRFs are widely used, achieving high performance results on problems with heterogeneous characteristics. Thus, it is important to understand in detail how each of those algorithms work in order to provide such reliable results.

SVMs were first introduced by Cherkassky [144], being defined for the classification of binary problems, i.e., linearly separable classes of objects (Figure 2.8). Thus, considering objects that belong to two classes (circle or triangle in Figure 2.8), it is considerably straightforward to draw a line that separates them. Such separating hyperplane can be written as $W.X + b = 0$, where W is a weight vector, i.e., $W = (w_1, w_2, ..., w_n)$, n is the number of attributes and b is a scalar that allows to increase the margin. The aim is to find the maximum margin, obtaining the support vectors and parallel hyperplanes (to the optimal hyperplane) that are closest to these support vectors in either class. If the training data is linearly separable, the selected hyperplanes should not have objects between them and should have its distance maximized (Figure 2.8).

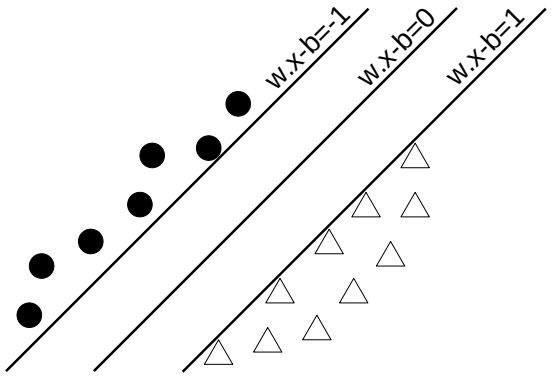


Figure 2.8: SVM margins illustration (based on [145]).

SVMs can also be used to separate classes that cannot be separated linearly (Figure 2.9, left). In such cases, the coordinates of the objects are mapped into a multi-dimensional feature space using non-linear functions, where the two classes can be separated with a linear classifier [145] (Figure 2.9, right). The non-linear mapping generated by the feature functions is computed with special non-linear functions called kernels [145]. There are many different types of kernels, which achieve better results depending on the problems requirements, i.e., how easily objects of two classes can be separated in the multi-dimensional space.

CRFs were first introduced by Lafferty et al. [139]. Assuming that we have an input sequence of observations (represented by X), and a state variable that needs to be inferred from the given observations (represented by Y), a CRF can be defined as "a form of undirected graphical model that defines a single log-linear distribution over label sequences (Y) given a particular observation sequence (X)" (Figure 2.10) [140]. This layout makes it possible to have efficient algorithms to train models, in order to learn conditional distributions between Y_j and feature functions from the observable data. To accomplish this, it is necessary to

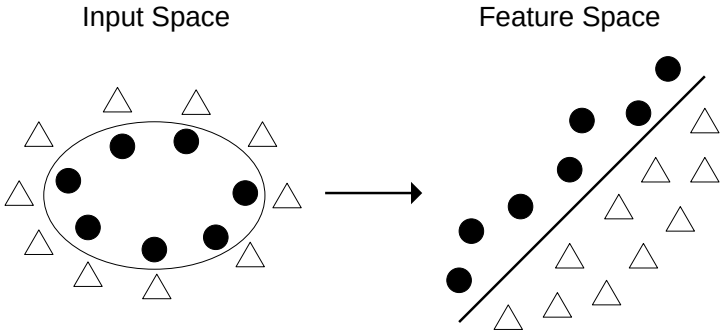


Figure 2.9: SVM kernel illustration (based on [145]).

determine the probability of a given label sequence Y given X. First, the model assigns a numerical weight to each feature, and then those weights are combined to determine the probability of Y_j. Such probability is calculated as follows:

$$p(y|x,\lambda) = \frac{1}{Z(x)} exp(\sum_j \lambda_j F_j(y,x)), \tag{2.6}$$

where λ_j is a parameter to be estimated from training data and indicates the informativeness of the respective feature, $Z(x)$ is a normalization factor and $F_j(y,x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i)$, where each $f_j(y_{i-1}, y_i, x, i)$ is either a state function $s(y_{i-1}, y_i, x, i)$ or a transition function $t(y_{i-1}, y_i, x, i)$ [140].

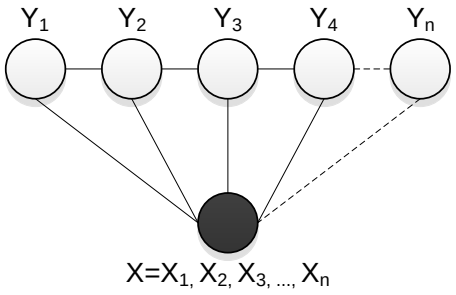


Figure 2.10: Graphical structure of CRFs for sequences. The variables corresponding to dark nodes are not generated by the model (based on [140]).

When considering higher-order models, each label depends on a specific number of o previous labels. Thus, the probability will consider not only the previous observation and its features, but o-previous observations and features, which better models dependencies and may provide improved results, depending on the target data and task. However, the training complexity of higher-order models increases exponentially with the pre-defined order o [146].

There is also a high research interest on using unlabeled data to improve the performance of supervised learning techniques, since unlabeled data is easy to obtain. This problem is

often referred as semi-supervised learning. There are several solutions to deal with labelled and unlabeled data at the same time, four of the solutions with more impact will be analyzed: self-training, co-training, expectation maximization and feature-based. Note that traditional supervised models have been extended to use information from unlabeled data, such as semi-supervised CRF [147, 148] and semi-supervised SVM [149]. The main idea of self-training is to retrain a model on its own labeled data on each round [150]. Thus, a classifier is trained on the available training data and used to label unlabeled examples. Afterwards, instances from the unlabeled data with a prediction confidence above a pre-defined threshold are added to the training set. This process is repeated for a number of predefined iterations. Such technique was previously applied in a number of NER solutions [151, 152], with some positive outcomes. On the other hand, co-training [153] takes advantage of two different views of the same data, using different and complementary feature sets to describe the same instance. That way, a classifier is trained on each view of the data considering the labelled data, and only the most confident predictions of each classifier on the unlabeled data are used to generate additional training data. Co-training has been used in a number of NER research works [154, 155]. A related idea is to use Expectation Maximization (EM) [156], in order to iteratively compute the Maximum Likelihood in the presence of missing or hidden data [157]. The goal of Maximum Likelihood is to estimate the model parameter(s) for which the observed data is the most likely. However, this method cannot deal with large amounts of labelled data, suffering a serious drop of performance [158]. A different and recent approach, which has been quite successful [159, 160], pre-processes the unlabeled data to extract features and then uses these features in a supervised model. Examples of features are POS tagging, word clustering (categorize the words of the text assuming that two words are similar if they appear in similar contexts or that they are exchangeable to some extent) and mutual information (standard measure of the strength of association between co-occurring items). An example of success of this approach is the system presented by Ando [161], which obtained the best performance in the BioCreative II Gene Mention task [90]. This system uses the Alternating Structure Optimization (ASO) [162] technique to create new additional features from standard features on the unlabeled data. In the end, the classifiers are trained with labelled data using the standard features and the new features learned from unlabeled data. They concluded that using features from unlabeled data minimizes the problem of unknown words, caused by the lack of labelled training data.

Since different models have different characteristics and feature sets, each one encodes the same knowledge through different techniques. Thus, in order to take advantage of the positive contributions of different models, it is common to combine them in a unique system. To accomplish this, it is necessary to use techniques to combine the results from the several models, such as:

- Union: use the results of the several models. For instance, Kuo et al. [136] use this

technique on their system;
- Intersection: use only the results that are common to the distinct models. For instance, Kuo et al. [136] use this technique on their system;
- Majority voting: each model contributes with one vote, which could have a specific weight depending on the needs. In the end, the class with more votes should be selected. For instance, Zhou et al. [163] use this combination strategy on an ensemble of two HMMs and one SVM.
- Machine learning: train a machine learning model to induce the final class from the results of the several models. For instance, Mika and Rost [164] implement a SVM to combine the results of three SVMs and one dictionary matching.

Post-processing On ML-based approaches, it is also necessary to perform post-processing techniques, in order to remove recognition errors and recognize more entity names. Identically to dictionary-based approaches, it is also necessary to perform abbreviation resolution, in order to extend recognized acronyms of entity's names. Additionally, machine learning recognition generates several errors that could be easily corrected using simple rules or methods:

- a single punctuation mark (parenthesis, bracket or quotation mark) on a recognized entity name clearly demonstrates that the labeling engine has made a mistake [165];
- extend incomplete names recognized by the machine learning procedure (e.g., only "p53" in "p53 mutant" was recognized) [166]. To accomplish this idea, a dictionary lookup solution may be used.
- remove stop words, e.g., "by" and "or" that have been wrongly recognized as part of recognized names [166];
- other errors identified in the specific problem, which are dependent on the used tokenizer, machine learning model and corpus.

2.2.3 Normalization and disambiguation

The goal of normalization is to associate each identified chunk of text with an unique concept from a curated knowledge base. Such process is performed by associating unique concept identifiers from databases and/or ontologies with each chunk of text previously recognized by ML-based NER solutions. The techniques applied in this process are similar to the methods applied on dictionary-based approaches for NER. However, the matches are performed between the dictionary's entries and the chunks of text previously recognized as entity names, which allows performing a more flexible matching through approximate matching approaches or regular expressions. The normalization process starts by verifying if the recognized name matches any name on biomedical resources. If there is no unique identifier related with the name, there is no solution to assign an identifier to the name, so it may be discarded as an

entity name. Otherwise, if the name has one unique identifier, it is immediately assigned. If the match associates the name with multiple identifiers, it is considered ambiguous since it has more than one sense. For instance, considering the term "culture", it could be associated to at least two different meanings: "laboratory culture" or "anthropological culture" [167].

Ambiguity of biomedical terms is very common, due to the complexity of the domain. For instance, Jimeno-Yepes and Aronson [167] analysed MEDLINE and concluded that the most ambiguous term is "study", which is mapped to six different concepts more than three million times. They also studied ambiguity in terms of concept types, concluding that the most ambiguous concepts belong to the "Gene or Genome" and "Amino Acid, Peptide, or Protein" UMLS semantic types. From a different perspective, Weeber et al. [168] analyzed MEDLINE and concluded that 11.7% of sentences were ambiguous relative to the UMLS Metathesaurus.

Disambiguation is the process of resolving ambiguous names to the correct concepts. When it is performed successfully, it increases the number of biomedical concept names normalized correctly, contributing to improved concept recognition and information extraction. Thus, the goal of Word Sense Disambiguation (WSD) solutions is to minimize this problem by identifying the meanings of ambiguous terms in a specific context [169, 170]. Such solutions require the application of advanced disambiguation techniques, which are not trivial and require a large amount of curated knowledge.

Other corpora (see Section 2.1.1) developed for concept recognition may be used to evaluate WSD solutions, since they also provide unique concept identifiers for each named entity. Nonetheless, various corpora were built specifically for WSD, providing terms, associated meanings and text passages for each meaning. For instance, considering the previous ambiguity example of "culture", one such corpus may provide manually annotated text passages associated with "laboratory culture" and passages that refer to "anthropological culture", distinguishing between the different meanings of "culture". The following corpora are commonly used in the development and evaluation of WSD solutions:

- NLM WSD test collection [168]: the last version contains more than 37 thousand MEDLINE abstracts annotated with 203 ambiguous words with almost 38 thousand occurrences. UMLS is used as the knowledge base, providing Concept Unique Identifiers (CUIs) for each term;
- Medstract [171]: focused on acronym disambiguation, contains 186 abstracts annotated with 173 acronym-meaning pairs;
- MuchMore [172]: based on the Springer corpus of medical abstracts, contains both English and German versions of the same 7823 abstracts. However, the inter-annotators agreement is considerably low, with 65% for German and 51% for English;

Current solutions for WSD can be categorized as being ML- or knowledge-based. ML-based solutions apply ML techniques to automatically learn which concept is associated with a spe-

cific term, taking advantage of both supervised, semi-supervised and unsupervised learning. Supervised learning approaches build feature vectors describing each ambiguous word and its context, using a ML model to classify it into one of its possible senses, i.e., concept identifiers. Such approaches take advantage of rich set of features [173, 174], such as:

- Linguistic: tokens, lemmas, POS, chunking and dependency parsing;
- Morphological: char n-grams and word shape;
- Local context: windows of features and/or conjunctions of tokens' features that co-occur frequently and that contain the ambiguous term;
- Distance and position: reflect the position of the token to the ambiguous term, through its distance and orientation (left or right of the term);
- Domain knowledge: recognize named entities, such as disorders, drugs and procedures;
- Document metadata: section heading and medical speciality.

The generated feature vectors are then provided as the input of classical classification algorithms, in order to classify each term as being one of the considered concept identifiers. For instance, Stevenson et al. [175] compared the application of various classification algorithms in the NLM WSD corpus, showing that Vector Space Model (VSM) achieves the best performance results, with an accuracy of 87.9%. The authors also showed the importance and positive impact of local context and domain knowledge features, using MeSH terms matching as input of the ML models. On the other hand, Joshi et al. [176] applied a rich feature set to show that SVMs outperform NB and decision trees in most cases.

Regarding semi-supervised approaches, Jimeno-Yepes and Aronson [177] showed the positive impact of using unlabeled data on biomedical WSD, applying both co-training and self-training. Thus, the authors applied NB with unlabeled documents containing the ambiguous terms, achieving improvements of almost 2% of accuracy.

Supervised and semi-supervised ML-based solutions are limited by the amount and quality of the annotated data. Considering the largest and one of the most used corpora, NLM WSD, it only provides 203 terms, which is considerably restrictive considering the complexity of the biomedical domain. Moreover, Liu et al. [178] showed that "at least a few dozen" labelled examples per ambiguous term are necessary to develop competitive ML-based solutions. Thus, generating manually annotated corpora for WSD requires a huge effort, requiring a large number of labelled examples per ambiguous term. Consequently, the interest for solutions that do not require labelled data has considerably increased, with the application of ML unsupervised algorithms and purely knowledge-based approaches. ML unsupervised algorithms typically apply clustering techniques to build document clusters. For instance, Schütze [179] applied this approach building a cluster of documents for each meaning of a term. Afterwards, a query with an ambiguous term is matched with the obtained clusters, assuming that the documents in the best matching cluster have the same meaning as the ambiguous term in the query. However, such approaches are typically general, and do not take advantage of domain

knowledge. Thus, when applied to biomedical WSD, the achieved performance results may not be state-of-the-art [180].

Knowledge-based approaches have wider applicability, taking advantage of the available knowledge resources and the provided links between concepts. At first, such approaches were applied to gene and protein names disambiguation, since a gene may have multiple associated species, and consequently as many concept identifiers. Thus, the idea is to use external information to detect the correct unique identifier [175, 181, 182]. Considering that each gene or protein has a large amount of related information on biomedical resources, such as diseases, functions, tissues, mutations and domains, for each identifier that is candidate for the ambiguous term, the method will find all information that is related with the gene or protein in the surrounding text, and the identifier with the highest likelihood is selected. For instance, specialized species disambiguation solutions may apply more complex approaches, taking advantage of ML and/or fine tuned filtering rules. Wang and Matthews [183] present a rule and ML-based system that improves the performance of baseline systems for concept identification.

In the last years, knowledge-based approaches started to be widely applied to perform general biomedical WSD. In most cases, the UMLS Metathesaurus is used as the knowledge resource, since it provides a wide coverage of the biomedical domain knowledge. Such approaches follow the strategy applied for gene and protein names disambiguation, matching the text context of the ambiguous term with knowledge resources, using the ambiguous term definition, synonyms and related concepts. Afterwards, advanced scoring strategies are applied to find the concept more related with the textual context of the ambiguous term. For instance, McInnes [184] applied the Machine Readable Dictionary (MRD) algorithm [185], in order to find the concept with highest cosine similarity with the information extracted from the text context. On the other hand, Agirre et al. [186] uses graph-based representation of the UMLS Metathesaurus, which provides a network of relations between the various concepts. In order to select the closest concept, Agirre and Soroa [187] adapted the Google Page Rank [188] algorithm for WSD, combining the ambiguous term context with the concept topology in the graph network. Jimeno-Yepes and Aronson [167] present a detailed analysis of knowledge-based approaches for biomedical WSD, comparing the achieved performance results in the NLM-WSD corpus. The authors showed that the MRD approach outperforms the Page Rank method, with an accuracy of 63.9%. However, the best results were achieved by combining three knowledge-based approaches, achieving an accuracy of 76.3%. Such results represent a drop of 10% of accuracy when compared with supervised ML-based approaches. However, the wider applicability of such approaches justifies its application.

2.3 Relation mining

Biomolecular events such as gene transcription, protein binding or cell cycle regulation, play a key role in the interpretation of biological processes and cellular functions. For instance, a given protein may regulate the expression of a gene, whose products are in turn involved in some biological process. These events, as well as their biological significance and impact, are usually described in the scientific literature, and building up the complex chains of events that compose a biological network is a very demanding and time-consuming task. Additionally, the yielded knowledge can also be used by the pharmaceutical industry for both drug discovery and design, as the identification of proteins involved in key events might result in the subsequent uncovering of new drug targets. Thus, automatic event extraction from scientific literature constitutes an important contribution, in order to help find hidden biological relationships and allow faster updating of existing knowledge. Moreover, by processing millions of scientific articles and through the application of the ABC model defined by Swanson [17], the automatic extraction of relations between concepts may contribute to new findings, generating new knowledge [21]. As a result, relation/event mining is considered an important and established way to extract information from biomedical literature [189], being actively researched by dozens of research groups around the Globe.

In the beginning, researchers focused their work on extracting direct and coarse-grained associations between two concepts, which are known as binary relations. Figure 2.11 illustrates the textual representation of binary relations. In the sentence "Alpha-synuclein and parkin contribute to the assembly of ubiquitin lysine 63-linked multiubiquitin chains.", two different relations can be inferred between the three proteins, since both "alpha-synuclein" and "parkin" contribute to the assembly of "ubiquitin lysine 63-linked multiubiquitin chains". Considering the valuable information obtained by extracting such relations, it has been applied targeting different tasks and domains, such as:

- Protein-Protein interactions (PPIs): contribute to a better understanding of biological functions and molecular processes;
- Gene-Drug: understand how specific drugs can be tailored to specific genetic contexts;
- Gene-Disorder: understand the role that genetic information plays on specific diseases and/or phenotypic phenomena;
- Drug-Drug interactions (DDIs): improve multi-drug therapy by understanding how a drug affects the activity of another;
- Drug-Disorder: understand adverse drug reactions to improve pharmacovigilance;
- Location: physical location associated with specific concepts, such as "contained in" and "has location";
- Functional: general functional relation between concepts, such as "is caused by" and "is treatment for".

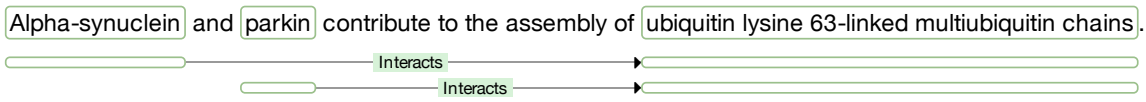


Figure 2.11: Relation mining illustration with the sample sentence "Alpha-synuclein and parkin contribute to the assembly of ubiquitin lysine 63-linked multiubiquitin chains.".

Even though binary relations already allow collecting and relating facts not achievable before, sometimes they cannot fully represent the biological meaning of the original text [190], since many information facts can only be ideally expressed in tertiary, quaternary, or even higher-order relationships. For instance, considering the the tertiary relation "proteins A and B synergistically activate gene C", it can be broke down into three binary relations: "protein A binds protein B", "protein A activates gene C", and "protein B activates gene C". However the combined binary statements are not equivalent to the original tertiary relationship [191]. As a consequence, there was a need for a representation strategy able to exemplify complex relations extracted from text. Such limitation was addressed by the BioNLP shared tasks [192–194], introducing the biomedical event extraction tasks to identify complex and nested relations from text. The aim is to extract not only relations between concepts, but also relations between concepts and another relations, and even relations between relations. Such tasks were the first step towards the extraction of specific pathways with precise information about the molecular events involved. Textual representation of complex relations typically occurs as a relation between a word indicating the type of relation, which we call the trigger, and one or more arguments, which may be a biomedical concept or another relation. For instance, Figure 2.12 contains two different biological events: *1*) Gene Expression between the trigger word "expression" and the protein "interferon regulatory factor 4"; and, *2*) Negative Regulation between the trigger "Down-regulation" and "expression", representing event 1.

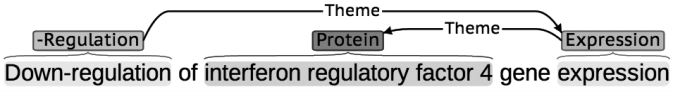


Figure 2.12: Textual representation of a complex biomedical event.

Instead of targeting coarse-grained relations as in previous tasks, the BioNLP shared tasks targeted very specific PPIs, in order to better understand the roles of specific concepts in a set of biological events, such as gene expression, transcription, phosphorylation and regulation. At first, researchers targeted the recognition of events particularly focused on transcription factors in human blood cells, using the GENIA corpus [195] as baseline for data preparation. Table 2.7 presents a brief description of target events considered in the GENIA shared task at BioNLP 2009 [192]. As we can see, various event types have different levels of complexity. Gene expression, transcription, protein catabolism, phosphorylation and localization are clas-

sified as simple events, since they only require unary arguments. On the other hand, binding and regulation events are considerably more complex, since binding requires the recognition of an arbitrary number of arguments, and regulation requires the identification of a recursive event structure. More recently, in the BioNLP 2011 [193] and 2013 [194] shared tasks, the organizers introduced new event annotation tasks, targeting different domains with different challenges, such as infectious diseases, bacteria genetics, cancer genetics and pathway curation.

Table 2.7: Description of the event types involved in the BioNLP 2009 shared task (Pr: Protein, Ev: Event, En: Entity, +: arguments that may be filled more than once per event).

Event type	Primary arguments	Secondary arguments
Gene expression	Theme(Pr)	
Transcription	Theme(Pr)	
Protein catabolism	Theme(Pr)	
Phosphorylation	Theme(Pr)	Site(En)
Localization	Theme(Pr)	AtLoc(En), ToLoc(En)
Binding	Theme(Pr)+	Site(En)+
Regulation	Theme(Pr/Ev), Cause(Pr/Ev)	Site(En), Csite(En)
Positive regulation	Theme(Pr/Ev), Cause(Pr/Ev)	Site(En), Csite(En)
Negative regulation	Theme(Pr/Ev), Cause(Pr/Ev)	Site(En), Csite(En)

Due to the BioNLP shared tasks, biomedical text mining researchers typically refer to binary relation mining as the task of relation extraction, and complex relation mining as event extraction. However, the implementation of relation and event mining solutions follow a similar general processing pipeline with comparable techniques, which may be composed by the following resources and steps (Figure 2.13):

- Corpus: annotated examples for development and/or evaluation;
- Pre-processing: processing methods to enable automatic relation mining;
- Concept recognition: automatically recognize concept names and associate identifiers from known knowledge bases;
- Document classification: in some cases, it may be useful to automatically classify the document as of interest for the target relation or not;
- Trigger recognition: identify the chunk of text that triggers the relation and serves as predicate;
- Relation extraction: automatically extract relations between concepts;
- Post-processing: refine recognized relations;
- Annotated corpus: output documents containing recognized concepts and target rela-

tions.

To concentrate the efforts on the novel aspects of biomedical relation and event mining, it is usually assumed that concept recognition has been already performed. Thus, the task typically begins with a gold standard set of concept annotations. Moreover, in order to guarantee solutions flexibility to different domains, concept names are typically normalized into a single representation. For instance, considering the sentence "BAG1 interacts with Tau.", the protein names "BAG1" and "Tau" are converted into representative and sequential tokens, such as "PRO1 interacts with PRO2".

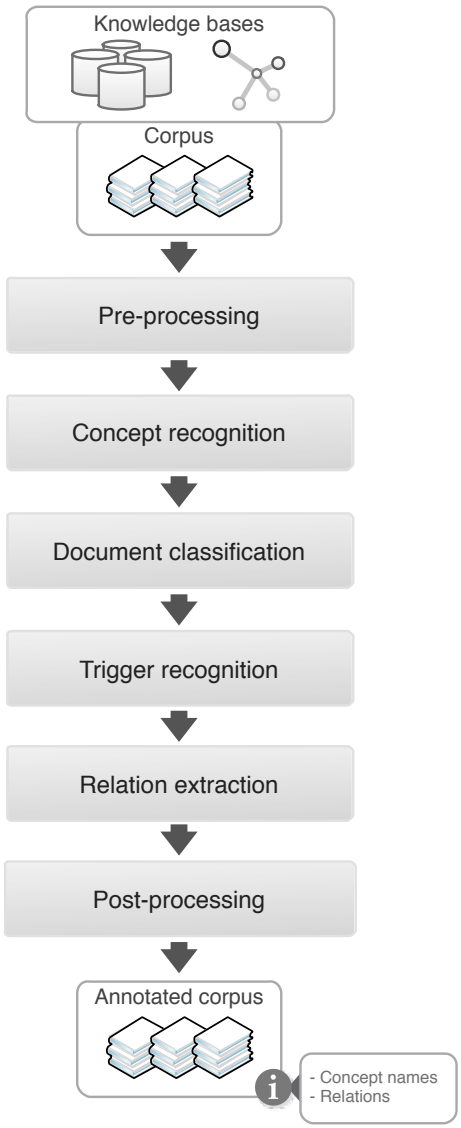


Figure 2.13: General processing pipeline of relation mining solutions.

2.3.1 Resources

Knowledge bases

Table 2.8 presents a list of relevant knowledge bases for binary relation mining, which are more focused on direct and general associations between concepts. Due to the importance of PPIs, there are many databases that may provide input information to extract such binary relations from the literature, such as STRING and BioGRID. On the other hand, there are also databases that explore DDIs in detail, such as DrugBank and PharmGKB. PharmGKB also catalogs the interactions between genes and diseases with drugs, in order to better understand the genetic origins of adverse drug events and the interaction with medicines.

Table 2.8: List of relevant knowledge bases to support binary biomedical relation mining.

Database	Relation(s)
STRING [16]	• PPI
BIND [196]	• PPI
MINT [197]	• PPI
IntAct [198]	• PPI
BioGRID [199]	• PPI
DrugBank [73]	• DDI
PharmGKB [74]	• DDI • Gene-Drug • Gene-Disease

Even though the previously presented databases are mainly focused on storing direct relationships between concepts, they can be also used to support complex relation mining, following the chains of relations. However, since complex relation mining tasks are mainly targeted to understand specific biological events, the presented list of databases is focused in such tasks. Table 2.9 presents a list of relevant knowledge bases for event mining. Metabolic pathway databases, such as KEGG, Reactome and BioC, intend to facilitate system-level understanding by cataloging every reaction between chemicals in a cell, providing detailed location information. This information can be used to support the extraction of biomolecular events chains, in order to discover new knowledge or to assist the curation of such complex pathways from documents. There are also specialized databases that intend to catalog how specific mechanisms occur considering a specific target or domain, storing relevant information regarding the participating agents and the connections between them. For instance, Transfac is focused in collecting eukaryotic transcription factors, including their experimentally-proven binding sites, and regulated genes. Transcription factors are recognized as important components of signaling cascades, controlling all types of cellular processes as well as the response

to external stimulus. Thus, such specialized databases can be used to support the extraction of biomedical events focused on specific tasks and domains.

Table 2.9: List of relevant knowledge bases to support complex biomedical relations mining.

Database	Information
KEGG [76]	• Metabolic pathway
BioCyc [200]	• Metabolic pathway
MetaCyc [200]	• Metabolic pathway
Reactome [201]	• Metabolic pathway
BioSystems [77]	• Metabolic pathway
WikiPathways [202]	• Metabolic pathway
Transfac [203]	• Transcription factors

Corpora

Various corpora were built to support the development of relation mining solutions, delivering documents with carefully annotated concept names and relations between them. Table 2.10 presents a list of relevant corpora considering various types of binary relations and granularity. Overall, there are not many corpora for binary relation mining, since such tasks are still quite challenging and the development has been highly focused on PPI mining. However, researchers have already started deviating from such tasks and started working on the tasks of DDI, Target-Disease and Gene-Disease relations. The number of available corpora for PPI mining reflects its importance in the field, since it is one of the core tasks in molecular biology. In order to unify and standardize such corpora, Pyysalo et al. [204] performed a comparative analysis and uncovered key similarities and differences. In the end, the authors developed a conversion tool to create a standard format for all PPI corpora. On the other hand, DDI mining has been widely promoted by the DDI Extraction challenges in 2011 [205] and 2013 [206], and researchers have already started working on extracting relations focused on genes, drugs and diseases, in order to properly understand the path from genotype to phenotype.

Regarding complex relations extraction, Table 2.11 presents a list of relevant corpora for biomedical event mining, considering target concepts and events, granularity and respective size. In this domain, each corpus or set of corpora is directly related with research efforts and target tasks. As we can see, most research efforts have been promoted by the BioNLP challenges, with seven corpora resulting from the organization of such collaborative efforts. Most tasks are focused on gene-centric events, due to the relevant information resulting from mining different biomolecular processes and functions, such as gene expression, transcription and regulation. The GENIA task was one of the first to be performed, defining the guidelines

Table 2.10: List of relevant corpora for biomedical relation mining.

Corpus	Year	Concepts	Relations	Granularity	Size
AIMed [207]	2005	• Protein	• PPI	Sentences	1955
BioInfer [208]	2007	• Protein	• PPI	Sentences	1100
HPRD50 [209]	2007	• Protein	• PPI	Sentences	145
IEPA [210]	2002	• Protein	• PPI	Sentences	486
LLL [211]	2005	• Protein	• PPI	Sentences	77
DDI Extraction 2011 [205]	2011	• Drug	• DDI	Abstracts	579
DDI Extraction 2013 [212]	2013	• Drug	• DDI	Abstracts	714
EU-ADR [213]	2012	• Gene • Drug • Disease	• Gene-Drug • Gene-Disease • Drug-Disease	Abstracts	100
ADE [214]	2012	• Gene • Disease	• Gene-Disease	Abstracts	1644
GENIA [195]	2013	• Organism • Cell • Tissue • DNA • RNA • Protein	• Has-region • Part-of • Has-variant	Abstracts	1999

and standards for future efforts. It keeps being actively researched, since it still presents various complex challenges and unsolved problems. Researchers have further investigated the application of event mining solutions considering other tasks and goals, such as events associated with infectious diseases and cancer genetics. Similar approaches were also applied to assist pathway curation, through the extraction of events between genes, chemicals and cellular components, such as transcription, translation, binding and regulation. Finally, out of the BioNLP challenges, the MLEE corpus targets the extraction of events associating anatomy concepts, such as cells, tissues and organs.

The various tasks vary significantly in terms of complexity, considering different concepts and events. Besides the complexity of the task in terms of biological information, we believe that a higher amount of concepts and events is directly related with increased complexity and ambiguity. For instance, the amount of concepts and events considered in the GENIA task is significantly different from that in the cancer genetics task, which includes a wider set of concepts and respective events.

Regarding corpora size and granularity, the amount of manually curated documents is reduced. The larger corpus only contains 1210 abstracts and 14 full-text documents. This is

justified by the effort required to manually annotate events and respective concepts. Moreover, researchers have already explored performing event mining in full-text documents, which presents different challenges and lower performance results. For instance, in the 2013 edition of the GENIA task, only full-text documents were used.

Table 2.11: List of relevant corpora for biomedical event mining.

Corpus	Year	# Concepts	# Events	Granularity	Size
Multi-Level Event Extraction (MLEE) [215]	2012	11	13	• Abstracts	• 262
BioNLP GENIA 2009 [192]	2009	1	9	• Abstracts	• 1210
BioNLP GENIA 2011 [216]	2011	1	9	• Abstracts • Full texts	• 1210 • 14
BioNLP GENIA 2013 [217]	2013	1	13	• Full texts	• 34
BioNLP Epigenetics and Post-translational Modifications 2011 [218]	2011	1	8	• Abstracts	• 1200
BioNLP Infectious Diseases 2011 [219]	2011	5	10	• Abstracts	• 800
BioNLP Cancer Genetics 2013 [220]	2013	18	37	• Abstracts	• 600
BioNLP Pathway Curation 2013 [221]	2013	4	19	• Abstracts	• 525

2.3.2 Document classification

Document classification intends to find articles that satisfy a specific information need. Thus, the goal is to develop automatic solutions to prioritize articles for literature curation, improving the selection of documents mentioning a particular concept or event of interest [222]. Since simple keyword queries are often inefficient in detecting relevant articles for complex biological events [223], such as PPIs, the goal is to find descriptions in the document that may indicate that it is relevant for the target relation mining task. Technically, the main goal of document classification solutions is to assign a score to each document that reflects the probability of containing such relations. In the end, documents above a pre-defined threshold are accepted for processing, and the remaining are discarded. Since such solutions typically

achieve high performance results, their integration in the relation mining pipeline provides two main advantages: *1*) reduce mistakes since a large amount of documents not containing relations are not processed; and, *2*) boost processing speed by discarding non-informative documents.

The development of document classification solutions targeting relation mining have been highly promoted by the BioCreative challenges, through the PPI Interaction Article Subtask (IAS) in BioCreative II [223], and the PPI Article Classification Task (ACT) in BioCreative III [224]. A total of 19 teams submitted 51 runs to the IAS task in BioCreative II, where the best performing solution [225] achieved an accuracy of 75.3% and an F-measure of 77.7%. On the other hand, a total of 52 runs were submitted by 10 teams to the BioCreative III ACT challenge, with the best performing system [226] achieving an accuracy of 89.2% and an F-measure of 61.3%. Overall, on both tasks of BioCreative, most teams applied some sort of ML-based technique, the best results being obtained using SVMs, Maximum Entropy or Large Margin classifiers. The top performing teams used various levels of lexical analysis, including POS tagging, NER, and dependency parsing to extract textual features for classification.

One of the best and most complete solutions was presented by Kim and Wilbur [226], which achieved the previously described best performing results in the ACT challenge at BioCreative III. It uses both word and dependency parsing features together with protein name identification, in order to effectively capture PPI patterns. As a classification module, a large margin classifier with Huber loss function is applied, which presents competitive results in comparison with SVMs. The authors showed that the used syntactic patterns contribute to a classification performance improvement in terms of recall. On the other hand, Lan et al. [227] compared the use of Bag-of-Words (BoW), interaction trigger words and protein name features in a SVM classifier for identifying articles discussing PPIs. They tested the classifiers using the BioCreative II data set, and reported a precision of 70% and a recall of 87% when using the BoW features. Their best result, when using a single classifier, was obtained with a feature set containing BoW features and protein names co-occurring with interaction trigger words, with an F-measure of 77%. Abi-Haidar et al. [228] tested three different classifiers in the BioCreative II data set, namely SVM, Singular Value Decomposition (SVD) and Variable Trigonometric Threshold (VTT). The authors reported a top F-measure of 78% using the VTT classifier and a feature set of 650 discriminating words. Finally, Suomela and Andrade [229] proposed a different approach based on word frequencies, which, given any two articles, decides which one is more related to a topic. The extracted keywords were restricted to words that commonly convey meaning, that is, nouns, verbs, and adjectives. The authors proposed a classification and ranking model to evaluate the entire MEDLINE database with respect to a topic of interest. The method, which presents an F-measure of 65%, relies on the different frequencies of discriminating words between the training set and other non-relevant articles on a reference set. This approach is also behind the MedlineRanker web-service

[230], which allows to retrieve a list of articles ranked by similarity to a training set defined by the user. This training set can be obtained from a PubMed search, or from PubMed document identifiers associated with a given indexing term from the MeSH vocabulary, for example. Another possibility, as referred by the authors, is to use a list of document identifiers obtained from a PPI database, therefore getting as the result articles related to that topic.

2.3.3 Trigger recognition

Trigger recognition is the first and one of the most important tasks to properly perform relation mining, since many approaches rely on its output to properly extract relations from text. Since events are defined around the trigger, which defines the type of event, trigger recognition is considered a fundamental step in event mining solutions. However, binary relation corpora typically do not provide trigger annotations, which makes the identification of triggers a non-mandatory step, since it may be performed without previously collecting triggers.

Approaches to perform trigger recognition can be categorized as being based on rules, dictionary matching and machine learning. Each solution presents different advantages and limitations, depending on the available resources and target task. Since binary relation mining corpora do not provide manually annotated triggers, it is not possible to train and evaluate ML-based solutions. Thus, binary relation mining systems typically take advantage of rule- and dictionary-based approaches for trigger recognition.

Rule based approaches apply a set of manually or automatically generated linguistic rules to extract trigger words. For instance, Casillas et al. [231] identified the most common trigger-based patterns from training data using lemmas, such as "phosphorylat* + of + PROTEIN", where "phosphorylat*" represents the trigger.

Regarding dictionary-based solutions, developers need to collect trigger words for each relation type, in order to build a focused knowledge resource, i.e., dictionary. In the end, the words in the dictionary are matched with the text and accepted as triggers for each relation type. However, such an approach accepts all trigger words without considering the textual context, possibly producing large amounts of false positives. To minimize this problem, manual linguistic rules can be applied, in order to filter provided triggers and significantly reduce the amount of false positives. For instance, Le Minh et al. [232] accepts only words that are present in specific contexts and with specific POS tags, such as "NN/NNS + of + PROTEIN" and "VBN + PROTEIN". On the other hand, Kilicoglu and Bergler [233] applied statistical measures based on linguistic features to collect "good" trigger words from training data.

ML based solutions intend to minimize various problems of rule and dictionary-based solutions, namely regarding context definition. ML-based solutions vary with the used statistical model and extracted feature. SVMs are the most commonly used ML model for this task. For

instance, Björne et al. [234] applied SVMs with a complex feature set consisting of tokens, dependency parsing tree and external resources to identify event triggers for each input sentence. The problem of multiple trigger types per chunk of text is solved through the application of composite labels. Miwa et al. [235] also took advantage of SVMs, but training two different models: one for trigger-protein (TP-T) relations and another for trigger-trigger (TT-T) relations, using the output of the TP-T predictor as an input feature for the TT-T model. Their system employs a complete feature set based on tokens, local context and dependency parsing with shortest paths features. On the other hand, Zhang et al. [236] used SVMs with neighborhood hash features to reflect the syntactic structure of the sentences, in combination with token and sentence-based features. Finally, Martinez and Baldwin [237] used SVMs in the perspective of WSD, by defining a list of target words, i.e., triggers. This solution also used features based on tokens, context, dependency parsing and external resources. Besides SVMs, CRFs have also been widely applied, presenting state-of-the-art results on sequence tagging problems. For instance, MacKinlay et al. [238] used CRFs with a feature set based on token, dependency parsing and context definition features. Martinez and Baldwin [237] also applied CRFs using a similar feature set as applied in the WSD approach.

Considering the BioNLP shared tasks [192, 194, 216], ML-based approaches were the most commonly used, followed by dictionary-based and rule-based systems. Regarding performance behavior, ML-based solutions present the best results, followed by dictionary matching approaches. Considering the GENIA event task of the BioNLP 2009 challenge, one of the best SVM-based solutions [239] achieved a total of 65% of F-measure in the recognition of nine different triggers.

2.3.4 Relation extraction

Extraction of relations from scientific documents can be performed through the application of different techniques, based on co-occurrences, rules, linguistic processing, ML and knowledge. Each approach presents different advantages and limitations, being more appropriate for different tasks, considering the available resources.

Co-occurrences

Co-occurrences assume that if two concepts are usually referred in a specific text passage, i.e., sentence, paragraph or section, they are related. Following this approach, the sentence on Figure 2.11 provides three different relations, since “alpha-synuclein” and “parkin” co-occur with “ubiquitin lysine 63-linked multiubiquitin chains” and with each other. As a straightforward approach, it has been widely applied to extract relations between many biomedical concepts, providing positive outcomes and significant new discoveries [21]. Typically, the application of co-occurrences is considered the baseline approach of any relation mining task. However, since all possible relations are provided, this approach typically provides a large

amount of false positives and achieves low precision results, which is a serious drawback. Nonetheless, dozens of solutions were developed to provide general and focused biomedical relations. For instance, Chen and Sharp [240] presented a system called Chilibot, which constructs content-rich relationship networks between genes, proteins, drugs and diseases. It applies co-occurrences to extract relations when two entities occur in the same abstract, but not the same sentence. On the other hand, iHOP [18] performs protein-protein interaction mining using co-occurrences in a sentence. In the end, the authors built an interaction network containing almost three thousand organisms and 110 thousand genes. Following a different approach, Tsuruoka et al. [19] provide a web-based system called FACTA, which helps finding newly associated concepts considering a pre-processed input query. Genes, proteins, drugs, diseases, symptoms, enzymes and chemical compounds are presented in a tabular format and ranked based on the co-occurrence statistics.

As far as we know, there are no co-occurrence-based solutions for event mining, due to the requirements and complexity of the task.

Rule-based

Rule-based approaches apply pattern-based rules to extract relations between concepts. Table 2.12 presents a list of sample rules to extract PPIs from scientific articles. Thus, each sentence is matched with the considered rules, if the match succeeds, it is considered that the sentence contains a relation. Afterwards, the sentence is processed to extract the relation(s) considering the previously recognized concepts and matched rule.

Table 2.12: Example of pattern-based rules to perform PPI mining.

Rule	Example
PRO1 word* TRIGGER word* PRO2	PRO1 interacts with PRO2
TRIGGER word* PRO1 word* PRO2	interaction between PRO1 and PRO2
PRO1 word* PRO2 word* TRIGGER	PRO1 and PRO2 interact
PRO1 word* PRO2 word* TRIGGER word* PRO3	PRO1 and PRO2 interact with PRO3

The application of rule-based approaches has been highly focused on PPIs mining. One of the first works was presented by Blaschke et al. [241], which defined a small set of trigger words with a single and simple extraction rule: “PRO1 TRIGGER PRO2”. The authors validated its applicability by reconstructing the protein interaction network in the Drosophila Pelle system, and by analyzing the cell cycle control in Drosophila. Plake et al. [242] presents a more detailed analyses applying 22 different rules for PPI mining, considering both POS tags and trigger words. The authors reported an F-measure of 52% in the BioCreative PPI corpus.

On the other hand, Ono et al. [243] applied a pre-processing step to simplify the rules matching step, using POS tags analyses to split a sentence into various parts. For instance, a sentence containing three proteins (PRO*) and two verbs (VB*), such as "PRO1 VB1 PRO2 VB2 CC PRO3", is divided into two parts: "PRO1 VB1 PRO2" and "PRO1 VB1 PRO3". Afterwards, the authors defined four target trigger words (interact, associate, bind and complex) and different rules for each one, in order to detect relations in the sentence parts. In a more updated research line, researchers apply rule-based techniques as a first candidate retrieval step. For instance, Bui et al. [244] presents an hybrid approach applying five different rules to extract candidate PPI pairs, which are then classified using a ML algorithm.

Regarding event mining, Bui et al. [39] present a considerably different approach, which extracts events by matching patterns collected from training data, which are based on previously defined features. At first, in order to select the sentences to be processed, the authors defined a set of containers (chunk, phrase and clause) with different pattern types. For instance, the pattern type "ARG1 - TRIGGER" accepts the expression event "interferon regulatory factor 4 gene expression" presented on Figure 2.12. Afterwards, a set of previously defined features is extracted from each of the accepted sentences, such as the POS tag of the trigger, the distance between triggers and concepts, and the number of events sharing the same pattern. The final patterns are based on the extracted features. Each generated pattern is then assigned a key by combining its event trigger, POS tag, pattern type, and container type. Such key is used to retrieve this pattern in the extraction step. In the end, this approach achieved the best strict-matching results in the BioNLP 2013 GENIA task, with a total F-measure of 48.9%.

Even though rule-based approaches are fundamental to understand some of the linguistic characteristics and patterns of biomedical relations, they typically present recall problems, since hand-made rules may be too specific and task-oriented. However, automatically generating such rules may contribute to significantly reduce such limitations. Thus, such approaches typically struggle to adapt to new biomedical domains of interest, hindering its wider application.

Linguistic-based

Linguistics-based approaches take advantage of the information provided by advanced linguistic parsers to automatically extract relations between concepts. Such approaches work by analyzing the linguistic dependencies between tokens, in order to find relations between nouns and specific predicates (triggers), which trigger the linguistic dependency and possibly the biomedical relation. Two different parsing techniques have been applied to perform such analyses: chunking and dependency parsing, which were previously described in Section 2.1.3. In the end, various rules are defined using linguistic parsing information to perform the actual relation extraction.

Huang et al. [245] extract PPIs by applying shallow parsing and pattern matching. This method works by extracting structures that are typically associated with protein interactions, such as appositive and coordinative structures. Appositive structures are composed of noun phrases that are side by side with one element serving to define or modify the other, and coordinative structures contain chunks that must be assembled together since they are semantically close, playing the same syntactic and grammatical role. Then, long sentences are split into sub-ones, from which relations are extracted by a pattern matching algorithm, along with automatically generated patterns. For instance, the pattern "NN IN PRO1 CC PRO2" (where NN is a noun, IN a preposition, CC a coordinating conjunction, and PRO* protein names) is able to recognize the relations "interaction between PRO1 and PRO2" and "association of PRO1 and PRO2". In the end, the authors report 66% of F-measure considering only four verbs: interact, bind, associate and modify.

RelEx [209] uses dependency parsing to extract PPIs. The idea is to create candidate relations by extracting paths connecting pairs of proteins from dependency parsing trees, applying the following rules:

- effector-relation-effectee (e.g., "A activates B"): extracts paths in the chunk dependency tree that lead from the effector to an effectee, i.e., from one protein to another;
- relation-of-effectee-by-effector (e.g., "Activation of A by B"): longest sequences of chunks that are connected by the terms "of", "by", "to", "on", "for", "in" and "through". A sequence is retained as candidate relation if it contains at least two of these terms and at least one between two chunks each containing at least one protein.
- relation-between-effector-and-effectee (e.g., "Interaction between A and B"): extracts two noun phrase chunks connected by a dependency of the type "between".

Post-processing modules are applied to filter candidate relations, focusing the extracted relations for PPI mining. Such step is performed by considering only a small set of terms that are typically used to describe a relation. In the end, the authors evaluated their method on the LLL and HPRD50 corpora, achieving F-measures of 82% and 78%, respectively.

Rinaldi et al. [246] applied a dependency parsing-based approach to extract gene-disease, gene-drug and drug-disease relations. The authors considered the PharmGKB database as the gold standard, which provides relations per document, to build and evaluate their method. The method works by collecting all paths between relevant concepts, which are then sorted by considering the gold standard. Thus, if the gold standard contains a relation between two concepts, the respective path is marked as a true positive, by incrementing the respective counter. Otherwise, the syntactic path is marked as a false positive. By filtering the paths by higher probability of delivering true positive relations, the authors evaluated their approach in 75 manually annotated unseen documents, achieving 30% of F-measure.

Linguistic-based approaches have also been widely applied in event mining [233, 247–250], taking advantage of syntactic paths to properly extracts events from literature. Such ap-

proaches can be differentiated by generating the linguistic-based rules manually or automatically. For instance, Bui and Sloot [247] manually defined specific rules to extract each event type from noun and verb phrase chunks. For instance, a noun phrase that contains two nouns, the protein and trigger, respectively. Post-processing methods are applied to determine the event type of an ambiguous trigger, and to check cross-references of regulatory events. A total F-measure of 43.9% in the BioNLP 2011 shared task was achieved. On the other hand, Kaljurand et al. [249] mapped syntactic relations to event structures, using training data to automatically calculate the probability of a given token to be a trigger, the probability of an event structure given the trigger, and the probability of a concept to be part of an event structure. Threshold values were then defined to filter triggers and syntactic paths. In the end, this approach achieved a total F-measure of 33% in the BioNLP 2009 shared task.

Overall, linguistic-based approaches are able to provide competitive performance results when specifically optimized for specific tasks. However, since dependency-parsing methods usually require a considerable amount of processing resources and time, processing a large amount of data may take some time, hindering real-time document processing. Nevertheless, its wider applicability to extract relations and events between different concepts is a positive outcome to consider.

Machine learning-based

ML-based approaches take advantage of statistical models to automatically extract relations from scientific literature, by classifying candidate relations as being a relation of interest or not. Such approaches typically achieve high performance results in relation mining, using rich feature sets to properly describe and differentiate between positive and negative relations. Thus, research on relation mining has been focused on the application of different ML models and feature sets. Overall, SVMs were the most applied ML model, varying with the applied kernel. Jung et al. [251] and Kim et al. [252] present a comparison of various SVM kernels applied to relation mining, including linear, polynomial, radial basis function, subsequence [253], tree [254], shortest path [255] and graph [256]. On the other hand, many feature types were investigated to properly describe the textual context. In summary, lexical, local context and syntatic parsing features were the most applied. For instance, Fayruzov et al. [257] and Van Landeghem et al. [258] have carefully analyzed the impact of lexical and syntactic features on PPI mining, confirming their positive contributions. Nonetheless, Miyao et al. [259] showed that the accuracy of syntactic parsers also contributes to the overall performance of PPI solutions.

Classic solutions apply a single ML model with a rich set of features using one or two syntactic parsers. For instance, Akane [260] applies SVMs with tree kernels and two syntactic parsers, using various syntactic path and context features related to words before, between and after the two interacting proteins. On the other hand, Kim et al. [252] applied a walk

kernel with one parser to explore the shortest paths between two proteins, and Airola et al. [261] presents an all-paths graph kernel with one parser to consider dependencies connecting two proteins outside and inside the shortest path.

Other solutions investigated the combination of various models to achieve improved results with heterogenous contributions. For instance, Miwa et al. [262] combines the input of two syntactic parsers with three SVM kernels (linear, tree and graph), in order to collect as much lexical and parsing features as possible. Such complete solution achieved high-performance results on five PPI corpora (AIMed, BioInfer, HPRD50, IEPA and LLL), with F-measures between 61% and 80%. On the other hand, Bui et al. [244] automatically categorized data into subsets based on its semantic properties, and trained a SVM model on each sub-set using manually tuned feature sets. By combining the inputs from the different SVM models, the authors achieved high-performance F-measure results, ranging from 60% to 84% on the same corpora.

Other research works targeted the extraction of different types of binary relations. For instance, JReX [263] proposes a system for extracting Gene-Drug, Gene-Disease and Drug-Disease relations. Since there is no gold standard corpus for such relations, the authors used the PharmGKB database to generate a training and evaluation corpus. Thus, based on the provided relations (without specific character positions of concept names), the authors identified the corresponding abstracts and considered a relation if the two participants appear in a sentence. Using this semi-gold standard corpus, a MEMM model was trained to classify candidate relations, provided by a dependency parsing-based filtering step. The classification model takes advantage of a rich feature set, based on lexical analysis, chunking, and dependency parsing paths and shortest paths between arguments. In the end, the authors reported F-measures of 75%, 69% and 78% for Gene-Drug, Gene-Disease and Drug-Disease mining, respectively.

Regarding event mining, most of the systems that participated in the BioNLP challenges applied ML-based solutions. For instance, in the GENIA task of BioNLP 2009 [192], 14 of the 22 participating teams applied ML, and in the same task of BioNLP 2011 [216], 7 of the 11 teams took advantage of ML. Such approaches can be divided in two groups based on how triggers and arguments are learned from annotated documents. The first group applies a typical processing pipeline with trigger recognition and argument detection being performed independently with a ML model for each. EventMine [264] and Turku [265] are two state-of-the-art solutions following this approach, which achieve high performance results in various event mining tasks. For instance, EventMine applies SVMs with input from GDep and Enju syntactic parsers. Argument detection is performed taking advantage of two different SVM classifiers, one for argument detection and another for multi-argument identification, in order to predict simple and recursive events. Thus, the argument detector identifies possible trigger-argument pairs. Each argument can be either an entity or the trigger of another event,

and is assigned a semantic role (theme or clause). A rich feature set is applied, using shortest paths between the candidate pair, between the argument and other entities, and between the trigger and other entities, as well as pair n-grams and local context features. Finally, the multi-argument event detector combines multiple trigger-argument pairs found by the argument detector to create complete event structures, and assigns an event type to them. In the end, the authors evaluated EventMine on three different tasks of BioNLP 2011, namely GENIA, Epigenetics and post-translational modifications, and Infectious diseases, achieving F-measures of 58%, 52% and 58%, respectively. In the second group, the idea is to reduce the propagation of mistakes made on trigger recognition. Thus, joint learning is applied to learn trigger and argument detection together, minimizing mistakes as much as possible. Riedel and McCallum [266] and Vlachos and Craven [267] have successfully applied this approach in the BioNLP challenges, applying different joint learning algorithms. For instance, Riedel and McCallum [266] divided the problem of event mining in three different sub-problems: *1)* find trigger labels and outgoing edges; *2)* find trigger labels and incoming edges; and, *3)* find pairs of proteins that appear in the same binding event. Considering a sentence and a set of candidate triggers, the goal of the inference algorithm is to maximize the contribution of the three components. For each component, a passive-aggressive online learning algorithm [268] is applied with different feature sets, containing syntactic and shortest paths, as well as local context and word n-gram features. In the end, the authors achieved F-measures of 53.10% and 53.40% in the GENIA and Infectious diseases tasks of the BioNLP 2011, respectively.

Overall, ML-based solutions present the best performance results on both relation and event mining, defining the state-of-the-art results on a number of benchmark datasets. Moreover, such solutions have the ability to adapt to different corpora, extracting PPIs from different domains with similar accuracy results. For instance, many ML-based solutions are typically applied to PPI mining considering cross-corpus evaluation, and only minor drops of performance are observed. However, it is important to keep in mind that using complex algorithms with resource-intensive linguistic processing techniques may result in solutions with positive accuracy but with considerably slow processing speeds, which hinders the wider applicability of relation mining.

Knowledge-based

Knowledge-based approaches take advantage of knowledge bases to infer biomedical concept relations based on their profiles, which are built using relations from literature or from curated databases and/or ontologies. By analyzing such profiles, researchers are able to obtain a score that reflects the probability of such concepts being related with each other, defining a threshold to accept possible relations. For instance, van Haagen et al. [269] performed PPI mining by building concept profiles from literature using co-occurrences of proteins with other proteins, drugs, diseases, disorders and chemicals. The contribution of each concept is

obtained through the uncertainty coefficient [270], which is an information-theoretical measure that considers the probability of direct relations, giving extra weight to concepts that are specific for the set of documents belonging to the protein for which the concept profile is built. Protein profiles are then compared using the inner product of uncertainty coefficients. Thus, if two proteins co-occur, the inner product of their concept profiles is high. The authors evaluated their approach by predicting PPIs already stored in six different PPI databases, showing significant improvements in coverage (76% versus 32%) and sensitivity (66% versus 41%). Finally, the applicability of their approach was illustrated by inferring the physical interaction between CAPN3 and PARVB.

Considering DDI mining, Tari et al. [271] extracted various facts of drug metabolism to collect not only DDIs that are explicitly mentioned in text, but also implicit interactions that can be inferred by reasoning. Explicit relations were collected through the application of dependency parsing, and implicit relations were obtained through logical inferences based on various properties of drug metabolism. Considering DrugBank as reference database with 494 DDIs, the authors achieved 77.7% precision on explicit relations and 81.3% precision on implicit relation extraction.

In a last trend of knowledge-based relation mining, researchers take advantage of graph-based databases to understand in detail how two concepts are related with each other. Such graph database can be built from literature or using available databases. For instance, Wren and Garner [272] identified related genes by analyzing the graph structure created by gene-gene co-occurrences collected from MEDLINE. In the end, the authors report about 97% of specificity at 85% of sensitivity. On the other hand, Kang [273] take advantage of a graph-based representation of UMLS to infer relations between drugs and adverse effects. Thus, considering the concepts in a sentence, the method searches the graph database for possible interactions considering a maximum number of hops. When evaluated in the ADE corpus, and considering a maximum distance of 4 hops, it achieved 50.5% of F-measure, outperforming a co-occurrence based approach by 34.4%. On the other hand, SemRep [274] is a general relation mining approach that combines dependency parsing analysis with domain knowledge (UMLS) to infer semantic prepositions, being more focused on extracting taxonomic relationships.

2.4 Summary

This chapter presented a careful analysis of the several tasks related to biomedical information mining from scientific literature, namely concept recognition and relation mining. Regarding concept recognition, relevant knowledge bases and corpora were presented, also describing in detail the advantages and limitations of each technique, and exposing the development details of rule, dictionary and ML-based approaches. A careful analysis of the

various approaches to perform normalization and disambiguation of concept names was also performed, namely ML and knowledge-based. Afterwards, we carefully analyzed the relevant knowledge bases and corpora for relation mining targeting different tasks, and presented various solutions that implement the most different techniques, namely co-occurrences, rule, linguistic, ML and knowledge-based.

Chapter 3

Gimli: machine learning-based biomedical named entity recognition

This chapter is based on:

- D. Campos, S. Matos, and J. L. Oliveira, "Gimli: open source and high-performance biomedical name recognition." *BMC bioinformatics*, vol. 14, p. 54, 2013
- D. Campos, S. Matos, and J. L. Oliveira, "Chemical name recognition with harmonized feature-rich conditional random fields," *Fourth BioCreative Challenge Evaluation Workshop*, vol. 2, pp. 82–87, 2013

One major focus of TM research has been on Named Entity Recognition (NER), a crucial initial step in information extraction, aimed at identifying chunks of text that refer to specific entities of interest. Several NER systems have been developed for the biomedical domain, using different approaches and techniques that can generally be categorized as being based on rules, dictionary matching or Machine Learning (ML). In this study we follow an ML approach, the goal being to train statistical models focused on recognizing specific entity names, using a feature-based representation of the observed data. This presents various advantages over other approaches, such as the recognition of new and short entity names. Moreover, ML solutions have been shown to achieve the best results for this specific domain.

3.1 Background

Various techniques for adapting and optimizing ML-based solutions for biomedical NER have been proposed in recent years. Overall, these efforts contain the following sub-tasks: pre-processing, feature extraction, modeling, and post-processing. In the initial step, the input data is pre-processed to make it readable by computers and to simplify the recognition process. This sub-task is one of the most important, since every single decision will affect the entire system behavior. Tokenization is a mandatory step, in order to divide natural language texts into discrete and meaningful units. There are several approaches to implement it, depending on the input data and desired output. For instance, Tsuruoka et al. [55] keep words that contain a dash as a single token, while Leaman and Gonzalez [165] create multiple tokens for the same word.

In the feature extraction step, it is important to obtain features that reflect the different characteristics of the sentences and tokens. At the token level, orthographic [165, 275–277] and morphological [55, 161, 276] features are commonly used in order to extract token formation patterns. It is also common to encode domain knowledge as features [63, 165] using external resources, such as lexicons of gene and protein names. At the sentence level, linguistic [137, 165] and local context features [55, 275, 277, 278], such as windows and conjunctions of features, are used to model the links between tokens.

The ultimate goal is to model the observed data using the features extracted in the previous step, thus creating a probabilistic description of the data classes. This task is accomplished using ML models, whose training can be classified as being supervised or semi-supervised, depending on unannotated data being used or not. Supervised learning, which only uses annotated data, has received most research interest in recent years. Consequently, different supervised models have been used on biomedical NER systems, such as CRFs [137, 165, 275, 278], SVMs [63] and MEMMs [55, 277].

Finally, the post-processing stage aims to improve the recognition results, cleaning annotation errors or refining incomplete annotations. The most common methods consist of removing annotations with unmatched parentheses [161, 278], adding the results of abbreviation resolution tools [63, 165], and extending names using a domain dictionary [278].

Although several open source solutions aimed at recognizing biomedical names have been proposed in recent years, most present one or more of the following limitations:

- are focused on a specific corpus and/or biomedical domain;
- do not take advantage of state-of-the-art techniques;
- present performance results that are deprecated and/or not in accordance with similar closed source solutions;
- are not configurable and/or easy to use;
- are not easily extensible to new features;
- are not easily scalable.

In this chapter we present Gimli, a new open source solution for automatic recognition of biomedical names. It extends and optimizes the most advanced state-of-the-art techniques in a simple and easy-to-use tool. By default, Gimli already provides high-performance trained models, supporting several known corpora formats. Moreover, it also allows easy and flexible development of new solutions focused on different semantic types, as well as training new ML models with different feature sets and characteristics.

3.2 Methods

This section presents a detailed description of the resources used and methods implemented, following the workflow of ML-based NER solutions. Figure 3.1 illustrates Gimli's architecture, presenting the connections between the various steps.

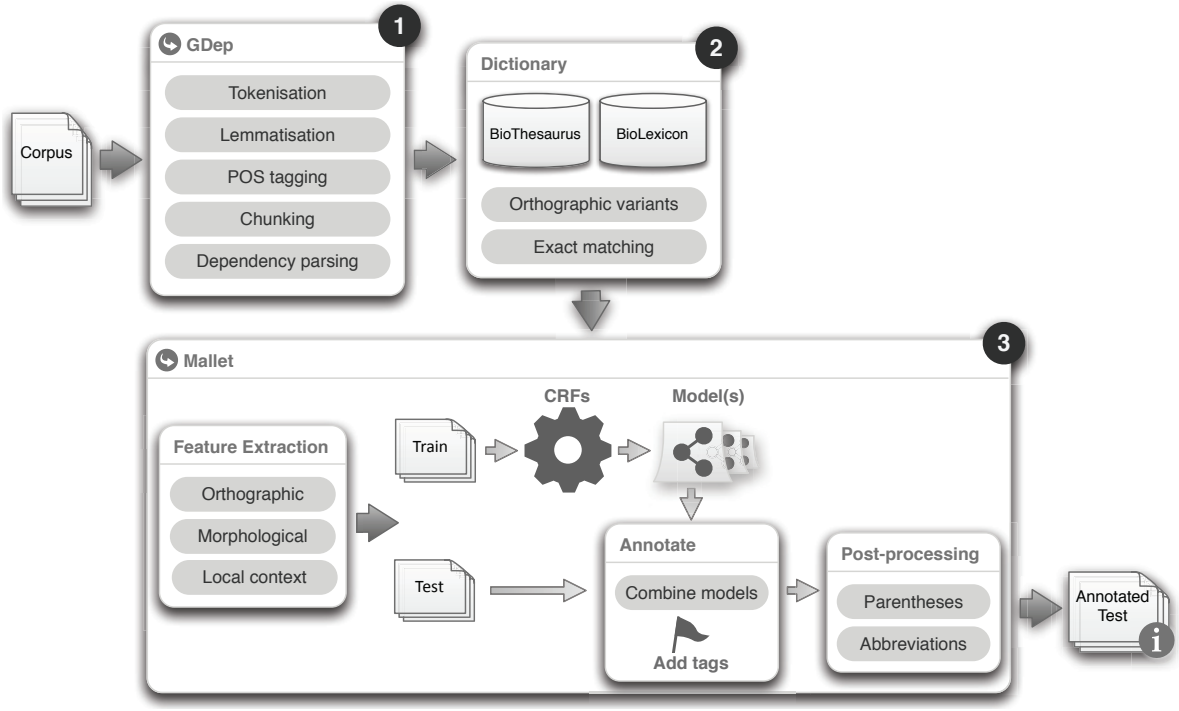


Figure 3.1: Overview of Gimli's architecture, presenting the workflow of required steps, tools and external resources.

3.2.1 Pre-processing

In recent years, various tokenization solutions have been developed for several domains and languages. Gimli uses the tokenizer from GENIA Tagger [55] (included in GDep) which was developed for biomedical documents and presents state-of-the-art results in this domain. However, words containing the symbols "/", "-" or "." are not always split into multiple

tokens. When working at the token level, this may create inconsistencies with the human provided annotations, constraining the model learning process and the recognition of some entity names. For instance, consider that "BRCA-1/2" is taken as one token and that in the gold standard only "BRCA-1" is tagged as an entity name. In the model training phase, the token "BRCA-1/2" as well as its local and contextual features will be considered as a negative instance, which will directly affect the final model. Thus, we decided to make the tokenizer behavior more consistent, by breaking words containing the symbols "/", "-" or "." into multiple tokens.

To train ML models, each token in the training data must be identified as being part, or not, of an entity name. We use the BIO encoding scheme, which is the *de facto* standard. In this scheme, tokens are tagged as being at the beginning (tag "B"), inside (tag "I") or outside (tag "O") of an entity name.

3.2.2 Features

Feature extraction is a crucial NER task, since the predictions will be performed based on the information that they encode. Nadeau and Sekine [279] present a complete survey on features used in general NER solutions. Gimli implements a rich set of features, including orthographic, morphological, linguistic parsing, external resources and local context features. We also propose improvements on various features, in order to optimize their behavior and performance results.

The purpose of orthographic features is to capture knowledge about word formation. For example, a word that starts with a capital letter could indicate the occurrence of an entity name (e.g., in the protein name "MyoD"). Table 3.1 lists the formation patterns used by Gimli to extract orthographic features from tokens.

Table 3.1: List of orthographic features organized by category.

CAPITALIZATION		COUNTING		SYMBOLS	
Feature	**Example**	**Feature**	**Example**	**Feature**	**Example**
InitCap	Kappa	SingleCap	U	Dash	-
EndCap	KappaB	TwoCap	US	BackSlash	/
AllCaps	DAV	ThreeCap	USS	OpenSquare	[
Lowercase	kappa	MoreCap	USSR	CloseSquare	]
LowAndCap	RalGDS	SingleDigit	1	OpenParen	(
		TwoDigit	12	CloseParen	)
		ThreeDigit	123	Colon	:
		MoreDigit	1234	SemiColon	;
				Comma	,
				Dot	.
				Apostrophe	'
				QuotationMa	"
				Percent	%
				Star	*
				Equal	=
				Plus	+

Morphological features, on the other hand, reflect common structures and/or sub-sequences of characters among several entity names, thus identifying similarities between distinct tokens. To accomplish this goal, three distinct types of morphological features are considered: suffixes and prefixes, char n-grams and word shape patterns. Particular prefixes and suffixes could be used to distinguish entity names. For instance, suffixes such as "ase", "ome" and "gen" frequently occur in gene/protein names [163]. A char n-gram is a subsequence of n characters from a given token. This feature type has an identical role to prefixes and suffixes, however it also finds common sub-sequences of characters in the middle of tokens. Finally, it is also important to extract the token's structure. Collins [280] proposed a method to generate a sequence of characters to reflect how letters and digits are organized in the token. We extended this idea to support symbols too. Thus, three distinct types of word shapes are used by Gimli:

- Word Shape Type I: replace sequence of digits by "*" (e.g., the structure of "Abc1234" is expressed as "Abc*");
- Word Shape Type II: replace each letter, digit and symbol by a morphological symbol (e.g., the structure of "Abc:1234" is expressed as "Aaa#1111").
- Word Shape Type III: replace each sequence of letters, digits and symbols by a morphological symbol (e.g., the structure of "Abc:1234" is expressed as "a#1").

The most basic internal feature is the token itself. However, in most cases, morphological variants of words have similar semantic interpretations, which can be considered as equivalent. For this reason, lemmatization is commonly used to group together all inflected forms of a word, so that they can be taken as one unique feature. On the other hand, it is also possible to associate each token with a particular grammatical category based on its definition and context, a procedure called POS tagging. Moreover, we also use chunking, dividing the text into syntactically correlated chunks of words (e.g., noun or verb phrases). The BIO encoding format is used to properly indicate the beginning and end of each chunk. For instance, considering two consecutive tokens that make part of a noun phrase chunk, the tag "B-NP" is associated with the first token and the tag "I-NP" with the second one. In the end, each tag is used as a feature of the respective token.

The previous features provide a local analysis of the sentence. To complement these with information about relations between the tokens of a sentence, we use features derived from dependency parsing. Namely, we follow a strategy similar to the one presented by Vlachos [281], considering only those dependencies that could indicate the presence of an entity name. Thus, we add as features of each token, the lemmas corresponding to each of the following: verbs for which the token acts as subject; verbs for which the token acts as object; nouns for which the token acts as modifier; and the modifiers of that token.

Gimli is further optimized by adding biomedical knowledge to its features. To provide this knowledge, dictionaries of specific domain terms and entity names are matched in the text and

the resulting tags are used as features. Thus, the tokens that make part of a matched term contain a feature that reflect such information. For instance, if the term "BRCA" is matched, the feature "LEXICON=PRGE" is added to the token. Two different types of dictionaries are used in Gimli:

- Gene and protein names: BioThesaurus is the most complete and up-to-date lexical resource for gene and protein names, containing almost 17 million unique names. Due to its size, we decided to filter this lexicon considering only human genes and proteins, obtaining almost 400 thousand unique names. In the end, this lexicon is used to indicate the presence of curated gene and protein names. Since these names could be present in text with small orthographic variations, the matching is performed according the following variation rules, adapted from [123]:
  - Replace white spaces per hyphens, and vice-versa;
  - Remove white spaces and hyphens;
  - Insert an hyphen on letter-digit sequences;
  - Replace Roman by Arabic numbers, and Arabic numbers by Greek letters;
  - Add the prefix "h" and the suffix "p" to acronyms
- Trigger words: specific domain terms may indicate the presence of biomedical names in the surrounding tokens. Instead of using words from training data as proposed in [163], we apply a more general solution, by matching the terms in BioLexicon. This lexical resource contains more than two million relevant biomedical terms, including nouns, verbs, adjectives and adverbs (e.g., "stimulate", and "activation").

Higher level relations between tokens and extracted features can be established through windows or conjunctions of features, reflecting the local context of each token. The application of windows consists of adding selected features from preceding and succeeding tokens as features of each token. On the other hand, conjunction of features consists of creating new features by grouping together features of the surrounding tokens. For instance, considering the sentence "Pharmacologic aspects of neonatal hyperbilirubinemia." and a {-1,1} range of tokens, the following features are added to the token "neonatal":

- Windows: the tokens "of" and "hyperbilirubinemia";
- Conjunctions: the new conjunction feature "of@-1_&_hyperbilirubinemia@1".

Our tests showed that the best results were obtained using conjunctions. However, Gimli does not use all of the features to generate conjunctions, since this would become impracticable, generating millions of new features. Tsai et al. [137] proposed the use of tokens from the following windows to generate the conjunctions: {-3,-1}, {-2,-1}, {-1,0}, {-1,1} and {0,1}. To improve the context knowledge, we propose a different approach, using lemmas and POS tags instead of tokens, since lemma conjunctions better reflect the pairwise patterns of words, and

the POS tags conjunctions provide grammar-based relations and patterns. Following the previous example, instead of the simple token-based conjunction feature, the token "neonatal" now has two conjunction features: POS=IN@-1_&_POS=NN@1 and LEMMA=of@-1_&_LEMMA=hyperbilirubinemia@1. The benefits of these choices were confirmed through various experiments.

3.2.3 Model

When ML techniques are applied to NER, an algorithm must build a feature-based statistical representation of target entity names from training data, in order to develop an appropriate response to unseen data. Such methodologies are commonly categorized as being supervised or semi-supervised. Semi-supervised solutions use both annotated and unannotated data, in order to derive features of the entity names that are not present in the annotated data. Specifically for this task, the usage of unannotated data could contribute to a better abstract learning of the named entities. However, the application of such techniques is computationally heavy and could be implemented as an extension to an equivalent supervised solution. Thus, we decided to follow a supervised training approach, through the application of CRFs [139], which were previously described in detail in Section 2.2.2.

The most recent results on biomedical NER clearly indicate that better performance results can be achieved by combining several systems with different characteristics. As an example, the top five systems of the BioCreative II gene mention challenge [90], used ensembles of NER systems, combining distinct models or combining models with dictionary and/or rule-based systems. Additionally, the application of machine learning-based harmonization solutions have been shown to deliver high improvements in terms of performance results [30].

We propose a new and simple combination strategy based on confidence scores. To achieve this, each model provides a confidence value for the annotations predicted for a given sentence. If the models that produced the overlapping annotations predict the same entity class, we follow a straightforward strategy, selecting the annotations from the model that has the highest confidence score and rejecting the predictions of other model(s). On the other hand, if we need to combine annotations of models that predict different entity classes (e.g., as in the JNLPBA corpus), this strategy is extended in order to allow distinct entity types in the same sentence. Thus, instead of selecting a single model to provide the predictions for the entire sentence, this choice is made for each annotation in the sentence. When two or more models provide different annotations for the same chunk of text, we select the annotation given by the model with the highest confidence score. If only one model provides an annotation for a chunk of text, that annotation is accepted.

3.2.4 Post-processing

In order to solve some errors generated by the CRF model, Gimli integrates a post-processing module that implements parentheses correction and abbreviation resolution. To perform parentheses correction, the number of parentheses (round, square and curly) on each annotation is verified and the annotation is removed if this is an odd number, since it clearly indicates a mistake by the ML model. We also tried to correct the annotations by removing or adding tokens up to the next or previous parenthesis. However, this solution provided worse results than simply removing the annotations.

Regarding abbreviation resolution, we adapt a simple but effective abbreviation definition recognizer [112], which is based on a set of pattern-matching rules to identify abbreviations and their full forms. Such patterns consider some constraints, namely: *a*) the first character of the acronym has to be the first character of the first word in the corresponding long form; *b*) the long form should be longer than the corresponding acronym; and *c*) the long form should not contain the candidate acronym. In the end, we are able to extract both short and long forms of each abbreviation in text. Thus, if one of the forms is annotated as an entity name, the other one is added as a new annotation. Additionally, if one of the forms is not completely annotated, Gimli expands the annotation boundaries using the result from the abbreviation extraction tool.

3.3 Results

To analyze the impact of various techniques and compare the final results with other existing solutions, we use common evaluation metrics: Precision (i.e., positive predictive value) the ability of a system to present only relevant items; Recall (i.e., sensitivity) the ability of a system to present all relevant items; and F-measure, the harmonic mean of precision and recall.

3.3.1 Corpora

There are several publicly available corpora that can be used for training and evaluation of NER systems. To allow direct comparison with other tools, we selected two of the most commonly used corpora: GENETAG and JNLPBA. GENETAG [92] is composed of 20000 sentences extracted from MEDLINE abstracts, not being focused on any specific domain. It contains mentions of proteins, DNAs and RNAs (grouped in only one semantic type), annotated by experts in biochemistry, genetics and molecular biology. This corpus was used in the BioCreative II challenge [90], providing 15000 sentences for training and 5000 sentences for testing. On the other hand, the JNLPBA corpus [91] contains 2404 abstracts extracted from MEDLINE using the MeSH terms “human”, “bloodcell” and “transcription factor”. The manual annotation of these abstracts was based on five classes of the GENIA ontology [195],

namely protein, DNA, RNA, cell line, and cell type. This corpus was used in the Bio-Entity Recognition Task in BioNLP/NLPBA 2004 [91], providing 2000 abstracts for training and the remaining 404 abstracts for testing.

Since GENETAG is not focused on any specific biomedical domain, its annotations are more heterogeneous than those of JNLPBA. A brief analysis, considering protein, DNA and RNA classes, shows that GENETAG contains almost 65% of unique entity names, as opposed to the 36% found in JNLPBA.

3.3.2 Preliminary experiments

During the development of Gimli, various optimizations and decisions had to be performed to achieve the best possible results. In order to run such experiments, we randomly split the training part of each corpus into training and development sets, using 80% of the data for training and the remaining 20% for development testing. Accordingly, from the 15000 sentences of the training part of GENETAG, 12000 sentences were used for training and 3000 sentences for development testing. Regarding JNLPBA, considering the 2000 training abstracts, we used 1600 abstracts for training and the remaining 400 abstracts for development testing. Most experiments on the development stage, namely tokenization and feature set optimization, were performed using first-order CRF models with forward (left to right) text parsing.

Tokenization

To evaluate the impact of the tokenization changes introduced in Gimli, we compared the results achieved against the use of the original tokenization. This analysis only applies to the GENETAG corpus, since JNLPBA is provided as tokenized text. Using the development set, an improvement of 8.28% in F-measure was achieved when applying a model trained on tokens provided by our proposed tokenization as compared to using the original version of GENIA Tagger. When applied to the final test set, and considering the alternative annotations provided, the improvement in F-measure was 2.53%. Such results clearly show the positive contribution of our tokenization approach on Gimli.

Feature set

Each feature encodes specific characteristics of target annotations, providing a different contribution in the learning process. In order to evaluate their impact in the recognition performance, we initially grouped features that encode similar information into logical sub-classes for each feature type, as shown in Table 3.2. We then followed a backward elimination approach to find the best feature set for each entity type, by removing each sub-class from the complete feature set and analyzing its impact in the results. Although small improvements or

drops may not be significant regarding performance improvements, they indicate that adding or removing a specific feature may have an impact on the final performance results, which is relevant when considering the inclusion (or not) of that feature. When such cases occurred, we decided to keep the feature when a small improvement occurred and remove it when a small drop was present. In the end, the features that presented a negative impact when removed from the initial set were included in the final feature set, as indicated in Table 3.2. For instance, our trigger words approach provided a slight positive impact in the recognition of gene and protein names in GENETAG, resulting in an F-measure improvement of 0.11%. However, a negative impact was observed on JNLPBA, with a 0.39% decrease of F-measure. We believe that the obtained results are a consequence of the corpus specificity, since BioLexicon terms may point to the presence of entity names that were not considered in the specific corpus and/or entity type.

Table 3.2: Feature set applied to each corpus and entity type. Features marked with an "X" are used in the final feature set for that entity type.

		GENETAG	JNLPBA				
		Protein	Protein	DNA	RNA	Cell Type	Cell Line
Base	Token	X		X	X		
Orthographic	Capitalization	X	X	X		X	X
	Counting	X	X	X	X		X
	Symbols	X	X	X	X		X
Linguistic	Lemma	X	X	X	X	X	X
	POS	X	X	X	X		X
	Chunk	X	X		X	X	X
	Dependency Parsing	X	X	X	X	X	X
Morphological	Char n-grams		X	X		X	
	Suffix	X	X	X	X	X	X
	Prefix	X	X	X	X	X	X
	Word Shape	X	X	X	X	X	X
Lexicons	Gene/Protein	X	X				
	Trigger Words	X					
Local Context	Conjunctions	X	X	X	X	X	X

The final feature sets seem to reflect the complexity and heterogeneity associated with each entity type and corpus, and may help experts to better understand the linguistic characteristics of each entity type on each corpus. For instance, the absence of the original tokens for protein, cell line and cell type on JNLPBA may indicate less heterogeneity, as the use of lemmatization appears to better reflect and generalize the target names. Overall, the feature set required by GENETAG is more complex than the ones used on JNLPBA, discarding the original tokens and some orthographic and morphological features. This is consistent with the idea that the entity names present on GENETAG are more heterogeneous than those present on JNLPBA, as suggested before.

Local context

Local context, as encoded in windows or conjunctions of features, has a great impact in recognition performance. We therefore analyzed in detail the impact of using these two

alternatives, considering basic and improved solutions. Thus, four different configurations were considered in our analysis:

- Token conjunctions: form conjunctions as the concatenation of tokens taken from the following windows {-3,-1}, {-2,-1}, {-1,0}, {-1,1} and {0,1};
- Optimized conjunctions: the same windows as the previous configuration but using lemmas and POS tags for the conjunctions, instead of tokens;
- Windows tokens: use each token from the window {-2,2};
- Windows optimized: use lemmas, lexicon matching, biomedical concepts matching and tokens in the window {-3,3}, and all the features in the window {-1,1}.

Table 3.3 presents the performance (F-measure) achieved with the four approaches. Results are shown for CRF models of order 1 and 2 with forward and backward parsing directions, as explained in the next section. Optimized conjunctions present the best results on both corpora, considerably outperforming conjunctions with tokens. Conjunctions of features seem to perform better than windows for this task, as indicated by the fact that using simple token conjunctions provided better results than even the optimized windows of features. Interestingly, while the optimized windows present better results than windows with tokens on GENETAG, in the case of JNLPBA using just the tokens provides better results for the models trained with backward parsing direction. Overall, optimized conjunctions present the most constant behavior, presenting the best results and less deviation. On the other hand, using tokens resulted in higher deviation on both approaches.

This analysis indicates that choosing the right method to encode local context is fundamental, since a wrong decision may deliver considerably worse results. As we can see, the average F-measure differences between the best and worst solutions on GENETAG and JNLPBA are of 2.13% and 1.73%, respectively.

Table 3.3: Comparison of F-measure results achieved by token-based and optimized windows and conjunctions in the development sets of both corpora, considering exact matching evaluation, different model orders and text parsing directions. Results for the JNLPBA corpus indicate the overall performance, i.e. across entity types. FW: Forward, and BW:Backward.

		GENETAG					JNLPBA				
		Order 1		Order 2			Order 1		Order 2		
		FW	BW	FW	BW		FW	BW	FW	BW	
Conjunctions	Optimized	**77.46%**	**76.80%**	**77.83%**	**78.53%**	77.65% ± 0.73%	**75.75%**	**76.29%**	**75.38%**	**76.06%**	75.87% ± 0.39%
	Tokens	-2.64%	-1.37%	-0.82%	-0.53%	76.32% ± 1.45%	-0.89%	-0.05%	-1.16%	-0.34%	75.26% ± 0.90%
Windows	Optimized	-2.11%	-1.64%	-1.04%	-1.85%	75.99% ± 0.86%	-0.58%	-2.15%	-1.93%	-2.27%	74.14% ± 0.74%
	Tokens	-3.20%	-2.47%	-1.47%	-1.42%	75.52% ± 1.44%	-1.30%	-0.92%	-2.38%	-1.47%	74.35% ± 0.99%

3.3.3 Model combination analysis

The usual direction to parse a text is from left to right (forward). However, previous studies [136, 278] have shown that parsing the text from right to left (backward) may provide

better results, which has been shown to be a consequence of the asymmetric implementation of CRF models in MALLET [136]. Additionally, we believe that using CRFs with different orders will extract different context based characteristics from text. Thus, we decided to train first and second order CRF models, considering both forward and backward text parsing.

Initial evaluation results on GENETAG and JNLPBA are presented in Table 3.4, using the previously selected feature set (Table 3.2). As we can see, the application of different CRF orders and parsing directions provides significant performance differences. For instance, considering RNA on JNLPBA, the difference between different parsing directions is above 3% of F-measure, and the difference between different CRF orders is approximately 2% of F-measure. Overall, backward models presented the best results, which confirms the benefit of using backward text parsing. Moreover, due to the names' heterogeneity existent in both corpora, different model orders are required. On GENETAG, the best results are achieved using second order models. On the other hand, the best results for protein and cell type on JNLPBA were achieved using first order models.

Table 3.4: Preliminary F-measure results on development sets.

GENETAG

		Order 1	Order 2
Protein	FW	77.46%	77.83%
	BW	76.80%	**78.53%**

JNLPBA

		Order 1	Order 2
Protein	FW	80.10%	79.44%
	BW	**80.33%**	79.82%
DNA	FW	68.19%	68.19%
	BW	69.18%	**70.25%**
RNA	FW	75.35%	**77.27%**
	BW	75.92%	73.71%
Cell Type	FW	71.45%	71.28%
	BW	**73.02%**	72.47%
Cell Line	FW	67.81%	**68.10%**
	BW	67.77%	67.62%

To combine the various models for each class on each corpus, we performed a sequential analysis of the combination results. Thus, we first combined the two best models for each class and, if the performance was better than the best model alone, we kept adding models to the two best, in order to find the best set. If the combination result of the two best models was not better than the best model, we tried combining the best model with others, until a better combination was obtained. If the combination did not improve the results, only the model with the best result was used. Table 3.5 presents the results of our analysis. Even with the simple combination approach used by Gimli, the harmonization strategy improved the best model results, with an average improvement of 0.5% of F-measure. Overall, the best

combination results were achieved by combining the two best performing models. Moreover, models with low performance results also contributed to a better model combination, by providing heterogeneity that is not present in other models. For instance, on cell line the best model combination was achieved by including the worst performing model.

Table 3.5: Combination results on development sets.

GENETAG

	Order 1		Order 2		
	FW	BW	FW	BW	F1
Protein				X	78.53%
			X	**X**	**79.00%**
	X		X	X	78.81%
		X	X	X	78.82%
	X	X	X	X	78.87%

JNLPBA

	Order 1		Order 2		
	FW	BW	FW	BW	F1
Protein		X			80.33%
	X	X			80.80%
	X	**X**		**X**	**80.81%**
	X	X	X		80.61%
		X		X	80.34%
DNA				X	70.25%
		X		**X**	**70.38%**
		X	X	X	69.86%
	X	X		X	70.32%
RNA			X		77.27%
		X	X		76.49%
	X		**X**		**77.62%**
	X	X	X		77.18%
	X		X	X	76.84%
Cell type		X			73.02%
		X		**X**	**73.19%**
	X	X		X	72.85%
		X	X	X	72.49%
Cell line			X		68.10%
	X		X		68.73%
	X	X	X		68.39%
	X		**X**	**X**	**69.48%**
	X	X	X	X	68.96%

Table 3.6 presents the final results achieved on both corpora, considering the final and unseen test data of both corpora. Note that the evaluation strategies of the two challenges are slightly different. On JNLPBA only full matches are considered correct, requiring both left and right boundaries to match exactly. On the other hand, GENETAG evaluation allows minor mistakes, based on alternative names that were previously accepted by human annotators during the preparation of the corpus.

3.3.4 Feature contributions

In order to evaluate the overall contribution of some high-end features implemented by Gimli, we performed an analysis on both corpora, considering the removal of such features

Table 3.6: Final Precision (P), Recall (R) and F-measure (F1) results achieved by Gimli on test data of both corpora.

GENETAG

	Protein
P	90.22%
R	84.32%
F1	**87.17%**

JNLPBA

	Protein	DNA	RNA	Cell Type	Cell Line	**Overall**
P	71.53%	74.56%	68.42%	80.44%	61.54%	72.85%
R	78.11%	64.68%	66.10%	62.73%	56.00%	71.62%
F1	74.68%	69.27%	67.24%	70.49%	58.64%	**72.23%**

from the best feature set for each entity type. Table 3.7 presents the observed differences, reflecting the features' contribution. Overall, removing conjunctions caused the highest negative impact, considerably reducing the performance results. Dependency parsing also contributed positively to the final results, namely on DNA and cell line. On the other hand, removing dependency parsing features improved the results for the RNA entity type. However, this is a consequence of the algorithm to combine the models of different entity types. When evaluated alone, RNA recognition presents an F-measure of 68.97%. Removing dependency parsing features, this value drops slightly to 68.91%, reflecting the positive contribution of such features. As expected, lexicons also provide a positive contribution, increasing the models' precision. Post-processing, on the other hand, introduces just a small positive contribution. For instance, on RNA, the absence of post-processing methods does not affect the performance in any way.

Table 3.7: F-measure contribution of key features on GENETAG and JNLPBA considering all semantic types.

	GENETAG	**JNLPBA**				
	Protein	**Protein**	**DNA**	**RNA**	**Cell type**	**Cell line**
Best performance	**87.17%**	74.68%	69.27%	67.24%	70.49%	58.64%
-External resources	-0.28%	-0.42%	-	-	-	-
-Dependency parsing	-0.07%	-0.27%	-1.18%	0.28%	-0.23%	-0.63%
-Conjunctions	-1.16%	-2.05%	-3.34%	-0.57%	-1.11%	-0.94%
-Post-processing	-0.12%	-0.06%	-0.07%	0.00%	-0.04%	-0.12%

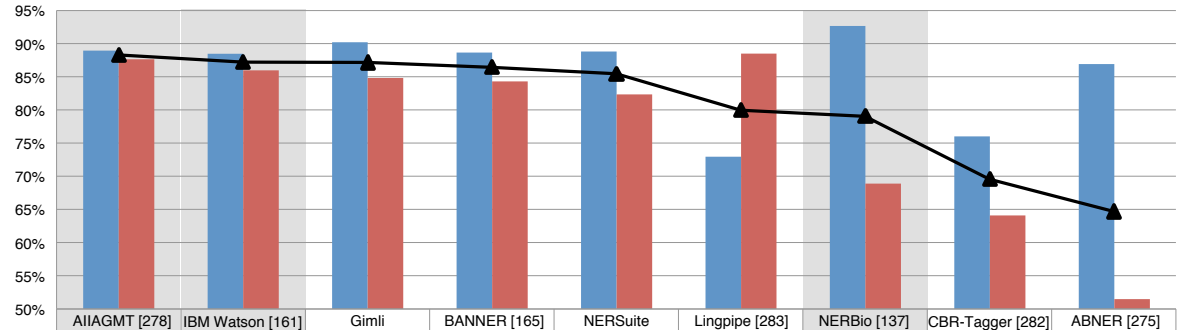
3.3.5 Performance analysis

To evaluate Gimli and understand its behavior in comparison with existing solutions, we collected the best open and closed source systems for biomedical named entity recognition. Table 3.8 presents a summary description of the systems' characteristics, comparing them against Gimli. Overall, we collected a total of 12 systems, where seven are open source and five closed source. Our study of these systems allowed to identify some current trends of biomedical NER systems:

- The most used ML model is CRF (6 systems);
- Almost all the discriminative ML models use orthographic, morphological and basic-linguistic (POS tags and lemmas) features;
- Only 3 systems use model combination, all of which are closed source;
- Only 5 systems use post-processing techniques, where 4 are closed source.
- 8 systems provide results on GENETAG and 6 on JNLPBA;
- Only 3 systems provide results on both corpora, where 2 are open source;

Based on these facts, we can argue that closed source solutions are commonly developed for a specific corpus, being focused on only one specific goal. However, those solutions present the most advanced techniques. On the contrary, open source solutions do not always take advantage of high-end techniques.

Figures 3.2 and 3.3 present the results obtained on GENETAG and JNLPBA corpus respectively, comparing Gimli against open and closed source systems. On the GENETAG corpus, Gimli outperforms all the open source solutions, achieving an F-measure of 87.17%. It presents an improvement of 0.74% over the second best system, BANNER. In comparison with NERSuite[1], Gimli presents an improvement of 1.72%. Overall, it presents the best results both on precision and recall. Considering closed source solutions, Gimli presents the third best result, with a similar performance as the winner of the BioCreative II Gene Mention challenge [90] (IBM Watson), which uses semi-supervised ML and forward and backward model combination. Overall, AIIAGMT [278] presents the best result on this corpus (with 88.30% of F-measure). However, the presented solution was prepared specifically for this corpus, applying a complex combination strategy that requires eight different CRF models using two different CRF frameworks.



	AIIAGMT [278]	IBM Watson [161]	Gimli	BANNER [165]	NERSuite	Lingpipe [283]	NERBio [137]	CBR-Tagger [282]	ABNER [275]
P	88.95%	88.48%	90.22%	88.66%	88.81%	72.95%	92.67%	76.01%	86.93%
R	87.65%	85.97%	84.82%	84.32%	82.34%	88.49%	68.91%	64.11%	51.49%
F1	88.30%	87.21%	87.17%	86.43%	85.45%	79.97%	79.05%	69.56%	64.68%

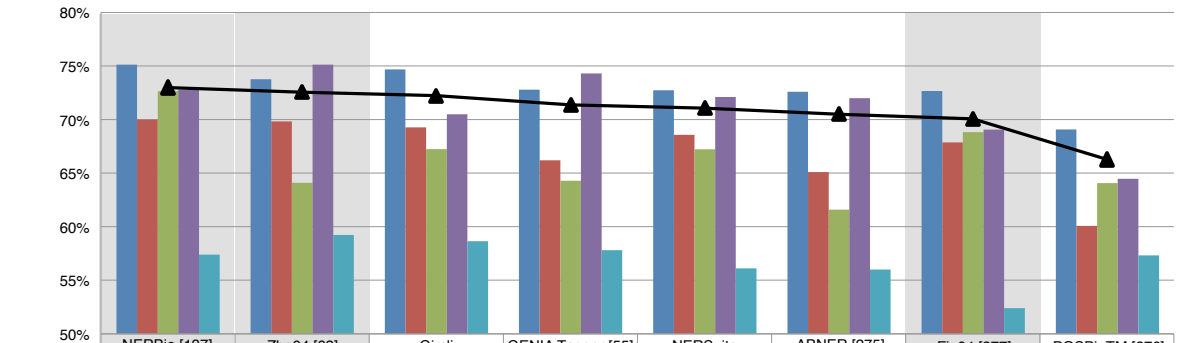
Figure 3.2: Comparison of the Precision (P), Recall (R) and F-measure (F1) results achieved by Gimli on GENETAG corpus, comparing with both open and closed source solutions. Results of closed source solutions are shown with a shaded background.

[1] `http://nersuite.nlplab.org`

Table 3.8: Summary of the open and closed source systems' characteristics, presenting the used programming languages, features, models and post-processing techniques. CBR-Tagger [282] and Lingpipe [283] were also included in this analysis.

		Open Source								Closed Source				
		2005 ABNER	2008 BANNER	2008 CBR-Tagger	2005 GENIA Tagger*	2012 Gimli	2007 Lingpipe	2010 NERSuite*	2004 POSBioTM	2008 AIIAGMT	2004 Fin04	2007 IBM Watson	2006 NERBio	2004 Zho04
Reference		[275]	[165]	[282]	[55]	-	[283]	-	[276]	-	-	[161]	[137]	[63]
Programming Language		Java	Java	Java	C++	Java	Java	C++	Java	-	-	-	-	-
Corpora	GENETAG	X	X	X		X	X	X		X		X	X	
	JNLPBA	X			X	X		X	X		X		X	X
Features	Orthographic	X	X		X	X		X	X	X	X	X	X	X
	Morphological	X	X		X	X		X	X	X	X	X	X	X
	Linguistic		X		X	X		X	X	X	X	X	X	X
	Context	X	X		X	X		X		X	X	X	X	
	Lexicons		X			X				X	X	X		X
Model	CRF	X	X			X		X	X	X			X	
	MEMM				X						X			
	HMM						X							X
	SVM													X
	CBR			X										
	ASO											X		
	Semi-supervised											X		
	Combination					X				X		X		X
Post-Processing	Parentheses		X			X				X		X		
	Abbreviation		X			X								X
	Lexicon									X				
	Pattern-based												X	

*No complete information is available. Extracted from source code analysis.



	NERBio [137]	Zho04 [63]	Gimli	GENIA Tagger [55]	NERSuite	ABNER [275]	Fin04 [277]	POSBioTM [276]
Protein	75.12%	73.77%	74.68%	72.79%	72.74%	72.60%	72.67%	69.07%
DNA	70.00%	69.83%	69.27%	66.20%	68.58%	65.10%	67.86%	60.08%
RNA	72.65%	64.10%	67.24%	64.29%	67.23%	61.60%	68.83%	64.07%
Cell Type	72.77%	75.13%	70.49%	74.31%	72.11%	72.00%	69.06%	64.48%
Cell Line	57.39%	59.23%	58.64%	57.81%	56.11%	56.00%	52.40%	57.33%
Overall	72.98%	72.55%	72.23%	71.37%	71.07%	70.50%	70.06%	66.28%

Figure 3.3: Comparison of the F-measure results achieved by Gimli on JNLPBA corpus, comparing with both open and closed source solutions. The overall result reflects the achieved performance considering the five entity types. Results of closed source solutions are shown with a shaded background.

Considering the JNLPBA corpus, Gimli outperforms all the open source solutions, achieving an overall F-measure of 72.23%. It presents an F-measure improvement of 0.86% in comparison with the second best system, GENIA Tagger. Compared to the best java-based solution (ABNER), Gimli presents an improvement of 1.73% of F-measure. It considerably outperforms open source systems in recognition of protein, DNA, RNA and cell line names. However, it is outperformed in the recognition of cell types.

Considering closed source solutions, Gimli presents the third best result, with similar results as the winner of the NLPBA Shared Task [91] (Zho04). When compared with the second best participant of this challenge (Fin04), Gimli presents an overall improvement of 2.17% of F-measure. NERBio, the best system on this corpus, implements a rule-based post-processing method that was prepared specifically for this corpus. Moreover, NERBio presents a very low performance result (79.05% of F-measure) on GENETAG, which could indicate some limitations in adapting this solution to different corpora.

Considering a non-blind model combination strategy, as taken by Hsu et al. [278], Gimli presents slightly better results, achieving an F-measure of 87.36% on GENETAG and 72.69% on JNLPBA. Such results outperform all the systems that participated on both challenges.

Overall, Gimli significantly outperforms all the existent open source solutions on both GENETAG and JNLPBA, by simply adapting the feature set used for each corpora and entity type. Moreover, it also presents competitive results when compared with similar closed source solutions for both corpora.

3.3.6 Chemical name recognition

Beyond the versatility already showed by Gimli by presenting high-performance results in the recognition of different concept types from different corpora, we decided to further test Gimli by participating in the BioCreative IV CHEMDNER task.

There is an increasing research interest in facilitating the access to information regarding chemical compounds and drugs described in text repositories [284]. Chemical and drug concepts are among the most frequently searched concepts in MEDLINE, a direct consequence of their important impact on chemistry, biology and medicine. However, there are various challenges that hinder the wider development of such solutions, such as the lack of suitable training and evaluation data, the difficulty in defining annotation guidelines of what actually constitutes a chemical compound or drug, and the heterogeneity in terms of scope and used textual data sources [104]. The BioCreative IV CHEMDNER task [104] was organized to address these issues, promoting the development of solutions to perform automatic recognition of mentions of chemical compounds and drugs on scientific documents, which is a challenging and complex task. Two different sub-tasks were defined:

- Chemical Entity Mention recognition (CEM): to provide, for a given document, the start and end indices corresponding to all mentioned chemical entities;
- Chemical Document Indexing (CDI): to provide, for a given document, a ranked list of mentioned chemical entities.

In order to participate in both sub-tasks, we took advantage of Gimli to deliver a machine learning-based solution using the provided manually annotated corpus[2], which is divided in three sets: train, development and test. The train set contains 3500 abstracts annotated with 29478 chemical annotations, and the development set contains 3500 abstracts with 29526 chemical annotations. Annotations are provided in seven classes: systematic, identifiers, formula, trivial abbreviation, family and multiple. However, we grouped all classes into a single "master" class. Finally, the test set contains 3000 abstracts.

Corpus pre-processing was performed applying the tools already integrated in Gimli. Thus, tokenization, lemmatization, POS tagging and chunking were performed using the custom version of GDep [28, 58]. Since documents were provided as abstracts, a tool to perform sentence splitting was required. Thus, we used Lingpipe[3] through a model trained on biomedical corpora.

Following the incremental strategy previously applied for feature selection, we collected the features with positive impact using the development set of the provided corpus. Table 3.9 presents the feature set used for the recognition of chemical names. In order to provide knowledge focused on chemical names recognition, we changed the used domain lexicons.

[2] http://www.biocreative.org/tasks/biocreative-iv/chemdner
[3] http://alias-i.com/lingpipe

Thus, dictionary matching was performed against a combined dictionary with terms from Jochem [120], ChEBI [83] and CTD [285]. As we can see, only capitalisation and dependency parsing features provided a negative impact, which reflects the heterogeneity and linguistic complexity of chemical names.

Table 3.9: Feature set applied in the recognition of chemical names. Features marked with an "X" were used in the final feature set.

		Chemicals
Base	Token	X
Orthographic	Capitalization	
	Counting	X
	Symbols	X
Linguistic	Lemma	X
	POS	X
	Chunk	X
	Dependency parsing	
Morphological	Char n-grams	X
	Suffix	X
	Prefix	X
	Word shape	X
Lexicons	Chemicals	X
Local Context	Conjunctions	X

In order to obtain models with heterogeneous characteristics and achieve improved results through combination, we considered CRF models with orders from 1 to 4. Since no significant performance variations were observed in the development set on models with orders 2, 3 and 4, and since training times are considerably lengthy in higher order models, we decided to use only models with orders 1 and 2. Annotations provided by such heterogeneous CRF models were improved by applying parentheses correction and abbreviation resolution post-processing modules, which also delivered improved performance results in chemical names recognition.

Combining annotations from heterogenous models follows the same algorithm previously applied to combine annotations from different concept types. Thus, the harmonization algorithm considers the confidence scores provided by each CRF model and selects the overlapping annotations with the highest scores. If an annotation does not intersect with others, it is added to the final list of annotations.

Since the CHEMDNER task required a confidence score for each predicted annotation, we developed a simple ranking algorithm. It is based on confidence scores provided by the CRF models, which is a value between 0 and 1 that reflects the certainty of the model generating each annotation. In that way, raking simply orders the annotations in descending order of scores. In the case of the CDI task, an additional filtering step was applied to remove repeated annotations with the same case-insensitive text. In the end, a list of unique text annotations is obtained.

Table 3.10 presents the final performance results achieved in the test set of the CEM and CDI sub-tasks. Our solution achieved high performance results on both entity mention

recognition and indexing, with 86.08% and 84.31% of F-measure, respectively. Comparing with the other 26 teams that participated in this challenge [104], our solution ranked in the 6^{th} and 7^{th} places in CEM and CDI, respectively. Considering ranking with statistically significant differences, we ranked at 4^{th} and 5^{th} in CEM and CDI, respectively. Moreover, the F-measure difference between our solution and the best performing solution in CEM is of just 1.31%.

Table 3.10: Precision, Recall and F-measure results achieved in the test set of CEM and CDI sub-tasks of the BioCreative IV CHEMDNER task.

	Precision	Recall	F-measure
CEM	86.50%	85.66%	86.08%
CDI	86.35%	82.37%	84.31%

The achieved performance results show that Gimli can be easily adapted for the recognition of heterogenous biomedical concepts, delivering state-of-the-art results. Nonetheless, further improvements may be developed to deliver even better results, using more and better domain knowledge, applying techniques for better context definition, and by taking advantage of an improved raking strategy.

3.3.7 Speed analysis

The various experiments to check training and tagging speed were performed in a machine with 8 processing cores @ 2.67 GHz and 16GB of Random Access Memory (RAM). The training speed varies with the corpus size, feature set complexity and model order. Considering the training parts of both corpora and the final feature set, a second-order CRF model takes on average one hour to be trained. On the other hand, a first-order CRF model requires on average 30 minutes. In order to check the tagging speed of Gimli, we developed a simple algorithm to annotate MEDLINE abstracts using multi-threading processing. This solution includes input Extensible Markup Language (XML) parsing, sentence splitting, Gimli integration and output generation in XML. It uses a single second-order CRF model, but model combination can be easily integrated with reduced speed impact, taking advantage of multi-threaded processing. During this analysis, we considered various configurations of Gimli, enabling and disabling the most resource expensive techniques. Thus, if users prioritize annotation speed over high performance results, windows can be used instead of conjunctions and dependency parsing can be removed from the workflow. Moreover, in order to use the available resources as much as possible, the number of running threads must be inversely proportional to the complexity of the used techniques, since complex techniques require more processing resources. The following results were obtained:

- Conjunctions with dependency parsing: 4 threads, 20 sentences/second;
- Conjunctions without dependency parsing: 6 threads, 86 sentences/second;
- Windows without dependency parsing: 8 threads, 232 sentences/second.

3.4 Discussion

Gimli is an off-the-shelf solution that can be used through two different endpoints, thinking on users with different goals and expertise:

- CLI: automatic scripts with easy access to main functionalities, allowing the annotation of documents using provided models, and training new models focused on different entity types, using a configuration file to customize the feature set and model parameters;
- Application Programming Interface (API): provides complete access to implemented features and associated infrastructure, allowing the easy integration of Gimli in complex text mining workflows, by using, extending and/or adapting the provided functionalities.

Overall, we believe that Gimli provides various characteristics that make it a state-of-the-art solution for biomedical NER:

- High-end techniques: Gimli applies various state-of-the-art techniques and proposes optimizations on various methods, presenting innovative and high-performance alternatives. Moreover, it integrates various solutions that are only present on closed source solutions, such as dependency parsing, chunking and model combination;
- Flexible: Gimli was built thinking on flexibility, founded on a strong infrastructure that allows adding new features and extending or changing existing ones. Moreover, Gimli offers the only CLI that allows feature set and model parameters definition;
- Scalable: the internal infrastructure is ready to scale, supporting the development of more complex solutions. Moreover, Gimli is ready to be used on multi-threaded applications, in order to process millions of documents;
- Documentation: we provide complete and detailed documentation of Gimli, in order to use both CLI and API. Together with the associated simplicity and self-explanatory code, we believe that Gimli is easy to use, change and extend.

Developers and researchers of the biomedical domain, especially text mining experts, can take advantage of the presented characteristics to develop their own NER and/or post-NER applications. Gimli reduces the required effort to develop innovative NER solutions, increasing the users' time to focus on their main goals. Thus, it can be used to support the development of various multi-disciplinary solutions: *a*) NER using different corpora and target entity names, such as disorders and chemicals; *b*) normalization; *c*) relation extraction, such as protein-protein interactions; and *d*) information retrieval.

3.5 Summary

This chapter presented Gimli (`http://bioinformatics.ua.pt/gimli`), a new open source and high-performance solution for biomedical named entity recognition on scientific documents, supporting the automatic recognition of gene/protein, DNA, RNA, cell line and cell type names. Gimli implements a machine learning-based solution, taking advantage of CRFs. Moreover, it supports a rich set of features, including orthographic, morphological, linguistic-based and also domain knowledge features, through the implementation of a lexicon matching technique. Additionally, Gimli implements advanced conjunctions of features, creating new features based on windows of lemmas and part-of-speech tags. Feature selection per concept type was performed by taking advantage of an incremental approach, analyzing the contribution of each feature type. In order to correct mistakes generated by the CRF models, Gimli also integrates a post-processing module, implementing parentheses correction and abbreviation resolution, aimed at extending incompletely tagged names. Finally, Gimli also supports the combination of several forward and backward models to achieve the best results.

In order to evaluate Gimli and compare it against existing systems, we used two well-known corpora: GENETAG and JNLPBA. In the end, it achieved F-measure results of 87.17% and 72.23% on each corpora, respectively. These results were compared to the systems that participated in the challenges where the corpora were used, BioCreative II Gene Mention and NLPBA Shared Task. Gimli outperforms all existing open source solutions on both corpora, presenting significant improvements both in results and techniques used.

Finally, Gimli was also applied in the recognition of chemical compound and drug names, as a participation in the BioCreative IV CHEMDNER task. With slight changes to adapt it to the chemical domain, Gimli delivered high performance results achieving 86.08% and 84.31% of F-measure in mention recognition and indexing, respectively. Such results show that Gimli is easily adapted to different concept recognition tasks delivering state-of-the-art results.

Chapter 4

Totum: biomedical named entity harmonization

This chapter is based on:

- D. Campos, S. Matos, I. Lewin, J. L. Oliveira, and D. Rebholz-Schuhmann, "Harmonization of gene/protein annotations: towards a gold standard MEDLINE." *Bioinformatics (Oxford, England)*, vol. 28, no. 9, pp. 1253–1261, May 2012

In Chapter 2, different approaches for NER were introduced, which can be categorized as being based on rules, dictionaries or machine learning. However, the most recent results clearly indicate that better performance can be achieved by using an ensemble of NER systems. As an example, the top five systems of the BioCreative II gene mention challenge used ensembles of NER solutions [90]. In these systems, each approach identifies entity mentions with different characteristics and based on different knowledge. Moreover, most of the NER solutions are trained and evaluated in only one corpus, which is usually focused in a specific biomedical domain and provides specific gene/protein names and contexts. As a consequence, when the system is applied to a corpus from a different domain, the global performance drops significantly. Although this occurs with machine learning approaches, it may also affect dictionary-based solutions, depending on the specificity of the used lexical resource. This is not only a consequence of the different domains, but also a result of the different annotation guidelines and their interpretation by human annotators. For instance, Colosimo et al. [286] presented a study with five thousand abstracts, obtaining an inter-annotators agreement of 87% for Fly, 91% for Yeast and 69% for Mouse in gene and protein names annotation.

In summary, various sources of variability can be identified in human annotated corpora: specific biomedical domain or sub-domain of the documents; annotation guidelines; and hu-

man annotators. Moreover, the different characteristics of NER systems introduce another source of variability for the harmonization task. As a result, considering the different underlying biological domains and the diversity of annotation types, combining gene/protein annotations from various systems is not a straightforward task. The harmonization method could take advantage of this variability, benefiting from the distinct background knowledge encoded by each system on each corpus, in order to obtain a more general solution, able to cope with the diversity of data found on a large-scale text repository such as MEDLINE.

This chapter presents Totum, an harmonization solution that addresses the problems of heterogeneous annotations. Section 4.1 presents the background of this work, and existent solutions for the combination problem. In Section 4.2 we present the proposed approach, and in Section 4.3 a comparison with state-of-the-art solutions, discussing the advantages and limitations.

4.1 Background

Nowadays, the annotation of biomedical documents is mainly performed manually by domain experts. Consequently, only small sets of documents have been manually annotated and made publicly available. The CALBC (Collaborative Annotation of a Large Biomedical Corpus) project intended to minimize this problem, providing a large-scale biomedical text corpus automatically annotated through the harmonization of several NER systems. This large corpus contains annotations of several biological semantic groups, such as diseases, species, chemicals and genes/proteins [108].

The CALBC corpus is focused in the immunology biomedical sub-domain, and is composed of abstracts collected from MEDLINE using the query "immunol*". To generate the first version of this corpus, four different NER and normalization systems were used:

- System 1: implements a dictionary-based approach that takes morphological variability into consideration. It uses several publicly available resources, such as Swiss-Prot [46] and ChEBI [83];
- System 2: applies a dictionary-based approach using Entrez Gene [66], Swiss-Prot, Genew [287], GDB [288] and OMIM [78] as terminological resources;
- System 3: implements a machine learning-based approach using CRFs, receiving orthographic and morphological features as input. It also integrates a dictionary-based step to identify gene mentions that were missed by the CRF. This system was trained using data from several corpora, including GENIA [195], PennBioIE [94], GENETAG [92], PIR [289] and AlMed [207]. In the end, the system performs normalization to provide identifiers for each gene/protein name;
- System 4: implements a dictionary-based solution, performing fuzzy matching and disambiguation to remove false positives.

These systems use different approaches to process the text, implementing different tokenization methods and/or strategies to deal with stopwords. Thus, we can argue that each system provides annotations with different characteristics, varying with the used techniques and resources. In order to take advantage of this variability, it is necessary to implement a method that will combine the several annotations, providing only one gene/protein name per chunk of text. To make this combination process possible, the several systems need to "speak and understand the same language". IeXML [290] facilitates such task, by defining an XML standard for representing abstracts, sentences and annotations. Using this cross corpus standard, we can combine the heterogeneous annotations, either by unifying and/or intersecting the annotations, or through the implementation of machine learning-based solutions.

Intersection requires the agreement of at least two systems for accepting an annotation, which improves precision but degrades recall. For instance, Torii et al. [291] presents a typical intersection solution to combine the annotations from four machine learning-based NER systems. Kuo et al. [136] presents another interesting solution to combine two CRF models, by intersecting the top ten adjacent annotations of each model and selecting the intersection with the best score. Union approaches, on the other hand, provide annotations performed by either one of the systems, improving recall but degrading precision. For instance, Ando [161] performs the union of two CRF models, removing annotations that overlap with longer ones.

Intersection and union solutions are widely used, due to the simplicity and positive outcomes of such methods. For instance, in the BioCreative II gene mention task [90], most of the participating systems that used an ensemble of systems applied intersection or union to combine the heterogeneous annotations. There are also solutions that use both techniques. For instance, Li et al. [292] and Hsu et al. [278] obtain the best results by intersecting the annotations of similar models, which are then unified to obtain a final set of annotations.

Machine learning-based solutions intend to learn the tokens' boundaries by experience, using manually annotated data for this purpose. The annotated data provides curated knowledge, which makes the decisions more accurate and supported. However, what makes this solution unique is also its biggest limitation, because manually annotated data is sparse in comparison with unannotated data, which could limit the learning window. Wilbur et al. [293] presents a machine learning solution to combine the annotations from the 19 NER systems that participated in the BioCreative II gene mention task, using a first order CRF with a simple set of features (tokens and systems' matches). Mika and Rost [164] present a different approach based on a weighted SVM, to perform the harmonization of three SVMs and one dictionary-based system. Both solutions presented positive results, by obtaining better performance in comparison with each system used in isolation. Even the systems with low performance contributed to an improved harmonization result, adding variability that was not provided by other NER systems. However, both approaches trained the models on the

same corpus being annotated, which demanded the use of a cross-validation strategy. During this process, both systems used almost the complete corpus for training purposes, which may create a model that is highly fitted to specific features of the training data. Consequently, the model could deviate from its target function, making it less effective when used in corpora with different characteristics.

The goal of the harmonization solution presented in this chapter is to provide automatic annotations for a large set of abstracts (almost one million) from MEDLINE, covering several sub-domains and organisms related to the immunology field. Since machine learning-based solutions improve both precision and recall through the usage of curated knowledge, our goal is to develop a solution less dependent on a specific corpus and able to annotate most of MEDLINE abstracts with high accuracy.

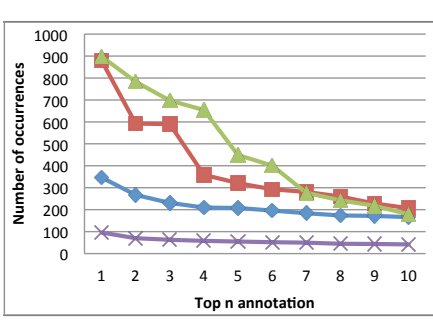
4.2 Methods

In order to develop a harmonization solution based on supervised machine learning, it is crucial to collect manually annotated data for the training procedures. To avoid the single corpus dependency, we used four of the biggest gold standard corpora, which cover different biomedical domains and organisms:

- **FSUPRGE** [93]: is a set of 3236 abstracts extracted from MEDLINE focused on gene regulation and expression, namely on regulatory events and all the components that are involved. The annotation process was semi-automatic, using a NER system that supports Active Learning (AL) to speed up the annotation process with no loss of annotation quality. During the AL process, the system selects the sentences that are expected to be more informative to the classifier, in order to be annotated by human experts;
- **JNLPBA** [91]: this corpus is a sub-set of the GENIA corpus, containing 2399 abstracts extracted from MEDLINE using the MeSH terms “human”, “bloodcell” and “transcription factor”. These abstracts were manually annotated based on the GENIA ontology. The JNLPBA corpus includes only five classes (protein, DNA, RNA, cell line, and cell type) from the 36 available in the GENIA ontology. Only the protein, DNA and RNA classes were used in this work;
- **PennBioIE**: is composed of several MEDLINE abstracts of two highly specialized biomedical sub-domains: the molecular genetics of cancer, and the inhibition of cytochrome P-450 enzymes. We use the oncology sub-set, which contains 1414 abstracts with annotations of proteins and RNAs;
- **GENETAG**: is composed of 20000 sentences extracted from MEDLINE abstracts, not being focused in any specific domain. It contains annotations of proteins, DNAs and RNAs, which were performed by experts from biochemistry, genetics and molecular

biology. This corpus was used in the BioCreative II challenge [90], providing 15000 sentences for training and 5000 sentences for testing. For this work, since the used systems implement normalization, it was necessary to find the original abstracts for each sentence. At the end, only 17590 sentences were used from the abstracts that were possible to collect without ambiguity.

Since each corpus is focused on a different goal and biomedical domain, the annotated entity names differ from corpus to corpus. The ten most frequent annotations (Figure 4.1) reflect this variability, presenting annotations that only appear in one corpus (e.g., overall, "KIR" only appears on FSUPRGE), and annotations shared by the corpora with significantly different proportions of occurrences (e.g., "NF-KappaB" is the most frequent annotation on JNLPBA, but only the eigth most frequent annotation on GENETAG and FSUPRGE). Moreover, the percentage of unique entity names is also different, which shows the entities sparseness and specificity of each corpus. For instance, since PennBioIE is focused on a very specialized sub-domain, the number of unique annotations in this corpus corresponds to just 18% of the complete set of annotations. On the other hand, since GENETAG is not focused on any sub-domain, 65% of its annotations are unique. Even when the proportion of unique entity names is not high, each corpus provides a unique set of names that is not available in any other corpora, delivering an extensive set of contexts where the gene/protein name could be found.



FSUPRGE	JNLPBA	PennBioIE	GENETAG	
KIR	NF-kappa B	K-ras	insulin	
cytokine	IL-2	beta-catenin	Ras	
IFN-gamma	NF-kappaB	p53	Sp1	
cytokines	IL-4	ras	p53	
IL-6	AP-1	N-myc	AP-1	
insulin	transcription factors	N-ras	CAT	
kinase	TNF-alpha	polymerase	JNK	
NF-kappaB	IFN-gamma	MYCN	NF-kappaB	
RNA polymerase	cytokines	APC	MAPK	
TNF-alpha	IL-10	H-ras	CREB	
57261	**45635**	**17524**	**21677**	**Corpus annotations (#)**
3.76%	**8.79%**	**27.39%**	**2.65%**	**Top 10 (%)**
26.57%	**36.38%**	**17.75%**	**64.59%**	**Unique annotations (%)**

Figure 4.1: Ten most frequent annotations on each curated corpus, reflecting the variability between the corpora. The percentage of unique annotations indicates the variability within each corpus. The highlighted annotation appear only on that specific corpus.

In order to obtain performance results, the corpora were divided into train and test sets. JNLPBA and GENETAG were already divided by the providers, using approximately 17% and 25% of the data for testing, respectively. On the other hand, PennBioIE and FSUPRGE were not divided, so we left 30% of the data for testing purposes. Since each corpus is provided in a specific format, all the data were converted to the IeXML format, creating one large corpus with 6566 abstracts for training and 2242 for testing.

After annotating the corpus using the four systems described in the Background section (S1-S4), there were several points of disagreement. Figure 4.2 shows some examples

that reflect this variability. For instance, some systems include the organism name in the gene/protein names and others do not (Figure 4.2: Example 1), which remains a point of active discussion among expert annotators. Other point of disagreement is the inclusion of the tokens "protein" or "gene" as suffix or prefix, causing the systems to have a different behavior (Figure 4.2: Example 3). Finally, there is also variability regarding the inclusion of greek letters in the entity names (Figure 4.2: Example 2).

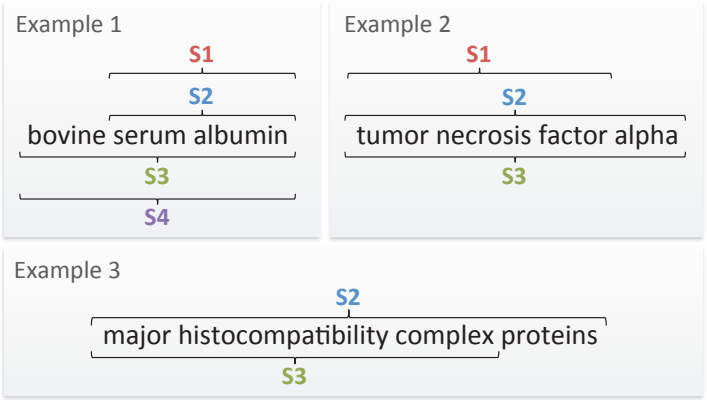


Figure 4.2: Examples of the annotations' variability provided by the four systems. Sn indicates the annotation performed by system n.

The observed annotations variability also leads to different performance results. Thus, it is important to understand that the performance results achieved by the used systems follow the behavior of publicly available solutions with similar characteristics. Consequently, we annotated the four corpora using six solutions, three based on machine-learning and three based on dictionaries. Kuo et al. [136] presents a CRF-based solution trained on GENETAG corpus, using orthographic and morphological features. It implements a bidirectional strategy, by combining two CRF models: one parsing the sentences from left to right (forward), and other parsing the sentences from right to left (backward). Another system is ABNER [275], which also applies CRFs trained on GENETAG corpus, using orthographic and morphological features. For the last ML-based solution, we trained ABNER on JNLPBA. Regarding dictionary-based solutions, the first one uses exact matching and BioThesaurus 7.0 [116] as the gene/protein names dictionary, removing uninformative terms that are not used in the scientific literature. The identification of the terms uses orthographic variability (e.g., "HZF[-]1" and "[Hh]zf[-]1") as described in [294]. The second solution is similar to the previous one, however it uses the Swiss-Prot subset of UniProt as the dictionary. After the matching process, basic disambiguation is performed through a specific term frequency associated with the term. The last solution also uses a disambiguation layer, but using BioThesaurus 7.0 instead.

Figure 4.3 compares these six public solutions with the four systems used in this work (S1-S4)[1], considering the four human annotated corpora and exact matching evaluation. In FSUPRGE two systems are above the average of public solutions, and the remaining are outside of the standard deviation range. Considering JNLPBA, one system is above the average, two are within the standard deviation, and one is outside that range. For GENETAG, one system is above the average of the public solutions, one is within the standard deviation range, and the remaining two are outside that range. Finally, all the used systems are above the average of the public solutions on PennBioIE. Remember that the ML-based solution that we use performs normalization, which does not happen on ML-based public systems. Thus, it is expectable that the ML-based public solutions provide better results, since the normalization step discards some names that were not possible to relate with unique identifiers.

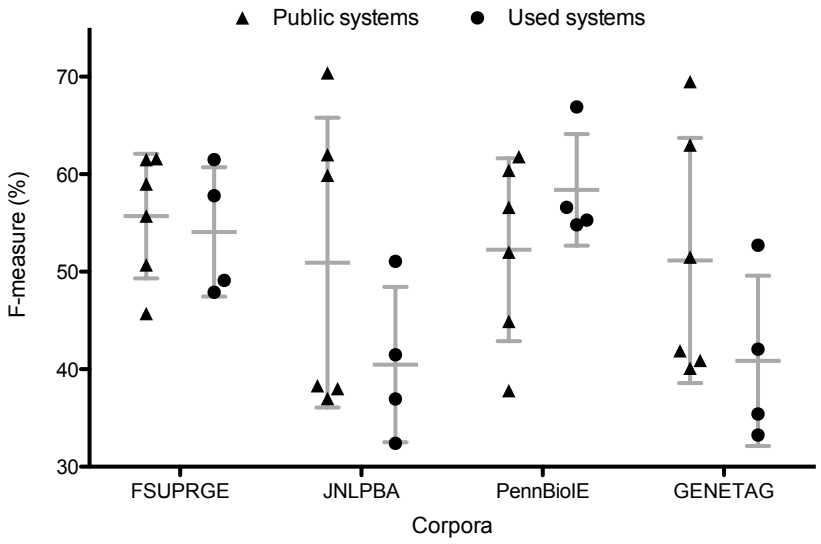


Figure 4.3: Comparison of systems S1-S4 against publicly available solutions, considering the four gold standard corpora, namely the whole set of FSUPRGE and PennBioIE and only the test parts of JNLPBA and GENETAG. The bars illustrate the mean and standard deviation of each set.

A brief analysis indicates that our set of systems follow the average behavior of the other solutions. In fact, a two-tailed non-parametric Mann-Whitney analysis showed no significant difference between the two sets of systems, resulting in p-values in the interval [0.2571; 0.9143].

After annotating the corpora with the four systems, since each system uses its own tokenization technique, we created a tokenization method compatible with all strategies, allowing the creation of a single data source that contains the systems' contributions and gold standard annotations. Such data source is in a CoNLL-like format [295], where each line contains six columns: token, BIO tags for each of the four systems, and gold standard BIO tag (Figure

[1]The used systems had to be anonymized due to project requirements.

4.4).

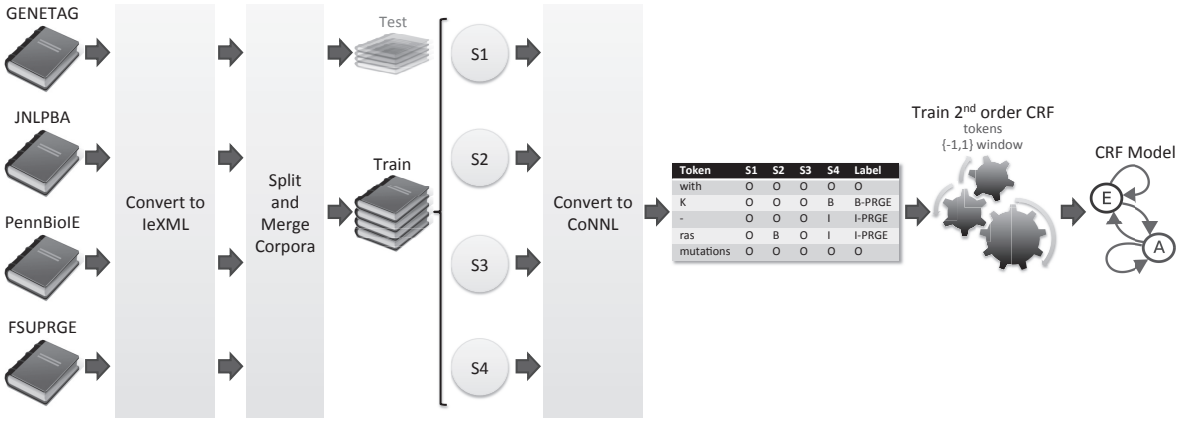


Figure 4.4: Illustration of the required steps to train the CRF model using the several corpora.

Using the data in the CoNLL-like format, we were able to train the machine learning model to harmonize gene/protein annotations. The training of such model may be supervised or semi-supervised, which use both annotated and unannotated data to obtain features of the entity names that are not present in the annotated data. Specifically for this task, the usage of unannotated data may contribute to a better abstract learning of the named entities boundaries. However, the application of such techniques is computationally heavy and may be performed as an extension to an equivalent supervised solution. Thus, we decided to use a supervised method, through the application of CRFs [139]. Initially, we applied a simple set of features: tokens, systems annotations tags, and a {-1,1} window of tokens to model local context. In order to optimize the set of features, we performed several experiments using POS, stemming, different window sizes, and different CRF orders. However, the performance always dropped and the initial set of features was kept. Figure 4.4 illustrates the workflow to convert the data and train the model.

This model can change the annotations' boundaries, remove incorrect annotations, and generate new annotations in comparison with the ones provided by the systems. However, if the systems being combined perform normalization, i.e., provide identifiers for the entities, creating new annotations may not be desirable, since assigning identifiers to such annotations will not always be possible. In order to create a Totum solution that does not create new annotations, we changed the training portion of the gold standard corpus, removing manual annotations, i.e. replacing the corresponding entity labels by "O", in those cases that were not identified as an entity by any of the four systems. Accordingly, we end up with a different gold standard corpus, adjusted to a different goal, which only contains gold standard entity labels where at least one system produced an entity output. This filtered version of the corpus contains 78% of the original gold standard annotations. Performing the CRF training in this new corpus, we get a new solution focused on changing the annotations boundaries

or removing incorrect ones. Furthermore, we also built a post-processing filter to remove new annotations, which may happen (not in significant proportions) since the model uses tokens as features to learn the boundaries. In the end, we provide two different solutions: one, identified as Totum, optimized for harmonizing annotations from NER systems, and the other, identified as TotumID, guided towards harmonizing the annotations and respective identifiers provided by normalization systems.

4.3 Results and discussion

4.3.1 Experimental setting

In order to obtain F-measure, precision and recall results that reflect the behavior of the several solutions, we have applied four matching techniques: exact, nested, and approximate matching using two different similarity thresholds. This detailed analysis is important since some post-NER tasks can be performed even if imprecise names are provided (e.g., relation extraction). Thus, we first performed exact alignment, which requires the boundaries of the entities to match exactly. Then, to perform approximate alignment, Inverse Document Frequency (IDF) scores of the tokens were calculated using the corpus of one million MEDLINE abstracts about immunology. With these scores, we can calculate a similarity value using the cosine between the two vectors of the tokens. For example, if annotator A1 annotates the phrase Pa = "T1 T2" and the annotator A2 the phrase Pb = "T1 T2 T3", there is no exact match. Thus, we consider the IDF scores of each token $fx = idf(Tx)$, calculating the cosine similarity between the vectors $v1 =< f1, f2, 0 >$ and $v2 =< f1, f2, f3 >$. A match is accepted if $cos(v1, v2)$ is equal or higher than a predefined value. In this work we used two different thresholds, 0.98 and 0.90. Finally, we also use nested alignment, in order to check when an annotation contains the boundaries of another.

To evaluate the performance of the two Totum solutions, we trained the harmonization model using the train part of the merged corpus, either with the original complete annotation set or with filtered annotations as explained in the previous section. This then allowed us to check the results on the unaltered test part of each corpus and on the merged test set, providing accurate information regarding the behavior of both solutions. Such solutions were compared against the two most common and state-of-the-art harmonization approaches: intersection (two vote agreement) and union (one vote agreement).

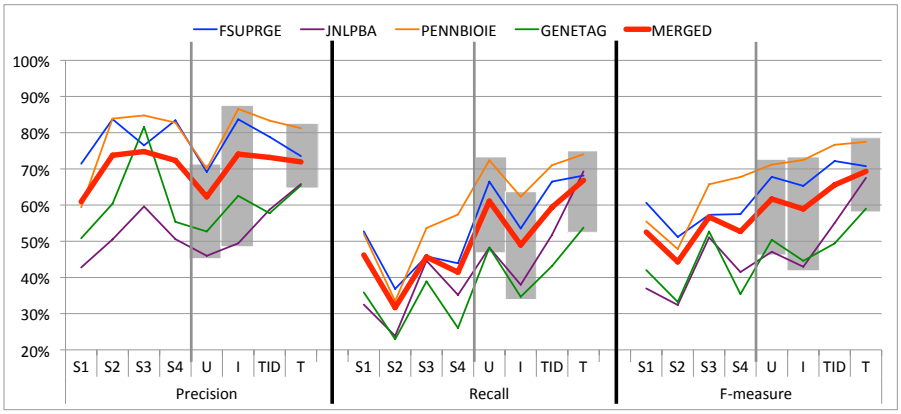
4.3.2 Performance analysis

Figure 4.5 presents an overview of the results obtained in the experiments, focusing on the comparison of Totum against Union and Intersection. Appendix A presents detailed and precise results. Overall, the harmonization solutions present better results than the average of the four systems. On the other hand, when comparing with the best performing system,

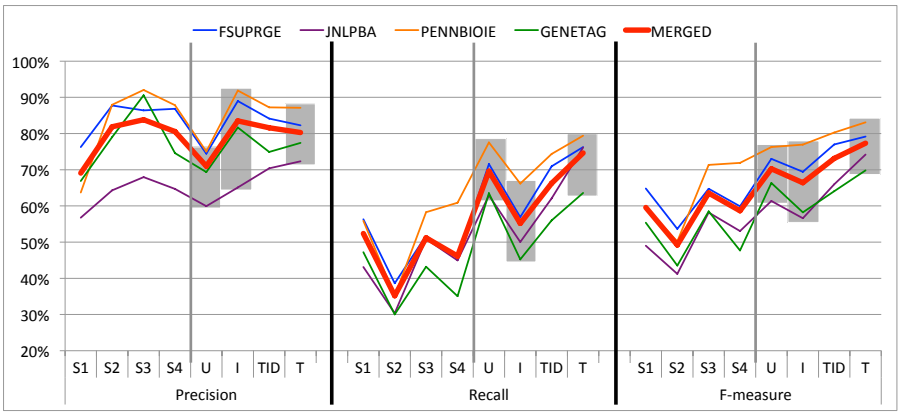
intersection and union have a better performance only on FSUPRGE, PennBioIE and Merged. Both Totum solutions present better results, with the exception of TotumID on GENETAG, which is outperformed if exact matching is considered.

Comparing the harmonization solutions, Totum significantly outperforms the other approaches. TotumID also presents better results than the two state-of-the-art solutions. Finally, union also presents better results than intersection. To analyze the improvements of both Totum approaches, we studied in detail the results achieved on the merged corpus, since it reflects better the global systems' behavior. Moreover, since there is no significant difference between the results of the two approximate matching technique, we only considered the cosine 0.98 alignment, which better expresses the process of discarding less informative tokens during the alignment. Therefore, comparing Totum with union, F-measure improvements of 7.61%, 7.06%, and 16.17% were obtained for exact (69.30%), approximate (77.34%) and nested (81.77%) matching, respectively. Against intersection, Totum achieved better performance by 10.34% for exact, 10.91% for approximate, and 22.25% for nested alignment. Comparing TotumID with union, it presents an improvement of 3.89% (65.58%) on exact, 2.83% (73.11%) on approximate and 5.22% (70.83%) on nested matching. Against intersection, TotumID presented better results, with improvements of 6.62%, 6.68%, 11.30% for exact, approximate and nested alignment, respectively. Considering the other corpora, Totum presented the best improvements on JNLPBA and less on PennBioIE. On the other hand, TotumID performed better on JNLPBA and worst on GENETAG, where it is slightly outperformed by union. Surprisingly, the best final results were achieved in the corpora for which we used smaller amounts of data for the training procedures (FSUPRGE and PennBioIE), which is a direct consequence of the better results achieved by the systems. In summary, both Totum approaches presented significant improvements in comparison with the two state-of-the-art solutions. However, the best results were achieved on nested alignment, which indicates that both Totum solutions provide longer names than the other approaches.

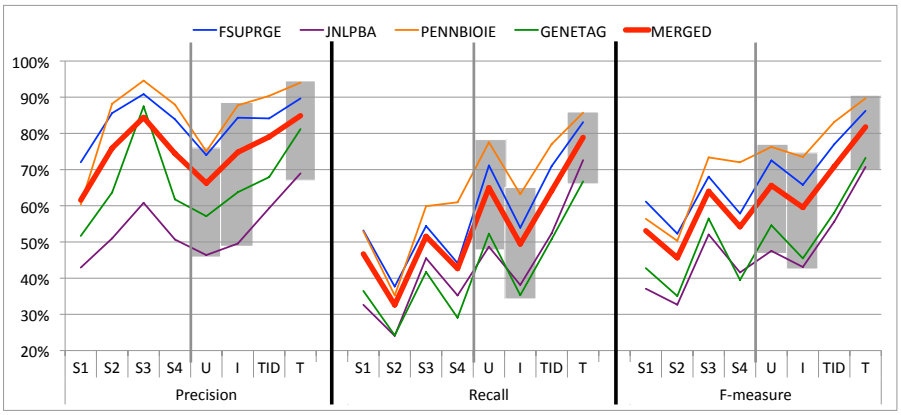
Regarding precision and recall, intersection presented better precision than union, since it uses two system votes to reach an agreement. On the other hand, union has better recall than intersection, because it only uses one vote. However, Totum presented better recall in all experiments. Thus, we can conclude that our solution is more sensitive than the other approaches, recognizing more entity names correctly. Regarding precision, Totum always performed better on nested matching. However, in the other matching techniques, intersection presented better precision. This means that our approach has increased specificity in comparison with the used systems and union. Overall, Totum significantly improved recall (sensitivity) in comparison with other approaches, with a small drop of precision (specificity) in comparison with intersection. Thus, we can argue that our solution deals better with heterogeneous annotations and features, considerably improving recall and with no precision loss.



(a) **Exact matching**



(b) **Cosine 0.98 matching**



(c) **Nested matching**

Figure 4.5: Overview of the results achieved by systems S1-S4 and harmonization solutions on the test parts of each corpus and on the merged test set, considering exact, cosine 0.98 and nested matching. The filled boxes indicate the range of performance results for Union, Intersection and Totum, across the five test sets. (Sn-System n; U-Union; I-Intersection; T-Totum; and TID-TotumID).

4.3.3 Annotations analysis

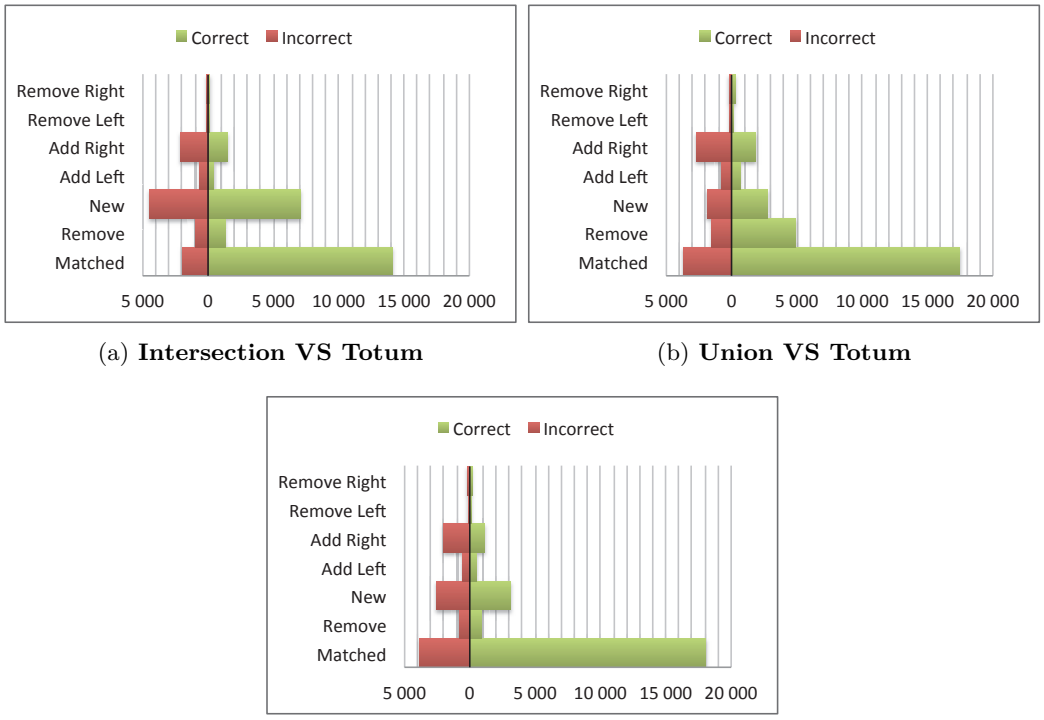
To understand the improved results provided by both Totum solutions, we have to study the generated annotations. Table 4.1 presents the number of annotations provided by the systems and harmonization solutions, when annotating the test parts of the gold standard corpora. System 1 provides more annotations than the other systems, which does not mean that it delivers the best results. Analyzing Figure 4.5, we can see that system 1 is outperformed by systems 3 and 4 in most of the corpora. The same pattern is verified in the harmonization solutions, where union presents the largest amount of annotations in almost all corpora. However, Totum provides the best trade-off between precision and recall, generating approximately the same number of annotations as in the gold standard corpus, and with fewer mistakes.

Table 4.1: Number of annotations generated by each system and harmonization solution in comparison with manually curated data, considering the test parts of the corpora. The highlighted boxes indicate the solution (and the harmonization method) that provided the higher number of annotations for each corpus.

	FSUPRGE	JNLPBA	PennBioIE	GENETAG	Merged
Gold	**17181**	**6142**	**5285**	**5716**	**34324**
System 1	12691	4670	4636	4034	26031
System 2	7562	2899	2109	2172	14742
System 3	10290	4599	3346	2725	20960
System 4	9043	4269	3663	2687	19662
Union	16524	6447	5460	5233	33664
Intersection	10986	4722	3805	3165	22678
TotumID	14025	5349	4660	4127	27866
Totum	16431	6467	5074	4694	31906

To analyze the generated annotations, we developed a tool to compare the exact annotations provided by two solutions, in order to study the changes promoted by solution *b* against solution *a*. We considered seven different categories of agreement and disagreement: Matched (the annotation is the same in the two solutions); New (the second solution adds an annotation that does not exists in the first one); Removed (the second solution removes an annotation provided by the first one); Add left (one or more tokens were added to the left side of the annotation); Add right (one or more tokens were added to the right side of the annotation); Remove left (one or more tokens were removed from the left side of the annotation); and Remove right (one or more tokens were removed from the right side of the annotation).

Additionally, for each annotation, we performed exact matching with the gold standard corpus to find if the change was correct or not. Figure 4.6 presents the results of comparing Totum with the other harmonization solutions, considering the merged test corpus. Overall, there is a high level of agreement between the several solutions, with an average of 85%



(a) **Intersection VS Totum**

(b) **Union VS Totum**

(c) **TotumID VS Totum**

Figure 4.6: Comparison of the annotations provided by Totum against the other harmonization solutions.

correct annotations. The biggest sources of disagreement are *new*, *remove*, *add right* and *add left* categories. The addition of new annotations is one of the most important, since it adds annotations that were not considered by other approaches. On average, 61% of these annotations are correct according to the gold standard. Considering nested alignment, more than 72% of those new annotations are correct. The impact of this task is reflected in the comparison with the intersection approach (Figure 4.6a). Ultimately, this task adds more true positives than false positives which contributes to a better precision, and reduces the number of false negatives contributing to a better recall. Another important category is *remove*, which discards false positives provided by other solutions. We can see the impact of this task in the comparison with union (Figure 4.6b), where more than 76% of the deletions are correct. Adding tokens to the right side is the category where Totum performs worst. In average, it changes 40% of the annotations to correct, 40% to incorrect, and 20% were wrong and remain wrong after the change. Finally, adding tokens to the left side presents a small positive contribution, by changing in average almost 50% to correct, 33% to incorrect, and 17% that are still wrong after the change.

The only difference between our two solutions is the compatibility with normalization systems. Thus, there is a high level of agreement between the two approaches, differing only

on the generation of new annotations (Figure 4.6c). Remove left and right did not present any significant results, which reinforces the idea that our solutions provide longer names in comparison with other approaches.

Due to the generation of longer names, Totum considers that the suffixes and prefixes "gene", "protein", and the ones relative to species and greek letters, always make part of the annotations. However, this is not consistent with the annotations on all corpora. For instance, in comparison with intersection, Totum corrects "IL-2" to "IL-2 gene", but changing "RFX-B" to "RFX-B protein" makes the annotation to be wrong according to the gold standard. Regarding the addition of greek letters, it corrects "SDF1" to "SDF1 alpha". Our solution also adds organism names on annotations, converting "CD81" to "mouse CD81" and "AML1" to "human AML1", which are not correct according to the manually annotated data. Furthermore, Totum may consider the same chunk of text as being an annotation or not, which could be correct or not depending on the corpus. For instance, in comparison with intersection, Totum removes the annotation "CD4", which is correct 51 times and wrong 22 times. The same occurs with the addition of the annotation "cytokine", which is correct 88 times and wrong 67 times. This behavior does not mean that Totum is completely wrong, since some corpora were annotated focusing in very specialized biomedical sub-domains, and consequently, some gene/protein names were discarded since they were not related with that sub-domain.

In summary, we can argue that Totum maintains a constant global behavior, allowing the annotation of large amounts of data following the same guidelines induced by training a machine learning model on several gold standard corpora.

4.4 Summary

In this chapter we presented Totum (`http://bioinformatics.ua.pt/totum`), a new cross-corpus solution to harmonize heterogeneous gene/protein names from several NER or normalization systems. This approach uses CRFs to take advantage of the variability existent in several corpora from different domains, learning the correct tags for the tokens and making the final result more precise and reasoned. In comparison with traditional harmonization solutions, which only allow fixing the annotations boundaries (by adding or removing tokens), our solution also allows creating new annotations or removing incorrect ones, which extends the traditional harmonization behavior. Totum is also compatible with normalization systems (TotumID), preserving the provided identifiers and avoiding the creation of new annotations which would not have an identifier assigned.

Analyzing the annotations provided by Totum, we concluded that improved results are achieved due to the deletion of incorrect annotations, the recognition of annotations discarded by other approaches, and the usage of the knowledge provided by the systems' annotations to

create new entity names. In the end, we may conclude that Totum provides longer annotations than the other approaches, presenting a similar behavior regarding the boundaries definition of the different gene/protein names.

The experiments demonstrate that both solutions outperform the most common and state-of-the-art approaches. Considering the merged corpus, and in comparison with an intersection approach, Totum presents F-measure improvements of up to 10.34%, 10.91% and 22.25% on exact, approximate and nested alignment, respectively. Comparing against union, improvements of 7.61%, 7.06% and 16.17% are achieved, regarding the same matching strategies.

Overall, Totum takes advantage of the annotations provided by several systems for different corpora, providing a solution that is not constrained to a specific corpus as the original systems are. In the end, the harmonized annotations provided by Totum present F-measures of 69.30%, 77.34% and 81.77% for exact, approximate and nested alignment. With these results, we believe that this approach is a step towards a homogeneous annotation of MEDLINE abstracts, supporting several biomedical domains and organisms.

Chapter 5

Neji: heterogeneous biomedical concept recognition

This chapter is based on:

- D. Campos, S. Matos, and J. L. Oliveira, "A modular framework for biomedical concept recognition." *BMC bioinformatics*, vol. 14, no. 1, p. 281, Sep. 2013
- T. Nunes, D. Campos, S. Matos, and J. L. Oliveira, "BeCAS: biomedical concept recognition services and visualization." *Bioinformatics (Oxford, England)*, vol. 29, no. 15, pp. 1915–1916, Jun. 2013

In an effort to deal with the complex challenges of NER and normalization, several systems have been developed for the biomedical domain, using different approaches and techniques that can generally be categorized as being based on rules, dictionaries or machine learning (Section 2.2). Each approach has different resource requirements and deals differently with the linguistic variability that resulted from the lack of naming standards and the introduction of idiosyncratic names by the scientific community [26]. In general, ML-based solutions are better adapted to deal with strong variability and highly dynamic vocabularies, such as in gene and protein names. However, this approach does not directly provide identifiers for the recognized names. Thus, normalization must be performed in an extra step in order to relate each name with concept identifiers from curated databases or ontologies. In this case, a concept corresponds to a biological entity present on curated and specialized resources used to represent and map current knowledge. On the other hand, dictionary-based approaches are appropriate to deal with precisely defined vocabularies of names (e.g., diseases and species). This approach requires the construction of a unique resource containing most of the identifiers and names of a specific semantic type. However, this presents various challenges, since the

necessary information is usually spread over dozens of data sources and unique identifiers are specified on a per-resource basis, which hinders mapping identifiers between heterogeneous databases. Moreover, the same name may refer to different concepts, depending on the context in which it occurs. For instance, "NF1" can refer to a disease ("Neurofibromatosis Type 1") or to a protein ("Neurofibromin 1"). Accordingly, the development of NER and normalization solutions requires the application of multiple techniques, which can be conceptualized as a simple processing pipeline [26]:

- Input: interpret and filter input data to be processed;
- Pre-processing: process the input data in order to simplify the recognition process;
- Recognition: identify entity mentions from pre-processed data;
- Post-processing: refine generated annotations, solving problems of the recognition process or extending recognized names;
- Output: generate a structured output with the final annotations.

Each step of the processing pipeline may involve the implementation of various methods to fulfill the associated requirements. Due to the specificities of the biomedical domain, methods developed for common English may not provide the best outcomes when used on scientific documents. For instance, He and Kayaalp [51] analyzed the application of various tokenizers, concluding that most solutions are too simplistic for real-life biomedical applications. Thus, it is important to develop and use methods optimized to deal with the special linguistic characteristics of biomedical terms.

5.1 Background

Based on the general processing pipeline and considering the requirements of the biomedical domain, various solutions were implemented and used to support and streamline the development of complex biomedical IE solutions. Figure 5.1 presents the spectrum of frameworks and tools considering their relative specificity for this domain. The edges of the spectrum represent two contrasting types of solutions:

- General frameworks (left edge), which support the development of IE solutions with a pre-defined and general processing pipeline;
- Specialized tools (right edge), centered on the recognition of specific biomedical entity types and providing end-user features.

UIMA [296] and GATE [297] are examples of frameworks that provide a general solution to support the development of complex IE systems, being independent of the target domain. Such goal is achieved by providing a flexible processing pipeline based on a modular infrastructure, enabling problem decomposition and consequent re-utilization of modules. Besides the flexibility and re-usage advantages, such solutions also provide a strong infrastructure,

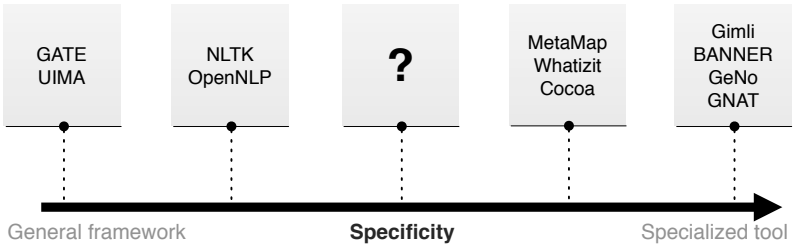


Figure 5.1: Spectrum of existing solutions for biomedical concept recognition according to their specificity.

such as cluster processing support for large amounts of data. However, due to the high level of abstraction, the development of new solutions may not be as straightforward as expected, requiring some time to correctly understand and have full control over the frameworks' features. Moreover, neither framework provides default modules optimized for the biomedical domain, which are provided by third parties, such as U-Compare [298] and JCoRe [299] for UIMA. Nevertheless, most of those modules are only available through web-services, which is an optimal solution for small experiments but not compatible with large scale and batch processing applications. Additionally, users must be careful when using modules from different providers in a single pipeline, since the application of different techniques (e.g., tokenization and sentence splitting) among different modules may considerably degrade performance results.

Toolkits such as NLTK [300] and OpenNLP, on the other hand, are not focused on providing a text processing pipeline, offering instead a multitude of implemented methods that developers can use and combine to build their own pipelines. Various features of OpenNLP are also available as modules for UIMA, which may simplify the creation of such pipelines. However, these solutions do not provide modules optimized for the biomedical domain. Instead, they allow training new modules focused on different goals and domains.

On the opposite edge of the spectrum are specialized NER and normalization tools, whose development was greatly promoted through the organization of challenges such as BioCreative [38, 90, 95] and JNLPBA [91]. Dozens of new solutions emerged using the resources provided by these challenges, which allowed a fair and fast comparison of divergent techniques. Gimli [28] and BANNER [165] are examples of NER solutions, and GeNo [301] and GNAT [181] are examples of NER and normalization tools. However, the resources provided by those challenges are too specific and focused on the recognition of particular entity types (e.g., gene and protein), generating highly optimized solutions that provide high performance results on tested corpora. NER solutions are typically open-source and publicly available as runnable applications, enabling re-usage of already implemented modules and fast development of new recognition systems. However, there is no explicit processing pipeline and such solutions are not flexible, limiting the addition or removal of processing modules. On the other hand,

normalization solutions are mostly not open-source, providing only web-services for remote usage, which is limited for batch processing.

There are also solutions focused on providing annotation of heterogeneous biomedical concepts. For instance, Whatizit [126], Cocoa[1] and NCBO Annotator [302] provide annotations of species, genes and proteins, and disorders, among others concepts. However, since they are provided as web-services, batch processing is limited and desirable functionalities, such as the possibility to configure annotation characteristics or to extend the provided features, are not available. MetaMap [303] is another tool that provides annotation of heterogeneous concepts, using the UMLS Methathesaurus and a set of rules for extracting text chunks and scoring them as candidates for concept names. However, MetaMap does not use dictionary matching or machine learning solutions, which have already proven to provide significantly better results than just rule-based approaches. Moreover, since it is provided as an end-user tool, it is also limited in terms of configurability and extensibility.

Considering the current frameworks and tools for the biomedical domain, we believe there is a lack of solutions that combine the advantages of the two edges of the spectrum: modularity, speed, usability and domain optimization. This chapter presents Neji, an open source framework for biomedical concept recognition that provides an automated and flexible processing pipeline that includes built-in methods optimized for the target domain. It supports the application of both machine learning and dictionary-based approaches, automatically combining generated annotations and supporting concept ambiguity. Neji also supports known input and output formats, and easy development of new pipelines and modules.

In the next section, we give a detailed description of Neji's modular architecture, presenting the core infrastructure, the included modules and its usability. Afterwards, Neji is evaluated in terms of concept annotation accuracy and speed. In the end, we discuss the main advantages and applications of Neji.

5.2 Methods

The design and implementation of Neji was focused on four crucial characteristics: modularity, scalability, speed and usability. In order to achieve modularity, every processing task is performed by an independent module, which can be executed ad-hoc or integrated in a processing pipeline. Nonetheless, each module has its own input and output specifications. Regarding scalability, the solution should be able to support simultaneous application of dozens of dictionaries and machine-learning models for concept recognition, while at the same time processing large data sets (i.e., millions of abstracts). One of the key features to deal with large data sets and considerably improve processing times is concurrent processing, allowing different CPU cores to process several documents at the same time. Additionally, it

[1] `http://npjoint.com`

is also fundamental to take processing speed into consideration when choosing libraries and techniques to perform the different steps. Finally, developers and researchers should be able to easily use pre-defined pipelines, implement custom pipelines with provided modules and/or implement new modules respecting previously specified interfaces. Moreover, typical processing modules, such as sentence splitting and tokenization, should be part of the framework and available for direct use and/or extension.

A framework with such characteristics should be an added value for the biomedical community, allowing any user to easily develop custom and complex solutions and use them according to their specific goals. Additionally, advanced users do not need to deal with various independent tools and libraries, allowing them more time to dedicate to their real goals.

5.2.1 Infrastructure

The core component of Neji is the pipeline, which allows users to submit various modules for execution following a FIFO (First In, First Out) strategy. Thus, a pipeline is a list of modules that are executed sequentially, considering specific goals and target chunks of text. Figure 5.2 illustrates the idea of this modular and flexible architecture. Each module is implemented as a custom Deterministic Finite Automaton (DFA), with specific matching rules and actions. We used the hierarchical text processing features of Monq.jfa[2] to support the pipeline infrastructure and module execution (Figure 5.3). When a pipeline is executed, the input documents are the input of the first module, and the output of the first module is the input of the second module and so on, until the last module provides the output to a storage resource specified by the user. Since different tasks have different requirements, different types of modules are defined:

- Tagger: processes the input data and reflects the changes in the same data. For instance, when performing sentence splitting, inline annotations can be provided to reveal the obtained sentences;
- Loader: loads information present on the input data into memory. For instance, if inline biomedical name annotations are present in the input text, a loader can be used to load such annotations into memory;
- Hybrid: processes input data and stores the results in internal memory. Inline annotations can also be provided as output. For instance, when performing sentence splitting, it should be useful to provide inline annotations of the sentences and load them into memory. Obviously, a tagger and a loader can be used instead, but some processing time would be wasted in reading the annotations from the tagger to the loader;
- Reader: a Tagger that is used to collect data of interest from the input resource;
- Writer: a Tagger that is used to generate output data to a specific resource.

[2] `http://monqjfa.berlios.de`

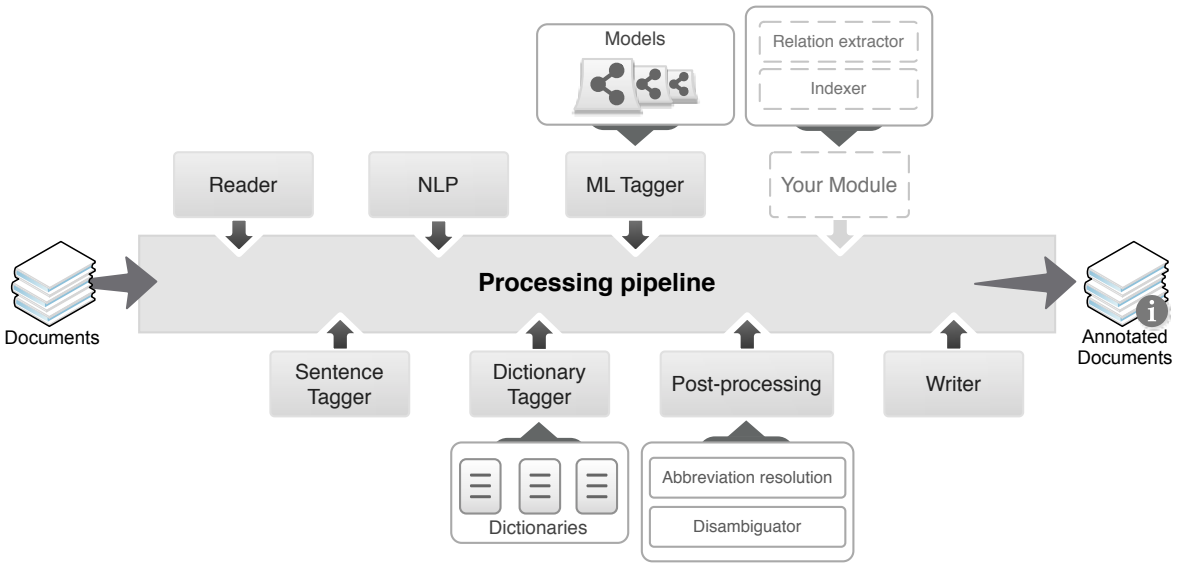


Figure 5.2: Illustration of the processing pipeline and modular architecture of Neji.

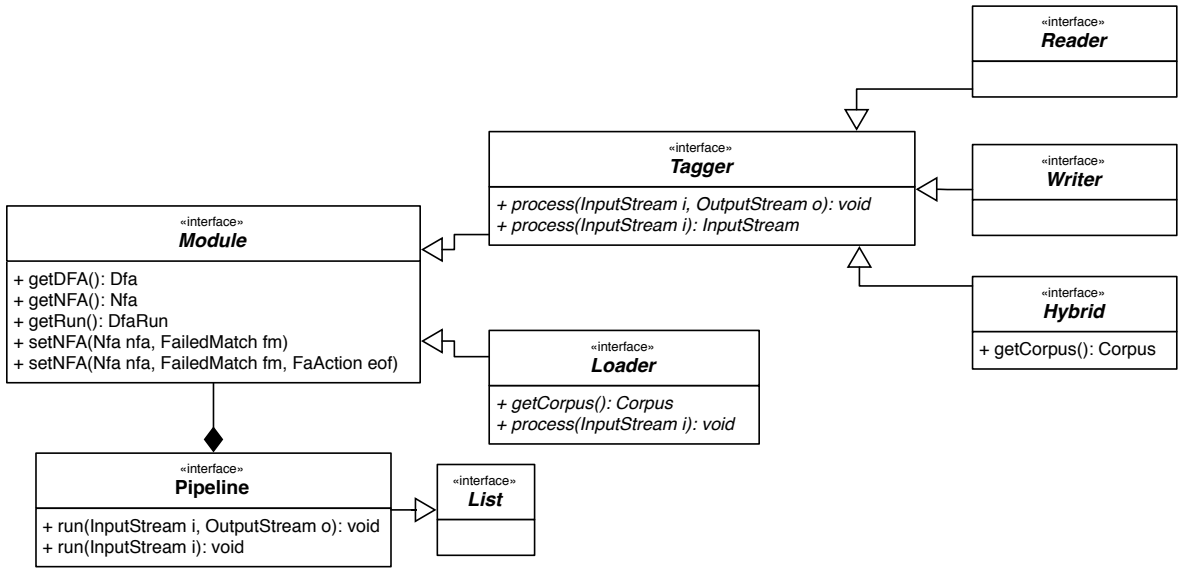


Figure 5.3: Interface diagram to model implementation of pipelines and respective modules.

In order to support default and basic behaviors, Neji already provides implementations of the various components, namely tagger, loader, reader, writer, hybrid and pipeline. Such architecture allows developers to easily build custom module types or pipelines.

Since Neji is a framework focused on biomedical concept recognition, it also defines and provides a flexible and complete data structure to represent a corpus. Thus, developers do not need to specify their own internal data structures, and they can easily extend the provided data representation. Figure 5.4 illustrates the final internal data representation of a corpus with sentences and respective annotations. Moreover, since Neji supports automatic annota-

tion of heterogeneous biomedical concepts, in which the existence of nested and/or intersected annotations is common, it is important to integrate a data structure that suits such characteristics in the best and most automated way as possible. A tree of annotations is the data structure that better fulfills such requirements, presenting various advantages over typical approaches (e.g., list of annotations): a) structured annotations provide enhanced information, since nested and intersected annotations and their respective identifiers are provided; b) the levels of the tree are directly associated with the detail of annotations, the deeper the level the more deeply an annotation is nested and/or intersected in others; c) the consistency of the tree and of the respective annotations can be maintained through automatic algorithms; d) ambiguity problems are clear; and e) filtering annotations can be as simple as pruning the tree. As illustrated in Figure 5.5, each sentence includes a tree of annotations. In order to facilitate the use and management of these trees, as well as for maintaining the consistency of the annotations, the following methods are provided:

- Sorted insert: when an annotation is added to the tree, it is automatically put in place, maintaining the tree consistency;
- Sorted delete: when an annotation is removed from the tree, all other annotations are put in place in order to keep tree consistency;
- Traversal: obtain a list of ordered annotations following typical tree traversal techniques: by level, pre and post-order;

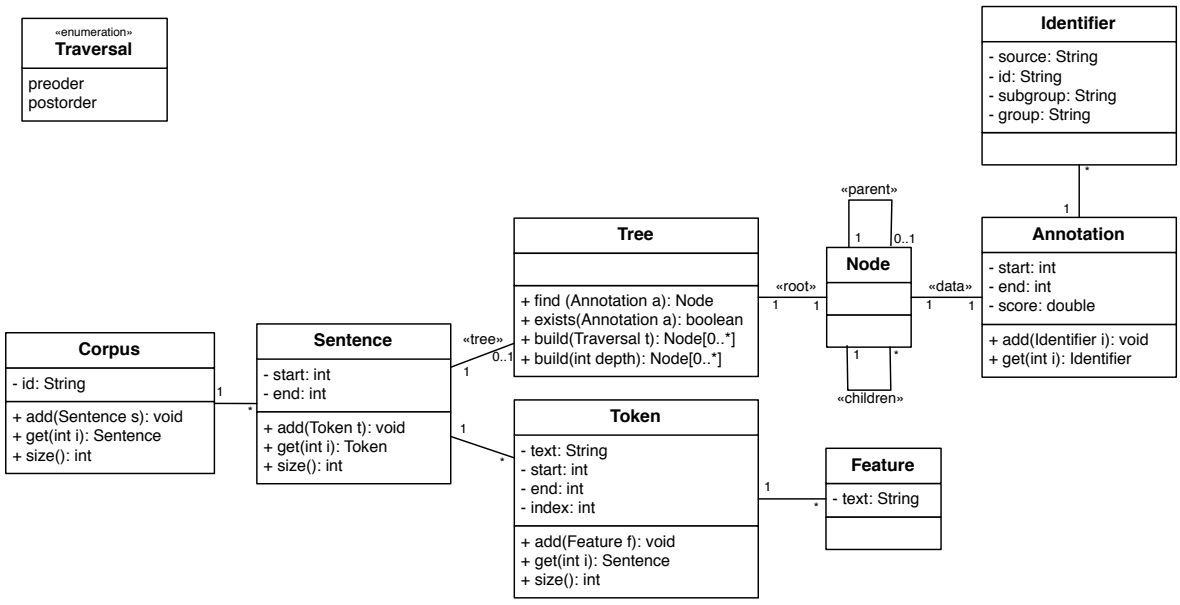


Figure 5.4: Overview of the internal data structure to support processed data.

Since an annotation without concept identifiers is less informative, it is important to provide an infrastructure that allows each annotation to contain various identifiers. Moreover,

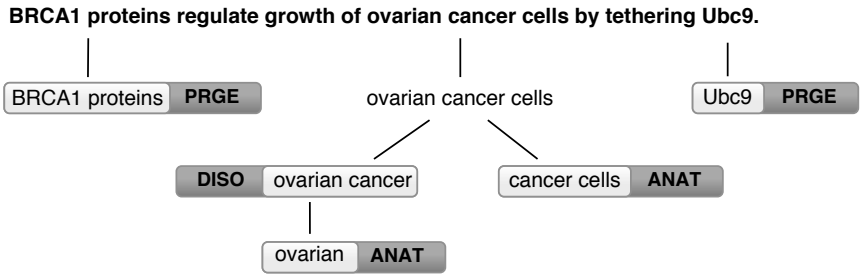


Figure 5.5: Illustration of implemented concept tree. Such structure automatically supports nested and intersected concepts, clearly exposing ambiguity problems (PRGE: Proteins and genes; DISO: Disorders; and ANAT: Anatomy).

each identifier should provide complete information regarding its original source and concept type. Thus, the following quadruple composes each identifier: source (original resource that contains the name and respective identifier); identifier (unique identifier of the concept in the previously specified resource); group (semantic group of the concept); and sub-group (semantic type of the concept).

5.2.2 Modules

With the proposed infrastructure, the conditions to build the required modules for text processing and concept recognition are now met. The modules presentation follows the processing pipeline previously presented and illustrated in Figure 5.2.

Readers

A reader module is used to interpret input data, in order to collect the relevant data and convert it into a format that is readable by the following modules. Instead of obtaining the relevant data and storing it into memory, we decided to use a tagger to mark the original input text with Regions of Interest (ROIs) tags ("<roi>text</roi>"). Thus, the following modules only have to match the ROI tags and process the contained text. Two different reader modules are already provided, allowing to process XML and raw text. The XML module allows developers to specify the tags of interest. For instance, considering the Pubmed XML format, if only titles and abstracts have to be processed, only the content of the tags "ArticleTitle" and "AbstractText" are of interest. On the other hand, the raw reader considers that all the input text is of interest to be processed.

Natural Language Processing

After obtaining the texts of interest, the next fundamental step is to perform sentence splitting, since a sentence is the basic unit of logical thought. This phase presents various complex challenges due to the specific characteristics of scientific biomedical texts [304, 305].

Thus, we integrated a module to perform sentence splitting taking advantage of the Lingpipe library, through a sentence splitting model trained on biomedical corpora that presents high-performance results [52]. NLP tasks are performed using GDep [58], a dependency parser for the biomedical domain built on top of the GENIA tagger, which performs tokenization, lemmatization, POS tagging, chunking and NER. Since we are not interested in the named entities provided by the GENIA tagger, we removed such module and its dependencies. Moreover, we decided to make the tokenizer behavior more consistent, by breaking words containing the symbols "/", "-" or "." into multiple tokens, which showed to provided improved results [28]. Because GDep combines all the tasks in order to perform dependency parsing, we decoupled the various processing tasks, obviously respecting all task dependencies and resources (tokenization < POS < lemmatization < chunking < dependency parsing). Thus, for each task, only the required resources (e.g., models) are loaded. For instance, if one needs the pipeline just for dictionary matching, only the tokenization plugin will be loaded and executed. On the other hand, when dependency parsing is required, all the processing tasks are performed and respective information provided. For instance, if a machine-learning model uses tokens, POS and lemmas as features, but not chunks or parsing features, these two tasks are not performed, making the process considerably faster.

Concept recognition

As stated before, distinct biomedical concepts require distinct approaches in order to achieve more accurate recognition. Thus, Neji provides concept recognition using both dictionary and machine learning-based approaches. Dictionary matching is offered using a modified version [97] of the dk.brics.automaton[3] library, which provides efficient regular expression matching with DFAs. Considering that each input string of symbols is a name from the dictionary, one can build a DFA to match all names in a dictionary. Additionally, each regular expression representing a name from the dictionary is associated with a specific identifier, enabling concept recognition. Such approach supports both exact and approximate matching, and performs the recognition of named entities in $O(n)$ time, where n is the size of the document. Since a large amount of false positives may be generated using approximate matching, and considering that we are dealing with a general biomedical solution, we decided to use case insensitive exact matching. Orthographic variants of names can be generated and provided in the dictionary. Even so, it is necessary to pay special attention to terms that are common English words. Thus, a list of non-informative words for the biomedical domain [57] is ignored during the matching process. Similarly, biomedical names with two characters or fewer are also discarded. However, such a strategy may mean that acronyms of known entity mentions would be missed, which can be overcome by a post-processing module for acronym resolution.

[3] `http://www.brics.dk/automaton`

Dictionaries are provided in Tab-separated Values (TSV) format with two fields: identifier and list of names. Identifiers should follow the format "source:identifier:type:group" and their respective names must be concatenated with a pipe ("|"). To allow easy configuration and support dozens of dictionaries, files must be provided in a folder with an additional priority file, which contains the file names of the dictionaries (one per line) and defines the priority to be used if a disambiguation method is applied. This simple strategy enables fast, easy and flexible configuration of dictionaries.

In order to optimize the concept recognition results, some directives are followed when applying dictionary matching, assuming that a different dictionary file is used for defining concepts in each semantic group or type:

- Considering one dictionary (i.e. same semantic group/type), only the entry with the largest span is matched;
- If two entries with the same text exist, in the same or in different dictionaries, both entries are matched and both identifiers are provided;

The support of machine learning-based solutions is provided through Gimli, which uses the CRFs implementation from MALLET to recognize various biomedical entity types, and provides high-performance results in two well-known corpora: GENETAG [92] and JNLPBA [91]. It also provides a complete set of basic and complex features, serving as a good starting point to develop NER solutions for the biomedical domain. Thus, various CRF models trained on Gimli can also be used in Neji, each one focused on a different biomedical concept type. Gimli already provides models for the recognition of gene and protein names, trained on GENETAG, and for the recognition of gene and protein, DNA, RNA, cell type and cell line names, trained on JNLPBA. Nonetheless, developers can also use Gimli to easily train new models on different corpora and/or focused on different entity types. However, Gimli only performs NER, not establishing a relation between chunks of text and unique identifiers from curated databases. Thus, we developed a simple and general normalization algorithm based on prioritized dictionaries. Following this algorithm, if an identifier is found in the first dictionary, the match is complete and the algorithm finishes. If no match is found in the first dictionary, the second one is used to find a match, and so on. In the end, if no matches are found in the provided dictionaries, the developers can choose to keep or discard the annotation. This configuration works well if the first dictionary contains a list of preferred names, and the remaining contain synonyms for each identifier. Using this setting, a mention to "TRAF2" would be matched in the first dictionary, since this is the preferred symbol for the gene associated with the protein with Uniprot accession Q12933, and the matching process would stop. Additionally, "TRAF2" is also a synonym for the gene "TANK" (Uniprot accession Q92844), but since this is defined in a dictionary with lower priority the match would not occur. Moreover, this strategy also provides flexibility to users, which only have to generate the various orthographic variants and prioritize them in the

dictionaries. Regarding the matching approach, if a partial match of the annotation is found in the dictionary, it is accepted as a valid identifier for the complete chunk of text. For instance, if only "BRCA1" is present in the dictionary, and the annotation "BRCA1 gene" is provided, the identifier of "BRCA1" is associated with the annotation. Conversely, if "BRCA1 gene" is in the dictionary and "BRCA1" is found in the text, a match is not obtained since "extra" tokens are only considered in the textual mention and not in the dictionary entries. ML models are provided to Neji following a similar approach of dictionaries, where a properties file defines the characteristics of each model.

Post-processing

Neji is also able to integrate post-processing modules, in order to optimize previously generated information. By default, an abbreviation resolution module is provided, in order to extend existing concepts. Thus, we adapted a simple but effective abbreviation definition recognizer [112], which is based on a set of pattern-matching rules to identify abbreviations and their full forms. In this way, we are able to extract both short and long forms of each abbreviation in text. If one of the forms is already provided as a concept, the other one is added as a new concept with the identifiers of the existing one. Additionally, any further occurrences of that entity are also automatically annotated. Depending on user requirements, it may be useful to filter concept annotations following pre-defined rules. Thus, Neji provides the ability to remove annotations from the concept tree based on three simple disambiguation strategies:

- By depth: remove annotations from the concept tree that are deeper than a specified depth;
- Nested same group: remove concept annotations that are nested on annotations of the same semantic group and with a larger span;
- By priority: remove nested and intersected concept annotations following a prioritized list.

Writers

Writers are used to store the recognized concepts in external resources, such as files and databases. If the user does not want to provide the result into an external resource, the corpus is programmatically available. Neji supports various well-known inline and stand-off formats used in the biomedical domain, such as IeXML [290], A1[4], CoNLL [295] and JavaScript Object Notation (JSON) [306]. Overall, identifiers are provided following the format "source:identifier:type:group", and using a pipe ("|") to concatenate various identifiers for a single annotation. IeXML is an inline annotation format based on XML tags, supporting

[4] `http://brat.nlplab.org/standoff.html`

two levels of detail, i.e. only one annotation nested or intersected in another. Moreover, various identifiers can be provided using IeXML. Both CoNLL and A1 support ambiguous and intersected concept annotations. However, complex identifiers are not supported in CoNLL, thus only the semantic group is provided. The output of the A1 format can be used with brat ,[307], in order to visualize and edit the generated annotations. Finally, the JSON format provides all the information contained in the tree, together with the sentence and respective character positions. We also provide our own format, in order to overcome some limitations of other formats regarding nested/intersected annotations and multiple identifiers. It can be seen as an alternative to JSON, being more readable and understandable by humans.

Figure 5.6 presents an example of the Neji output generated for a sentence. As we can see, each sentence has its own identifier, start and end character positions, and respective text. Regarding annotations, an indentation-based approach is used to reflect the tree hierarchy, accompanied with the respective term identifier, start and end character positions, and associated text and identifiers.

```
S77     9820 9986    In Fanconi anemia, death, bone marrow transplant …
        T1      9823 9836       Fanconi anemia  UMLS:C0015625:T047:DISO
        T2      9846 9856       bone marrow     UMLS:C0005953:T024:ANAT
                T2-1    9846 9849       bone    UMLS:C0262950:T023:ANAT
                T2-2    9851 9856       marrow  UMLS:C0376152:T023:ANAT
```

Figure 5.6: Example of the Neji output format.

5.2.3 Parallel processing

In order to simplify the use of the various modules and required resources, we developed a method to manage these resources, which we call Context. It automatically loads the resources that are required to run a specific pipeline. Thus, researchers do not need to deal with repetitive and time consuming tasks such as loading dictionaries, ML models, parsers and sentence splitters. Additionally, we also provide parallel processing of documents through multi-threading support. To accomplish this, the libraries and respective dependencies used were adapted to allow multi-threaded execution, solving some limitations with MALLET and GDep. The Context also supports multi-threading, by automatically generating the required duplicate resources when necessary. For instance, concurrent annotation of documents using one ML model is not possible, requiring one instance of the ML model for each thread. In order to apply parallel processing, each pipeline must be implemented in a Processor, which is a runnable pipeline with context and input and output resources specification. Base implementation of a Processor is already provided, which simplifies the development of alternative runnable pipelines. A Batch is also provided, which performs concurrent processing of input resources using a specific Processor and Context. Considering the typical use case scenario of parallel processing in the biomedical domain, i.e., process files in an input folder and provide

the results to an output folder, we developed a Batch executor to make the applicability of parallel processing easier. The Batch automatically generates the required Processor threads to process specific files in a folder. Custom arguments for the processors can be also provided, which takes advantage of Java reflection.

5.2.4 Usage

In order to make the annotation process as simple as possible in typical use cases, Neji integrates a simple but powerful CLI tool, which is flexible and provides a complete set of features:

- Annotate using dictionaries and/or ML models with respective normalization dictionaries;
- Various input and output formats. When the XML input format is used, the XML tags should be indicated;
- Parsing level customization. By default, Neji automatically finds the appropriate parsing level considering the ML model characteristics;
- Number of threads customization;
- Wildcard input filter to properly indicate the files to process;
- Support for compressed and uncompressed files.

The features provided by the CLI tool allow annotating a corpus using a simple bash command, such as:

```
./neji.sh -i input/ -if XML -o output/ -of XML
-x AbstractText,ArticleTitle -d resources/dictionaries/
-m resources/models/ -c -t 6
```

In this example, Neji uses six threads to annotate the compressed XML documents in the input folder with the specified dictionaries and machine-learning models, providing the resulting XML documents to the output folder. Note that only the text inside the specified tags is annotated. If users do not want to use the provided CLI, it is also straightforward to develop a processor and process the documents using the batch helper. First, a processor taking advantage of the pipeline features must be implemented.

Figure 5.7:a presents the construction of a complete pipeline processor that produces the same results as the previous bash command, considering a specific context, input and output documents provided in the constructor. Afterwards, this pipeline processor must be used to perform batch processing of documents. Figure 5.7:b shows how a context is created considering input models and dictionaries folders, and how a batch is created for specific input and output folders. Finally, the batch is executed considering the provided context and all documents are annotated. Complete and detailed documentation on how to use the CLI tool, build custom processors, and build processing modules is provided in the Neji's web page.

```
public class XMLProcessor extends BaseProcessor {
        …
        @Override
        public void run() {
                ContextProcessors cp = context.take(); //Take parser, sentence splitter and CRF
                Corpus corpus = getInputCorpus().getCorpus(); //Get corpus to store processed data
                Pipeline p = new DefaultPipeline(); //Create pipeline
                p.add(new XMLReader(new String[]{"ArticleTitle", "AbstractText"})); //Reader
                p.add(new SentenceTagger(cp.getSentenceSplitter())); //Sentence tagger
                p.add(new NLP(corpus, cp.getParser())); //NLP
                for (Dictionary d : context.getDictionaries()) { //Dictionary matching
                        p.add(new DictionaryHybrid(d, corpus));
                }
                for (int i = 0; i < context.getModels().size(); i++) { //Machine learning
                        p.add(new MLHybrid(corpus, context.getModels().get(i), cp.getCRF(i)));
                }
                p.add(new IeXMLWriter(corpus)); //Writer
                p.run(getInputCorpus().getInStream(), getOutputCorpus().getOutStream()); //Run pipeline
                context.put(cp); //Put parser, sentence splitter and CRF back
        }
}
```

***a)* Pipeline processor.**

```
Context context = new Context(modelsFolder, dictionariesFolder); //Create context
boolean areFilesCompressed = true;
int numThreads = 6;
Batch batch = new FileBatchExecutor(inputFolder, InputFormat.XML, outputFolder, OutputFormat.XML,
                                    areFilesCompressed, numThreads); //Create batch
Class c = XMLProcessor.class; //Get Processor class
batch.run(c, context); //Run batch processing
```

***b)* Batch executor with context and pipeline processor.**

Figure 5.7: Java code snippets to create a runnable processing pipeline and use it in a batch executor with context.

5.3 Results

To provide general feedback regarding Neji's reliability as a framework, it is fundamental to evaluate its behaviour on real life problems. Thus, we believe that such framework should be evaluated considering two key characteristics:

- Concept annotation: what is the quality of the produced concept annotations?
- Speed: how long it takes to process a specific amount of documents?

Accordingly, we collected manually annotated corpora, dictionaries and ML models to take advantage of Neji, and compared the achieved performance results with existing solutions.

5.3.1 Corpora

Our primary analysis was centered on the CRAFT corpus [107], one of the largest publicly available gold standard corpora for the biomedical domain, focused on multiple biomedical

concept types with heterogeneous characteristics. The initial release contains a set of 67 full-text articles (more than 21 thousand sentences) manually annotated by domain experts, focused on nine biomedical ontologies and terminological resources: Chemical Entities of Biological Interest (ChEBI); Cell Ontology; Entrez Gene; Gene Ontology (biological process, cellular component, and molecular function); NCBI Taxonomy; Protein Ontology and Sequence Ontology. Overall, it contains almost 100 thousand annotations. However, CRAFT does not include anatomical and disorder concepts, which we believe are fundamental to cover the general biomedical concept spectrum. Thus, we decided to use two other corpora for concept annotation evaluation. The AnEM [105] corpus is focused on anatomical entities, using a fine-grained classification system based on the Common Anatomy Reference Ontology (CARO). The annotated concepts are precisely divided into eleven anatomical class labels, such as "Organ", "Tissue", "Cell" and "Organism substance". This corpus is based on 250 abstracts and 250 full-text extracts (article sections) randomly selected from PubMed and from PubMed Central (PMC), containing 3135 manually annotated concepts. For testing purposes, 100 abstracts and 100 full-text extracts are provided, summing together 1879 annotated concepts. Finally, the third was the NCBI disease corpus [308], produced by expert annotators using the Unified Medical Language System (UMLS) as reference resource and containing disease concepts classified into four class labels: Specific Disease, Disease Class, Composite Mention and Modifier. It contains 793 abstracts (6651 sentences) from PubMed with 6900 disease mentions. For testing purposes, 100 abstracts with 961 mentions are provided. In the end, we used the 67 full-text articles of the CRAFT corpus, and the test parts of both AnEM and NCBI corpora, in order to allow direct and fair comparison.

5.3.2 Resources

Considering the three corpora, we collected the ML models and/or dictionaries described below to recognize biomedical concepts of each type. Resources for the "Disorders" and "Anatomy" types were used for annotating the NCBI disease and AnEM corpus, respectively, and the remaining were considered for the CRAFT corpus:

- Genes and proteins: due to the variability of gene and protein names, their recognition was performed using a ML model trained on GENETAG. It applies a complete and complex set of features, namely lemmas, POS, chunking, orthographic, local context (windows) and morphological features. LexEBI [117], which contains a filtered version of BioThesaurus [116], the most complete resource of gene and protein names, is used to perform normalization. The dictionary was further filtered to only include gene and protein names for 21 of the most commonly studied species[5]. Two different dictionaries

[5]A. thaliana, B. taurus, C. elegans, C. reinhardtii, D. rerio, D. discoideum, A. mellifera, C. albicans, D. melanogaster, H. sapiens, M. musculus, R. norvegicus, S. cerevisiae, Hepatitis C virus, M. pneumoniae, P. falciparum, P. carinii, S. pombe, Z. mays, E. coli and X. laevis.

were generated: the first with preferred names and the second with synonyms for each identifier. Additionally, for each dictionary a set of orthographic and semantic variants was generated using the Lexical Variants Generation (LVG) tool [118], namely: *a)* add derivational, uninflected and inflectional name variants; *b)* strip ambiguous words, punctuation symbols and plural suffixes; *c)* add known synonyms and variants from biomedical databases; and *d)* invert names around commas. In the end, four dictionaries were used with the following matching priority: *1)* preferred terms; *2)* synonyms; *3)* preferred terms with variants; and *4)* synonyms with variants. A simple filtering of gene and protein identifiers was also applied as a post-processing step, by discarding identifiers associated with species that were not found in the document. Thus, if identifiers for human and mouse proteins were provided for a recognized protein name and mice were not referred in the document, the identifier for the mouse protein was removed from the protein annotation;

- Chemicals: a dictionary of chemical names was built using the ChEBI database of molecular entities [83];
- Species: the dictionary provided by LINNAEUS [97] was extended by adding the entries from the NCBI Taxonomy assigned to taxonomical ranks above "species", that is, from "genus" to "domain". For each entry, we included the names from NCBI as well as the synonyms obtained from the corresponding concept in the Unified Medical Language System (UMLS) Metathesaurus [118]. Furthermore, less specific names for species that also appeared as names in higher taxonomical levels, such as the genera "rat" or "mouse", were filtered and kept only at the highest level, in order to approximate the annotation guidelines used in the CRAFT corpus;
- Cells: cell names were compiled from the "Cell" and "Cell Component" semantic types in the UMLS Metathesaurus;
- Cellular Component, Biological Process and Molecular Function: terms for these concept types were obtained from the corresponding sub-ontologies of the Gene Ontology (GO) [47], and expanded with synonyms from the corresponding concepts in the UMLS Metathesaurus. Additionally, UMLS concepts assigned to the UMLS semantic types "Physiologic Function", "Organism Function", "Organ or Tissue Function", "Cell Function", "Molecular Function" and "Genetic Function" were also included since they identify concepts closely related to biological processes and molecular functions, even if they are not directly mapped to GO terms;
- Disorders: names and synonyms for abnormalities, dysfunctions, symptoms and diseases were extracted from the Metathesaurus. We considered the following UMLS semantic types assigned to the "Disorders" semantic group: "Acquired Abnormality", "Anatomical Abnormality", "Congenital Abnormality", "Disease or Syndrome", "Mental or Behavioral Dysfunction", "Neoplastic Process", "Pathologic Function" and "Sign

or Symptom";

- Anatomy: anatomical entities were extracted from the Metathesaurus, considering the following semantic types grouped under the "Anatomy" semantic group: "Anatomical Structure", "Body Location or Region", "Body Part", "Organ, or Organ Component", "Body Space or Junction", "Body Substance", "Body System", "Cell", "Cell Component", "Embryonic Structure" and "Tissue". The semantic type "Fully Formed Anatomical Structure" was not included, as it contains only a few very general terms, such as "total body" or "whole body structures". The terms from the "Cellular Component" sub-ontology in GO were also included. Additionally, we included the terms from the "Neoplastic Process" semantic type since this most closely matches the "Pathological Formation" annotation type included in the AnEM corpus.

As a filtering step to eliminate inconsistent names and names that would generate a large number of false positives, we rejected names with one or two characters, names starting with a word from a strict list of stopwords (e.g. "*very* long chain fatty acid metabolic process", "*the* cell"), and also any single word name if that word was included in a broader list of stopwords generated from the list of most frequent words in MEDLINE. Some relevant terms that occur very frequently in MEDLINE, such as general names of diseases (e.g. "cancer", "diabetes"), Gene Ontology terms (e.g. "expression", "transcription") and species (e.g. "human", "Saccharomyces"), were not included in this stopword list, to allow identifying them in texts. As can be seen, different resources are used for each of the considered concepts in order to provide the best and most complete results as possible, an approach greatly simplified by Neji's modular pipeline. In the end, our dictionaries contain almost 1 million concept identifiers with 7 million name variants.

5.3.3 Concept annotation evaluation

Two different evaluation approaches were performed, in order to fully assess the quality of the provided concept names and identifiers:

- Named entities: evaluate the quality of the provided text mentions discarding the assigned identifiers;
- Normalization: evaluate the quality of the text mentions together with the assigned identifiers.

Regarding the evaluation of named entities, five matching techniques were considered:

- Exact: annotation is accepted if both left and right sides match with the gold standard annotation;
- Left: annotation is accepted if the left side matches;
- Right: annotation is accepted if the right side matches;

- Shared: annotation is accepted if the left or the right sides match;
- Overlap: annotation is accepted if there is any kind of match: exact, nested or intersected.

Such matching strategies allow a better understanding of annotation quality, since a non-exact matching does not mean that the correct concept was not recognized. For instance, considering gene and protein names, some systems and/or corpora include the organism name in the concept name and others do not, which remains a point of active discussion among expert annotators. Other point of disagreement is the inclusion of the tokens "protein" or "gene" as suffix or prefix, or including Greek letters in entity mentions [30]. Such analysis is also important since various post-NER tasks can be performed even if imprecise names are provided (e.g., relation and event mining).

The performance results on the various corpora were analyzed against previously published works to provide fair comparison. However, a complete comparison considering the five matching strategies is not always possible, since these different results are not stated in some works.

Regarding normalization and identifiers matching, we also considered two different matching strategies:

- Exact: annotation is accepted if one identifier is provided and it matches exactly with the gold standard;
- Contains: annotation is accepted if the provided list of identifiers contains the gold standard identifier.

Considering both matching strategies allows a more thorough analysis of the validity of the identifiers assigned to each entity mention. This evaluation was performed on the CRAFT corpus, since among the corpora considered in this work, only this one provides concept identifiers.

Common evaluation metrics were used to analyze and compare the achieved results: Precision (the ability of a system to present only relevant items); Recall (the ability of a system to present all relevant items); and F-measure (the harmonic mean of precision and recall). Note that the presented results are micro-averaged, meaning that a general matrix of TP, FP and FN values is built from all documents to obtain final precision, recall and f-measure scores.

CRAFT

Considering the databases and ontologies used in the annotation of CRAFT, we defined six concept classes: species, cell, cellular component, chemical, gene and protein, and biological processes and molecular functions. Biological processes and molecular functions are grouped into a single class, since annotations are provided in a single file using a single concept

type. Moreover, since gene and protein are provided through Entrez Gene (EZ) and Protein Ontology (PO), we decided to perform two different evaluations regarding the recognition of named entities: 1) against concepts provided by EZ; and 2) against concepts provided by EZ and/or PO. The performance on this NER task was compared against the results published by Verspoor et al. [52], who presented state-of-the-art results on CRAFT for sentence splitting, tokenization, POS tagging, syntactic parsing and named entity recognition. However, it only presents results for gene and protein recognition, where BANNER claims the best performing results. Thus, we decided to also use Cocoa and Whatizit to compare the achieved performance results for all the concept classes. Since Cocoa concept classes do not match directly to the ones provided in CRAFT, we had to group them together to better fulfill the requirements and to achieve better results:

- Species: "Organism" and "Organism1";
- Cell: "Cell";
- Cellular Component: "Cellular component", "Location" and "Complex";
- Chemical: "Chemical";
- Gene and Protein: "Protein", "Molecule" and "Category";
- Biological Process and Molecular Function: "Bio Process" and "Process".

Whatizit was used through the "whatizitUkPmcAll" pipeline, which is used in Europe PubMed Central [48] to provide species, chemical, gene and protein, cellular component, biological process, molecular function and disorder concept annotations. To match the output with CRAFT, biological process and molecular function annotations were grouped into a single concept class, and disorders annotations were discarded.

Figure 5.8 presents the named entity recognition results achieved by Neji, Whatizit, Cocoa and BANNER on the CRAFT corpus, considering the various matching strategies. As we can see, there are considerable variations between the various matching strategies. For instance, on gene and protein names recognition, Neji, Whatizit and Cocoa perform much better on left matching in comparison to right matching, which confirms the variability of annotation guidelines. Moreover, Neji and Cocoa also present better results on right matching on cell recognition, which indicates the presence of word prefixes on the gold standard that are being discarded by the automatic solutions. Those facts reflect the high variability of biomedical concept names, with different guidelines being followed by manual annotators leading to the generation of heterogeneous resources. Thus, as stated before, such discrepancies should be taken into account when evaluating solutions on corpora that follow different annotation guidelines.

Overall, Neji presents the best results, with significant improvements on various concept types, namely on concepts associated with GO (cellular component, biological process and molecular function), chemical and gene/protein. In more detail, we can see that Neji is the solution that presents overall best recall results without losing precision. Additionally, Neji

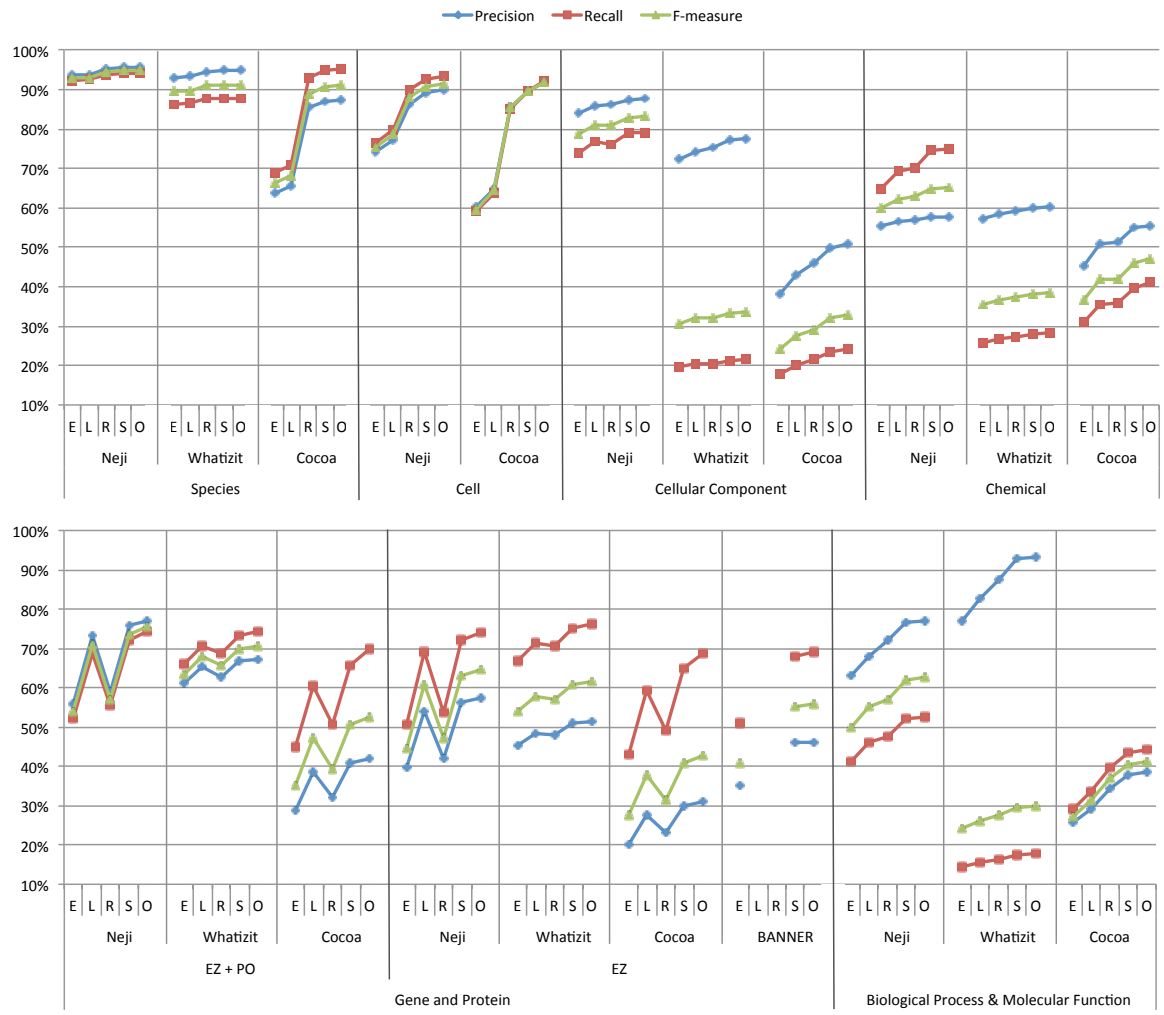


Figure 5.8: Evaluation results for named entity recognition, considering precision, recall, and F-measure achieved on CRAFT corpus, using exact (E), left (L), right (R), shared (S) and overlap (O) names matching. Evaluation considers species, cell, cellular component, gene and protein, chemical, biological processes and molecular functions concept names.

also presents a positive constant behavior, with an average variation of 9% of F-measure between exact and overlap matching. However, Whatizit is the most constant solution, with an average variation of 4% of F-measure. On the other hand, Cocoa has the highest variation, with 18% of F-measure.

Neji obtained state-of-the-art results on the recognition of species and cell concepts, with overlap F-measure results of 94.7% and 91.5%, respectively. Extending LINNAEUS dictionaries allowed an improvement of more than 8% of F-measure on overlap matching, from 86.1% to 94.7%. Nonetheless, both Cocoa and Whatizit present competitive results on species, and Cocoa also achieved state-of-the-art results on cell identification. Neji achieved an F-measure of 83.2% on overlap matching in the recognition of cellular component names, which is signifi-

cantly better than Cocoa and Whatizt. For instance, a detailed analysis showed that Cocoa's performance is considerably degraded by the presence of terms such as "cell" and "cellular". Regarding gene and protein recognition, Neji ML model presents better results than Cocoa, BANNER and Whatizit on left, shared and overlap matching. Its performance drop on exact and right matching appears to be a consequence of the different annotation guidelines in CRAFT and GENETAG, which was used to train Neji's ML model. Specifically, species names, and suffixes such as "gene" and "protein" are considered as part of the concept name in GENETAG but not in CRAFT, causing an erroneous evaluation when exact matching is taken into account. Considering only the concepts from Entrez Gene, Neji showed an improvement of more than 3% of F-measure on overlap matching against the second best, Whatizit. When compared against BANNER, an improvement of 8% was achieved. Regarding Entrez Gene and/or Protein Ontology concepts, Neji presents an improvement of more than 5% of F-measure against Whatizit and 23% against Cocoa, on overlap matching. Finally, the results achieved on chemical and biological processes and molecular functions are considerably better than Cocoa and Whatizit. However, we believe there is margin for progress, by: 1) collecting more name variants to improve the recall for biological processes and molecular functions; and 2) refining existing chemical dictionaries to improve precision.

Regarding normalization, previous works have presented performance evaluation results for specific entity types on specifically developed corpora, such as AIMed [207] and/or BioInfer [208] corpora for gene and proteins. Therefore, we evaluated the entity normalization performance achieved with Neji on the CRAFT corpus and compared it to the results obtained using the available pipelines in Whatizit, as this was the only freely available system that allowed recognition of various concept types with identifiers for each recognized concept name.

In order to collect the performance results, we first converted the identifiers provided by Neji and Whatizit to the ones used in the CRAFT corpus, since the resources used for each concept type were generally different. However, this mapping may deliver various problems, such as absent and ambiguous mapping, i.e., one identifier that is mapped to multiple identifiers, that will directly affect the obtained results. Table 5.1 presents a detailed analysis of the identifier mapping for Cell, Gene and Protein, and Biological Process and Molecular Function concept names, considering the annotations provided by Neji and Whatizit. Uniprot identifiers for genes and proteins were mapped to Entrez Gene (EZ) and Protein Ontology (PO) identifiers using the mapping provided by Uniprot to EZ and the mapping provided by PO to Uniprot. The UMLS concept identifiers assigned by Neji to Cell concept names were mapped to Cell Ontology (CO) identifiers through the mapping to the Foundational Model of Anatomy (FMA) ontology available in CO. However, this mapping is highly limited, since it only covers approximately 30% of CO. Finally, the dictionaries used in Neji for the recognition of Biological Process and Molecular Function concept names include some concepts from

various UMLS semantic types that are not mapped to GO entries, as used in the CRAFT corpus.

The analysis of Table 5.1 shows that only 53% of the identifiers provided by Whatizit could be mapped to Entrez Gene. Nonetheless, most of the recognized concept names (95%) were associated to at least one identifier that could be mapped to an Entrez Gene identifier. On the other hand, all Uniprot identifiers provided by Neji were mapped to corresponding Entrez Gene entries. Considering the Uniprot to PO mapping, only 22% of the identifiers provided by Whatizit were successfully mapped, while a PO identifier could be assigned to 78% of the recognized concepts. Regarding Neji, 95% of the Uniprot IDs were mapped to PO, and a PO identifier was assigned to 99% of the recognized concepts. Various facts contribute to identifier mapping discrepancies between the two systems: 1) Neji uses Uniprot entries for 21 species while Whatizit uses the entire Uniprot database, resulting in more concept names and much more Uniprot identifiers; 2) the version of Uniprot used by Whatizit may not correspond to the version used for mapping; 3) not all Uniprot entries have a corresponding Entrez Gene entry; and 4) protein ontology does not map to all entries of Uniprot. Regarding cell identifiers mapping, 64% of the UMLS identifiers were successfully mapped into CO identifiers, resulting in 91% of the recognized concept names having CO identifiers. Finally, since Neji uses both GO and UMLS for representing Biological Process and Molecular Function concepts, we analyzed the mapping between the provided UMLS identifiers and corresponding GO entries. Considering only the annotations that contain UMLS identifiers, only 32% of the recognized concept names were mapped with GO identifiers. Overall, considering both UMLS and GO, 81% of the recognized concept names were provided with GO identifiers.

Table 5.1: Statistics of mapping identifiers between different resources for cell, gene and protein, and biological process and molecular function concept names. The analysis considers the number of identifiers and concept names provided by each solution and the percentage that were successfully mapped.

	From	To	Solution	# Identifiers	Mapped identifiers	# Concept names	Mapped concept names
Gene and Protein	Uniprot	Entrez Gene	Neji	51118	100%	13239	100%
			Whatizit	123136	53%	18079	95%
	Uniprot	Protein Ontology	Neji	51118	95%	13239	99%
			Whatizit	123136	22%	18079	78%
Cell	UMLS	Cell Ontology	Neji	8390	64%	5926	91%
Biological Process and Molecular Function*	UMLS	Gene Ontology	Neji	6079	28%	5377	32%

*Only concept names with UMLS identifiers are considered.

Figure 5.9 presents the results achieved by Neji and Whatizit in the CRAFT corpus, considering the various strategies for matching the text chunks to the entries in the dictionary and the two identifier matching techniques ("exact" and "contains"). Overall, Neji considerably outperformed Whatizit on identifier matching for Species, Cellular Component, Chemical and Biological Process and Molecular Function concept names, with the exception of Gene and Protein concepts, where both solutions presented similar results. Moreover, there was no high variability in identifiers matching when the various dictionary matching strategies were compared, again with the exception of gene and protein concept names. In this case, it is clear from the results that different annotation characteristics between the train and test corpora also have a substantial impact on the normalization performance. On the other hand, there is a significant difference in the results if we require that the correct identifier is returned ("exact") or that the correct identifier is included in the returned list of identifiers ("contains"), highlighting the ambiguity in the concept names recognized in the texts.

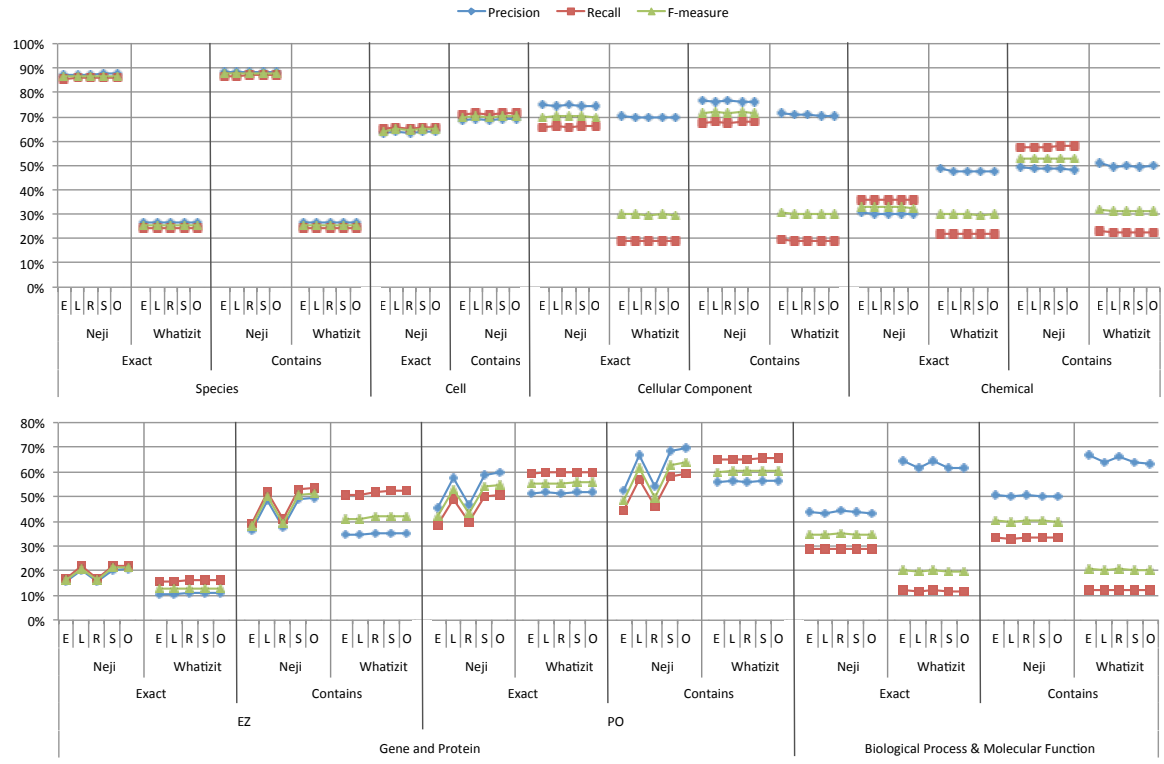


Figure 5.9: Evaluation results for normalization considering precision, recall, and F-measure achieved on CRAFT corpus, using exact (E), left (L), right (R), shared (S) and overlap (O) names matching and "exact" and "contains" matching of identifiers. Evaluation considers species, cell, cellular component, gene and protein, chemical, biological processes and molecular functions concept names.

Neji obtained state-of-the-art results in the recognition of species, with an F-measure of 87.8% and no significant variance between "exact" and "contains" matching of identifiers.

During the annotation of species in CRAFT, experts were required to assume the closest semantic match, which means that the mention "rat" was annotated as the genus "Rattus" (NCBI identifier 10114), even if from context it was known to be the common laboratory rat species "Rattus norvegicus" (NCBI identifier 10116). Such fact considerably affected the performance of Whatizit, since it only provides more specific species identifiers. For example, by considering just two of those cases and converting from "Rattus" (NCBI:10114) to "Rattus norvegicus" (NCBI:10116) and "Mus" (NCBI:10088) to "Mus musculus" (NCBI:10090), Whatizit results would achieve an F-measure of 87.5%, similar to that achieved with Neji.

Neji presented competitive results on Cell concepts normalization, with a small variance between "exact" (64.9% of F-measure) and "contains" identifier matching (70.5% of F-measure). Such results represent a small drop when compared with the performance obtained on exact named entities matching (F-measure of 75.4%). Regarding GO concept types, namely Cellular Component, Biological Process and Molecular Function, Neji considerably outperformed Whatizit, again with a small difference between "exact" and "contains" matching of identifiers. Considering "contains" matching, Neji presents an F-measure of 71.8% on Cellular Component, and 40.1% of F-measure on Biological Process and Molecular Function. Comparing those results with exact named entity matching, they represent an average drop of 8% of F-measure. The performance on Biological Process and Molecular Function is affected by the absent mappings between some UMLS concepts and GO identifiers.

Neji also outperformed Whatizit on Chemical concepts normalization, with an F-measure of 33.1% on "exact" and 53.1% on "contains" identifier matching. The high difference between "exact" and "contains" matching reflects the high ambiguity present on ChEBI. For instance, the annotation "protein" on CRAFT contains the ChEBI identifier 36080 ("protein"), but the dictionary matching provides both 36080 and 16541 identifiers, which corresponds to "protein polypeptide chain" and also contains "protein" as a synonym. The best normalization results were achieved when exact named entity matching was considered, which shows that accepting approximate matching of named entities may degrade normalization performance by leading to more false positives identifiers.

Finally, in order to present results for Gene and Protein concepts, two different evaluations were performed: 1) against Entrez Gene identifiers; and 2) against Protein Ontology identifiers. On both evaluations and systems, there was a considerable variation between the various names matching strategies and between "exact" and "contains" identifier matching, a consequence of the cross species ambiguity of gene and protein names. Regarding Entrez Gene, Neji and Whatizit present low performance results on "exact" identifier matching, achieving F-measures of 21.4% and 13.0%, respectively, when using overlap dictionary matching. When "contains" identifier matching was considered, the performance of Neji and Whatizit improved considerably, achieving F-measures of 52% and 42% for overlap dictionary matching, respectively. Concerning normalization to Protein Ontology, the achieved

performance results are considerably better, with Neji and Whatizit achieving F-measures of 55.0% and 55.6%, respectively, for "exact" identifier matching and using overlap dictionary matching. When "contains" matching was considered, both solutions presented considerable improvements, with Neji achieving 64.0% of F-measure and Whatizit 60.7%. Evaluating the normalization to both EZ and PO, Whatizit presented the most constant behavior, a consequence of the different annotation guidelines followed in CRAFT and in the training corpus used to generate the ML model used by Neji. However, when all evaluation strategies are considered, Neji provides better results.

Overall, the presented analysis shows that Neji achieves competitive performance results on normalization, presenting small and anticipated performance drops when compared to named entities evaluation. Nonetheless, we consider that there is still margin for improvement, namely for chemicals and gene and protein normalization.

AnEM

To evaluate the recognition of anatomical concepts, we combined all sub-classes of the AnEM corpus into a single class. As a consequence, the systems were evaluated targeting the general ability to recognize anatomical entities, discarding the capability to classify and distinguish specific sub-anatomical classes. Thus, Neji was compared with the systems used in Ohta et al. [105], i.e. MetaMap and NERSuite, which provide state-of-the-art results on this corpus. NERSuite was trained using the training part of the corpus, being optimized for these specific annotation guidelines. Cocoa provides anatomical classes following the AnEM classification approach. Thus, we annotated the corpus using Cocoa and mapped the respective classes to the single anatomical class. Body part concepts provided by Cocoa were also mapped to the single class.

Figure 5.10 compares the results achieved by Neji, Cocoa, MetaMap and NERSuite on AnEM corpus, considering exact, left, right, shared and overlap names matching. Overall, there is a significant variation between the various matching techniques, which is observed in all systems. Even NERSuite has problems to identify the exact names' boundaries, namely the right boundary. Such variation reflects the complexity of inferring the variable boundaries of anatomical names. Nonetheless, Cocoa was the system that presented better results, with 83.5% of F-measure on overlap matching. Neji also presented competitive results, with 83.1% of F-measure on overlap matching. On the other hand, MetaMap was the system that performed worst. Surprisingly, NERSuite did not perform better than Neji and Cocoa, which may indicate that ML-based solutions are not required for the general recognition of anatomical entities.

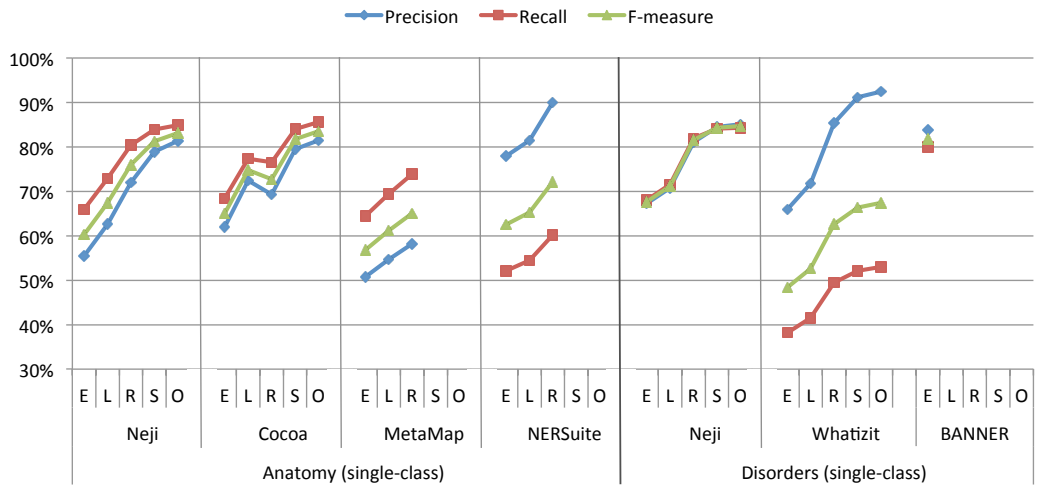


Figure 5.10: Comparison of precision, recall, and F-measure results achieved on AnEM and NCBI corpora, considering exact (E), left (L), right (R), shared (S) and overlap (O) matching. The various sub-classes from each corpus were merged into a single class, in order to evaluate the general ability to recognize disorder and anatomical concept names.

NCBI

Similarly to the AnEM corpus, we also combined NCBI sub-classes into a single class, in order to evaluate the general ability to identify names of disorders. The comparison was performed against BANNER and Whatizit. BANNER was used in Doğan and Lu [308] to present state-of-the-art results for ML-based solutions in this corpus. Although our approach is not ML-based and therefore not trained using the corpus, we believe this comparison is also relevant to provide feedback regarding the overall performance. Whatizit was used through the "whatizitDiseaseUMLSDict" pipeline.

Figure 5.10 compares the named entity recognition results achieved by Neji, Whatizit and BANNER on the NCBI corpus. There is also a significant variation between the various matching techniques, namely on right matching. This means that various concepts are not precisely identified due to the presence or absence of word prefixes. For instance, in our case, the gold standard annotation "atrophic benign epidermolysis bullosa" was typically provided just as "epidermolysis bullosa". Even though the text chunk is not correct, it points to the same concept. Comparing the two dictionary-based approaches, Neji presented significantly better results than Whatizit, with an improvement of more than 17% of F-measure on overlap matching. On the other hand, BANNER, a ML-based solution trained on this corpus, achieved significantly better results than Neji when exact matching is considered. However, the high-performance results obtained with Neji when fuzzy matching is used, seem to indicate a mismatch between the terms in the dictionary used and the annotation guidelines for this corpus.

Summarizing, we can argue that Neji presents highly competitive results, with significant

improvements for some semantic groups, namely species, cell, cellular component, gene and protein, and anatomy.

5.3.4 Speed evaluation

One important characteristic of concept recognition solutions is annotation speed, since large data sets may be annotated to collect as much information as possible. To evaluate the annotation speed achievable with Neji, various experiments were performed using the CRAFT corpus, which contains 67 full-text articles with 21749 sentences. The documents were processed on a machine with 8 processing cores @ 2.67 GHz and 16GB of RAM.

The annotation process using the dictionaries and ML model previously described and using 5 threads took 124 seconds, which corresponds to processing 175 sentences per second, and around 1.8 seconds to process a full text article. Considering that MEDLINE contains 11 millions abstracts[6], and that each abstract contains on average 7.2 sentences [109], this configuration could annotate the entire MEDLINE in five days. Since generating the complex features for the ML model and collecting POS and chunking features is resource intensive, we also measured the processing speed without using ML, applying only dictionary matching and tokenization from the NLP module. With this configuration, the CRAFT corpus was processed in 18 seconds, which corresponds to 1208 sentences/second. Thus, a full text article was processed in 0.28 seconds, and the entire MEDLINE could predictably be annotated in 18 hours. To contextualize the achieved results, we compared Neji with other existing tools. Even though BANNER applies ML for gene and protein names recognition only, it took more than 9 minutes to annotate CRAFT. On the other hand, the rule-based solution MetaMap took more than 2 minutes to process a single full-text file. We believe that the presented processing speeds provide a positive contribution to the biomedical community, making annotation of large data sets with dozens of biomedical concepts easily accessible.

5.3.5 Real-time annotation

Since the availability of no-installation, no-maintenance and modular solutions for heterogenous biomedical concept recognition is still scarce, we decided to take advantage of Neji to offer a complete web-service to be easily integrated in any text-processing pipeline. Whatizit [126], for instance, offers dictionary-based annotation of documents with a large set of vocabularies and is available both through a web-service and a web page. Yet, annotation of concepts from different types is only possible by repeating the annotation process several times and combining the results generated by different pipelines. iHOP [18] is a web-application offering programmatic access to pre-annotated abstracts from MEDLINE. It is a protein-centric system and does not allow the annotation of external documents submitted by the user. Another solution focused on genes, proteins and small-molecules is Reflect [309],

[6] `http://www.nlm.nih.gov/bsd/medline_lang_distr.html`

a web-service that annotates these concepts on web pages and provides, through popups, additional information such as synonyms, database identifiers and related literature. Cocoa[7] is a multiple concept annotator with an online interface and an Hypertext Transfer Protocol (HTTP) API. It annotates entities in user submitted text, but it is limited to named entities and does not provide concept identifiers or external references. Overall, only few text-mining solutions for concept identification are available as web-services and most of them focus on a small number of concept types. Of those, most omit concepts that intersect other recognized concepts or that are nested within broader concepts. Moreover, to the best of our knowledge, there is no solution available that allows users to select the entity types they want to annotate on a single service invocation. BeCAS, the Biomedical Concept Annotation System, is a web-based tool for on-demand document processing and annotation that can be integrated on larger text-processing pipelines, used directly through a user-friendly and highly interactive web interface or incorporated on external web pages through a simple yet flexible widget.

BeCAS concept recognition features take advantage of Neji, by applying the previously described processing modules, dictionaries and ML models. Representational State Transfer (REST) web services were built in Java on top of Neji, pre-loading and keeping in memory all dictionaries, models and parsers for on-demand usage. The article fetching modules were built in Python and the web interface was developed using HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and Javascript. In summary, BeCAS exposes its functionalities through three interfaces: an HTTP REST API, a widget embeddable in web pages and an interactive web-application. It provides annotations both for user-supplied texts and for MEDLINE abstracts, which are automatically fetched from PubMed.

BeCAS web interface was built with a strong focus on usability. Specific entity types can be highlighted or muted in real-time by using simple toggle controls, and nested and intersected annotations are also easily identified by the colour coding scheme used. An infobox with links to external databases is displayed by placing the mouse over highlighted entities and users can explore this same information, grouped by concept type, through the concept tree (Fig. 5.11). Annotated text can be exported in several formats such as JSON and A1. Users and other websites can link to annotated PubMed publications by using direct links (e.g., `http://bioinformatics.ua.pt/becas/pmid/22957306`).

Concept highlighting with external references can easily be integrated in any website through the use of the BeCAS Javascript widget. Host pages only need to include a <script> tag linking to the plugin and a few configuration parameters. Every feature implemented in the main web interface is exposed by the widget, apart from the concept tree.

Text can be annotated programmatically using one of BeCAS HTTP REST endpoints. Clients should make HTTP POST requests to one of the endpoints with a JSON encoded payload, specifying the text to annotate, the desired output format and types of entities that

[7] `http://npjoint.com`

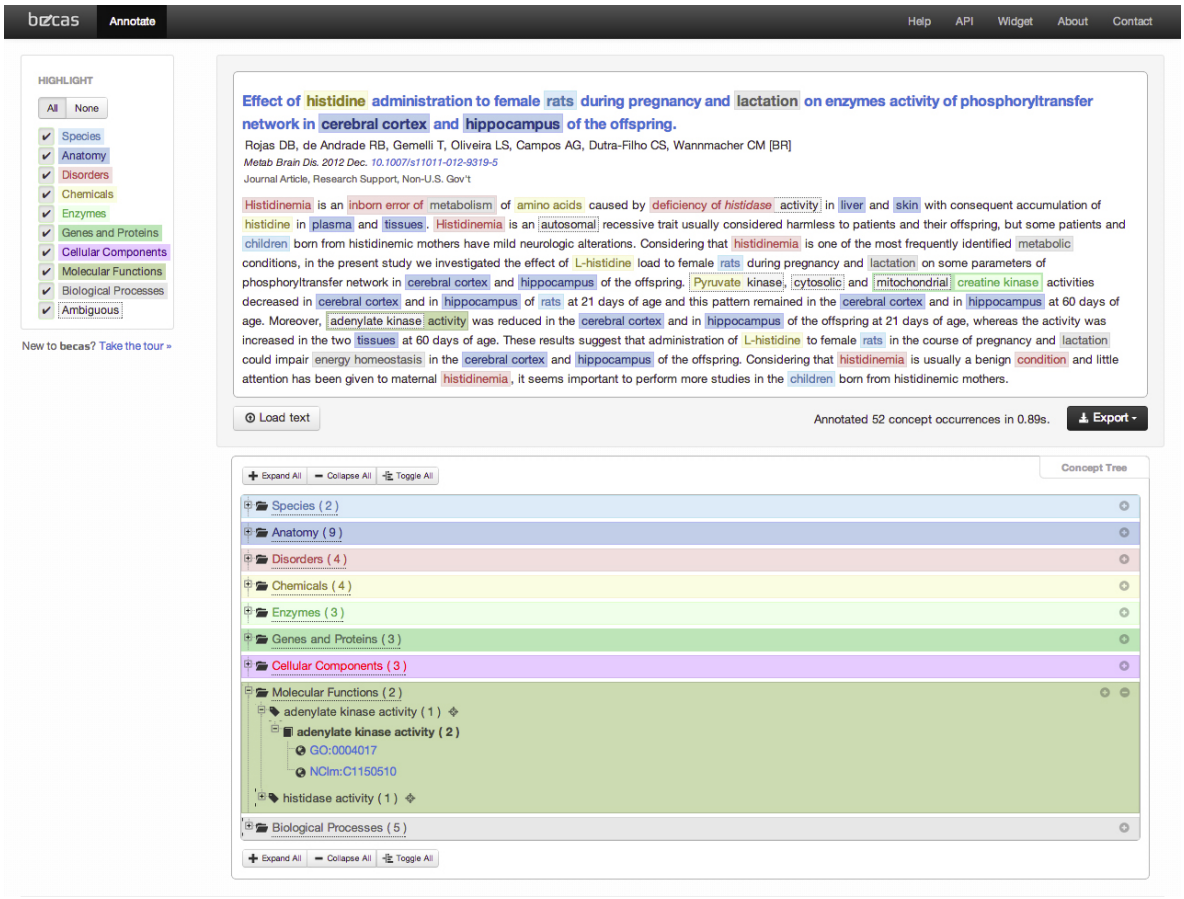


Figure 5.11: BeCAS Web interface showing an annotated PubMed abstract. Each annotated entity type can be highlighted separately (left). The concept tree (bottom) displays all annotations along with the associated concepts and external references.

should be annotated. Due to inherent representation constraints, the available output formats support different levels of granularity in the results. CoNLL format is the most comprehensive, providing sentence splitting, tokenization, lemmatization, POS tagging, chunking and identification of isolated, nested and intersected concepts. JSON format includes sentence splitting and concept identification. IeXML formatted results contain the same information as JSON, but nested and intersected annotations are limited to a depth of one level, with deeper annotations resolved to the largest span. Results in A1 format provide concept identifiers, including nested and intersected annotations.

Apart from supplying text directly to the API, BeCAS is capable of fetching and annotating PubMed articles. A client can issue an HTTP POST request to one of the abstract annotation endpoints, optionally providing a JSON encoded payload of entity types for annotation. Since publications have multiple fields, such as the title, abstract, authors, MeSH terms and others, results are provided exclusively as PubMed annotated IeXML or JSON. The service returns XML documents delivered by the Entrez eFetch Utility, with the "Arti-

cleTitle" and "AbstractText" fields enriched with IeXML annotations.

Overall, BeCAS provides three distinct user interfaces for biomedical concept identification, presenting state-of-the-art performance, as evaluated on various corpora. It currently recognizes and annotates 1.2 million concepts and enriches them with 1.6 million external references to 30 online resources. The REST API is suitable for integration in custom text-processing pipelines, while the widget can be easily integrated in any web page. Finally, users can also use BeCAS annotation services as a standalone web-application. In the future, we plan to add support for more entity types and continue to improve annotation performance, with focus on concept disambiguation.

5.4 Discussion

The inherent characteristics, features and performance provided by the Neji framework represent various technical and theoretical advantages to end-users, contributing to an improved and faster research in biomedical text mining and information extraction. First of all, the large dictionaries used in our experiments, in combination with the achieved processing speeds, are good indicators of the scalability of the presented solution. Additionally, the achieved high-performance results against gold standard corpora show the solution's reliability. Overall, the flexibility, scalability, speed and performance results offered by the proposed framework expedite the processing of the increasing scientific biomedical literature. The features provided greatly simplify NER and normalization tasks, offering annotations for a large number of entity types using both dictionary and machine learning-based approaches. Using the state-of-the-art modules incorporated in Neji, developers and researchers can bypass normally complex and time-consuming tasks, allowing them to focus on further analysis of these annotations. Users can also take advantage of the integrated natural language processing tools, eliminating the need for developing wrappers or integration solutions. The adoption of the same techniques for linguistic processing means that all modules are based on the same consistent information, such as tokens, lemmas, POS tags, chunks and parsing trees. This approach builds an integrated development ecosystem that minimizes cascading errors. For instance, if concept recognition is performed using linguistic information from one parser, and relation extraction is performed afterwards using information provided by another parser, it is hard to keep consistency between the two solutions, since the application of distinct sentence splitting and tokenization techniques provide different and hard to combine interpretations of data. Thus, performing all tasks using the same linguistic information will deliver better and more consistent results.

Besides using the provided modules directly, researchers may also adapt them or integrate new ones, allowing the construction of specialized processing pipelines for text mining purposes. As presented, Neji is ready to be used by users with different levels of expertise. It

allows obtaining heterogeneous concepts of several types in a straightforward way, by using the CLI tool or by building a pipeline with existing modules. Users also have the power to optimize concept recognition for their specific goals, which is achieved by having access to the innovative concept tree. Such structure supports both nested and intersected annotations and, combined with the support for multiple identifiers from different semantic groups per concept, enables easy detection of ambiguity problems. Additionally, Neji also integrates helpers for simple concept disambiguation, merging nested annotations and selecting intersections. If required, users can also develop their own modules, such as readers, writers or WSD. Overall, Neji was built considering different development configurations and environments: a) as the core framework to support all developed tasks; b) as an API to integrate in your favorite development framework; and c) as a concept recognizer, storing the results in an external resource, and then using your favorite framework for subsequent tasks.

Besides the flexible CLI tool, we also developed BeCAS, a web-page, set of web-services and widget for on-demand biomedical concept recognition, taking advantage of Neji's processing speeds and flexibility. The web-services are suitable for integration in custom text-processing pipelines, while the widget can be easily integrated in any web page. Finally, users can also use BeCAS annotation as a standalone web-application.

A large and diverse set of annotations can be obtained by processing a large set of documents. Such annotations can be exploited in various ways. Perhaps, the most straightforward one is to use these annotations together with the provided identifiers and connections to ontologies and other domain resources, to support a semantically enabled literature retrieval system [310–312]. Using these annotations, it also becomes simpler to implement a query expansion scheme [313], taking advantage of the ontological relationships between the identified concepts. Another use of such annotations is to extract co-occurrence based association metrics between concepts [20, 314]. This can also be extended to extracting semantic concept profiles that represent the semantic context in which a given concept occurs, as described in [270]. Creating these profiles is highly dependent on the annotation of a large set of documents with diverse and rich concepts from various semantic types. Co-occurrence and context-based association metrics can in turn be exploited for discovering implicit (A-B-C) concept relations from the literature, therefore supporting hypothesis generation and knowledge discovery.

With this analysis, we show that Neji is a good starting point to develop complex biomedical text mining projects, supporting advanced and reliable features and giving users the power to choose the best behaviors considering the complete tree of recognized concepts and their specific goals.

5.5 Summary

This chapter presented Neji (`http://bioinformatics.ua.pt/neji`), an open source and modular framework optimized for general biomedical concept recognition. It was developed considering scalability, flexibility, speed and usability. Neji integrates state-of-the-art and optimized solutions for biomedical natural language processing, such as sentence splitting, tokenization, lemmatization, POS tagging, chunking and dependency parsing. Concept recognition is supported through dictionary matching and machine learning, integrating features to perform normalization of recognized chunks of text. Various known biomedical input and output formats are also supported, namely Raw, XML, A1 and CoNLL. Recognized concepts are stored in an innovative concept tree, supporting nested and intersected concepts with multiple identifiers. Such structure provides enriched concept information and gives users the power to decide the best behavior for their specific goals, using the included methods for handling and processing the tree.

The application of Neji on real life problems was also presented, achieving high-performance results when evaluated against manually annotated corpora. To the best of our knowledge, the analysis presented constitutes the most comprehensive evaluation of named entity recognition and normalization for such a heterogeneous set of biomedical concept types. Additionally, the presented processing speeds make the annotation of large document sets a reality. We also described the simple usage of Neji through the integrated CLI tool, which allows annotating thousands or millions of documents with a simple bash command. Furthermore, we illustrated the simplicity of developing a custom pipeline using existing modules. In the end we describe BeCAS (`http://bioinformatics.ua.pt/becas`), a web-page, set of web-services and widget for on-demand biomedical concept recognition built on top of Neji.

We believe that the characteristics and complex features provided by Neji fill the gap between general frameworks (e.g., UIMA and GATE) and more specialized tools (e.g., NER and normalization). It streamlines and facilitates biomedical concept recognition, using both dictionary and machine learning-based approaches to extract multiple concept types in an integrated ecosystem. Neji simplifies concept recognition tasks in biomedical information extraction, and it can be easily integrated in complex workflows contributing towards more accurate knowledge discovery.

Chapter 6

TrigNER: biomedical event trigger recognition

This chapter is based on:

- D. Campos, Q.-C. Bui, S. Matos, and J. L. Oliveira, "TrigNER: automatically optimized biomedical event trigger recognition on scientific documents," *Source code for biology and medicine*, vol. 9, no. 1, Jan. 2014

Automatic extraction of biological events from text constitutes an important contribution for the biomedical community, in order to help find hidden relationships and allow faster updating of existing knowledge. As previously described in Section 2.3, the development of automatic solutions to extract biomedical events from scientific documents has been greatly promoted by the BioNLP shared tasks [192, 193], aimed at the recognition of events particularly focused on genes and proteins. More recently, the extraction of events focused on infectious diseases, bacteria and cancer genetics were also targeted. In general, the proposed approaches to event extraction consist of two subsequent sub-tasks:

- Trigger recognition: aimed at identifying the chunk of text that indicates the event and serves as a predicate;
- Argument recognition: aimed at identifying the entities and/or event that take part in the event.

6.1 Background

Trigger recognition is the first and crucial task of event recognition, since the following task(s) completely rely on its output. This was clearly shown by Björne et al. [239], who stated

a drop of more than 20 points in performance between using predicted and gold standard triggers. However, trigger recognition presents various complex and unsolved challenges, namely:

- The same chunk of text may be a trigger word or not depending on the textual context;
- The same chunk of text may be a trigger of two or more event types;
- Triggers of different event types have different linguistic characteristics;
- Large amount and variety of event types.

As described in Section 2.3.3, approaches to perform event trigger recognition can be categorized as being based on rules, dictionary matching and machine learning. ML-based approaches were the most commonly used in previous BioNLP event extraction challenges, followed by dictionary-based systems and rule-based solutions. Regarding performance behavior, ML-based solutions present the best results, followed by dictionary matching approaches. However, current ML-based approaches still present various limitations, namely:

- The problem of a single chunk of text with multiple trigger types is not properly and generally solved;
- Current solutions do not consider the heterogeneous linguistic characteristics of different event types;
- Feature set selection is typically performed manually;
- Availability of open source solutions is limited;
- Existing solutions are not usually configurable and/or extendable, limiting their application in different domains and with different event types.

This chapter proposes an innovative, open source and high performance machine learning-based approach for event trigger recognition, aimed at minimizing the aforementioned limitations. It takes advantage of a high-end feature set and is focused on automatic optimization per event type. Such a method makes the application of complex trigger recognition techniques a simple routine task, contributing to improved and faster biomedical event recognition.

6.2 Methods

This section presents the applied processing pipeline and supporting data structure, which will serve as support to extract linguistic features and train machine-learning models to automatically recognize triggers.

6.2.1 Pipeline

Since a trigger recognition solution must be combined with other methods to perform event extraction, such a system must be implemented on top of a modular and flexible architecture, in order to allow easy integration of new modules and respective features. Thus, our

solution was developed on top of Neji [31], an open source framework that provides a modular processing pipeline for biomedical concept recognition. Neji integrates various modules optimized for the biomedical domain, such as natural language processing (sentence splitting, tokenization, lemmatization, part-of-speech tagging, chunking and dependency parsing) and concept recognition (dictionaries and machine learning). Popular biomedical input and output formats are also supported. The processing pipeline applied in our system is illustrated on Figure 6.1, which contains the following general modules and steps:

- Reader: read input data and mark the text regions of interest;
- NLP: perform sentence splitting using LingPipe, and tokenization, lemmatization, POS tagging, chunking and dependency parsing using a custom version of GDep [58], with an optimized tokenization;
- Concept loader: load relevant concepts;
- Dictionary tagger: perform trigger recognition using one or multiple previously built dictionaries;
- Machine learning: perform trigger recognition using one or multiple previously trained models;
- Post-processing: remove false positive trigger names through rule-based approaches;
- Writer: write the output to an external resource.

6.2.2 Data structure

After reading input data in RAW format and performing NLP processing, it is fundamental to store relevant linguistic information in a structured manner, in order to facilitate further processing. Figure 6.2 illustrates the internal data representation to support all the information associated with a corpus. The core components are sentences and tokens, which provide their relative positions regarding the input text. Chunking output is stored using the target token positions and a label for the corresponding chunk type. Moreover, dependency-parsing output is stored as an undirected graph, where nodes are tokens and edges contain labels to describe each linguistic dependency. Such graph representation allows easy traversing of the various dependencies and extracting paths for any given token. The graph implementation is based on the JGraphT library[1], which contains methods to simplify path and shortest path construction.

The support for other features and/or information associated with each token is provided through a map of keys and values, where a key identifies a type of feature and the value is the feature itself. However, since each feature type may contain multiple values, the mapping is performed between a key and a list of values. This implementation is based on a Multimap from the Guava library[2]. Thus, since lemmas and part-of-speech tags are specific to each

[1] `http://jgrapht.org`
[2] `https://code.google.com/p/guava-libraries`

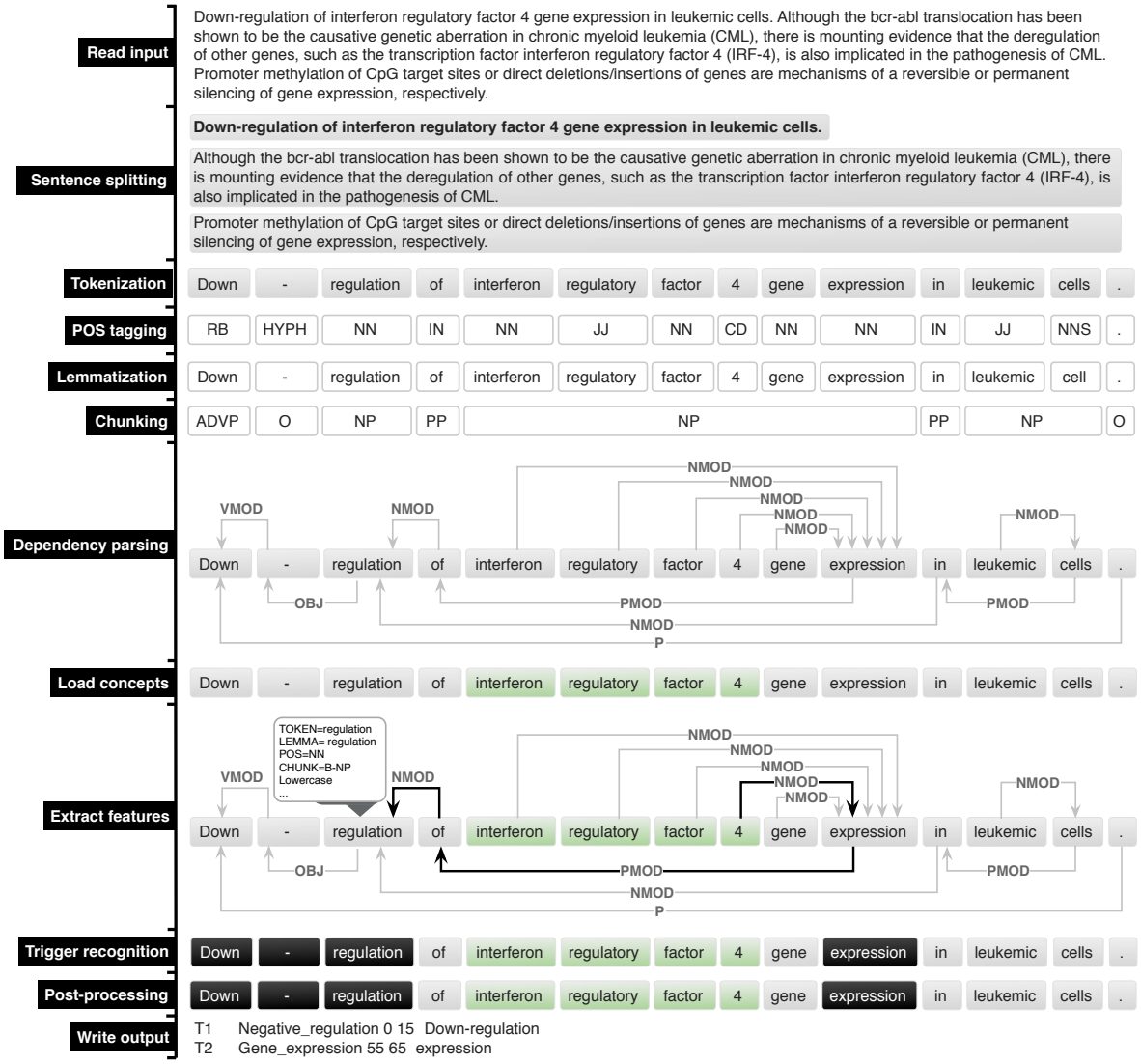


Figure 6.1: Illustration of the processing pipeline for the sentence "Down-regulation of interferon regulatory factor 4 gene expression in leukemic cells.", highlighting the output of linguistic parsing, shortest paths, provided concepts and extracted triggers.

token, they are provided as features in the multimap. Moreover, to cope with nested and intersected concept and trigger annotations, it is important to integrate a data structure that suits such characteristics in the best and most automated way. This is achieved through a tree of annotations, which offers various advantages over typical approaches (e.g., list of annotations), such as automatic maintenance of structured annotations and easy identification of ambiguity problems. The extracted and stored information is also illustrated on Figure 6.1.

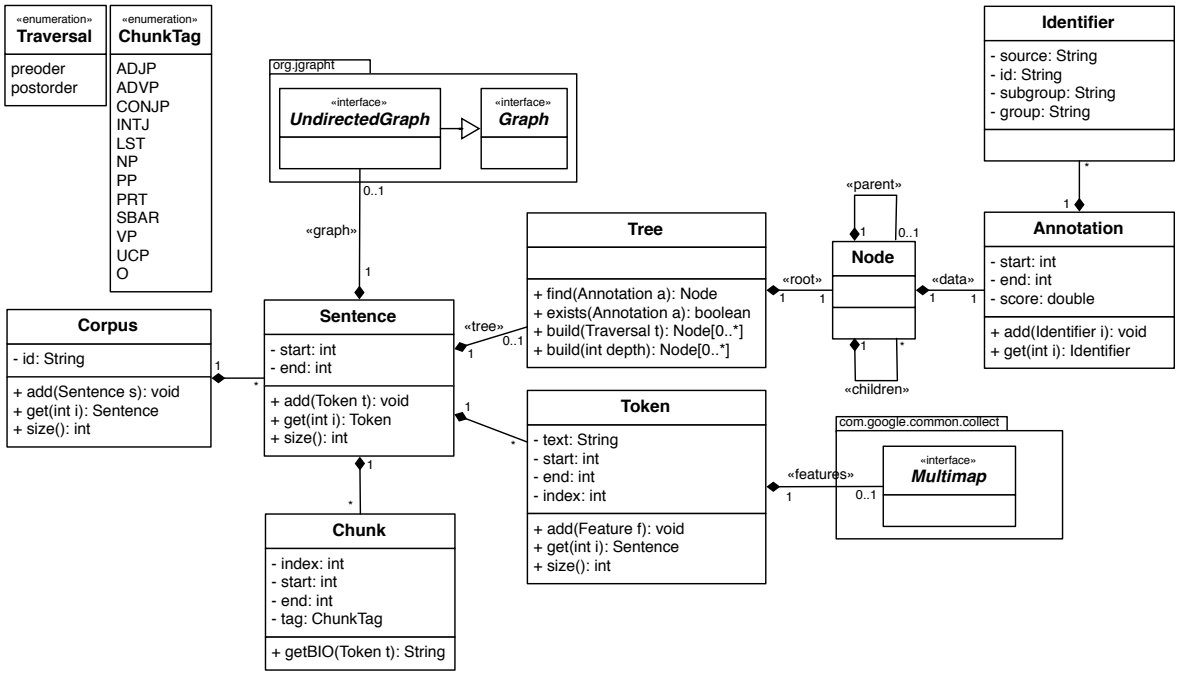


Figure 6.2: Internal data structure to support a corpus with multiple sentences and associated information, namely tokens, chunks, dependency parsing graph, concept tree and features.

6.2.3 Modules

Loading concepts

Since the extraction of biomedical events requires previous annotation of biomedical concepts, we support both loading and automatically identifying those concepts in the texts. If manual annotations are available, they should be provided in A1 format. On the other hand, dictionary or machine learning-based approaches can be applied to perform automatic recognition of such biomedical concepts.

Dictionary matching

When data containing manual annotations of event triggers are unavailable or scarce, training machine learning models may not be possible. Thus, we also provide the ability to perform trigger recognition using dictionaries. Such functionality is achieved by case-insensitive exact dictionary matching, using DFA through a custom version of the dk.brics.automaton library. Dictionaries are provided in TSV files with two fields: identifier and respective list of names. The responsibility for building such dictionaries is left to the user.

Machine learning

When ML techniques are applied to trigger recognition, an algorithm must build a feature and statistic-based representation of target trigger words from training data, in order to develop an appropriate response to unseen data. Such methodologies are commonly categorized as being supervised or semi-supervised. Semi-supervised solutions use both annotated and unannotated data, in order to obtain features of the trigger words that are not present in the annotated data. Specifically for this task, the use of unannotated data could contribute to a better abstract learning of triggers. However, the application of such techniques is computationally heavy and could be implemented as an extension to an equivalent supervised solution. Thus, we decided to follow a supervised training approach, through the application of CRFs [139]. The support for CRF models is provided through Gimli [28], an open-source biomedical concept recognition tool based on the MALLET framework that provides high-performance results in two well-known corpora: GENETAG [92] and JNLPBA [91]. Gimli implements a comprehensive set of features optimized for the biomedical domain, therefore serving as a good starting point for trigger recognition.

The proposed solution supports a complex and high-end feature set, extracting features based on tokens, sentences, concepts, dependency parsing trees and external resources. On top of those, different strategies to model local context are also provided.

Token Token-based features intend to capture specific knowledge regarding each token, namely linguistic, orthographic and morphological characteristics. The most basic feature is the token text. However, in most cases, morphological variants of words have similar semantic interpretations, which can be considered as equivalent. For this reason, lemmatization is used to group together inflected forms of a word, so that they can be analyzed as a single item. On the other hand, it is also possible to associate each token with a particular grammatical category based on its definition and context, a procedure called POS tagging. Moreover, we also use chunking, dividing the text into syntactically correlated chunks of words (e.g., noun or verb phrases). The BIO encoding format is used to properly indicate the beginning and end of each chunk. For instance, considering two consecutive tokens that constitute a noun phrase chunk, the tag "B-NP" is associated with the first token and the tag "I-NP" with the second one. In the end, each tag is used as a feature of the respective token. Regarding orthographic features, their purpose is to capture token formation characteristics, through three different types of features:

- Capitalization: reflect uppercase and lowercase characteristics, such as "InitUpp" (token starts with uppercase character) and "MixCase" (token has both lowercase and uppercase characters);
- Counting: count the number of uppercase characters and numbers, and provide token length;

- Symbol: reflect the occurrence of symbol characters, such as dots, commas and semicolons.

On the other hand, morphological features reflect common structures and/or sub-sequences of characters among several tokens, identifying similarities between distinct triggers. Three different types of morphological features are considered: suffixes and prefixes, char n-grams and word shape patterns. Particular prefixes and suffixes could be used to distinguish trigger names, such as the 2-character prefix "co" for the "coexpression" trigger. A char n-gram is a subsequence of n characters from a given token, which finds common sub-sequences of characters in the middle of tokens. Finally, it is also important to extract the token's structure, reflecting how letters, digits and symbols are organized in the token. For instance, the structure of "Abc:1234" is expressed as "Aaa#1111".

Sentence Sentence based features intend to reflect general characteristics of the sentence where the target token is present. Features are provided to reflect the number of tokens present on each sentence. Considering an average number of 25 tokens per sentence, we decided to generate the following seven clusters: 1) less than 15 tokens; 2) between 15 and 20 tokens; 3) between 20 and 25 tokens; 4) between 25 and 30 tokens; 5) between 30 and 35 tokens; 6) between 35 and 40 tokens; and 7) more than 40 tokens.

Concepts These features reflect information regarding the concept annotations previously provided, such as gene and protein names. Four different types of concept-based features are generated:

- Tags: a tag is provided when the token is part of a concept name, such as "Concept=Protein";
- Names: the names of the concepts in the sentence are also added as features. When the concept name contains more than one token, it is concatenated with "_". For instance, considering the protein in Figure 6.1, the feature "CONCEPT_NAME=interferon_regulatory_factor_4" is added to all the tokens in the sentence;
- Heads: a feature is added to reflect the head token of the concept name. For instance, considering the protein name "interferon regulatory factor 4" (Figure 6.1), the feature "CONCEPT_PROTEIN_HEAD=interferon" is added to all the tokens in the sentence;
- Counting: a feature is added with the number of annotations per concept type in the sentence. For instance, if the sentence containing the token has two genes and one chemical annotation, the features "NUM_PROTEIN=2" and "NUM_CHEMICAL=1" are added to each token in the sentence.

External resources Further optimization can be achieved by adding biomedical knowledge to the feature set. To provide this knowledge, dictionaries of specific domain terms and trigger

words are matched in the text and the resulting tags are used as features. Thus, the tokens that are part of a matched term contain a feature that reflects such information. For instance, if a dictionary of gene expression triggers is provided, and the token "coexpressed" is matched, the feature "Trigger=Gene_expression" is added to the token.

Dependency parsing The previous features provide a local analysis of the sentence. To complement these with information about relations between the tokens of a sentence, we use features derived from dependency parsing. First, we consider modifier features that could indicate the presence of a trigger word. This is done by adding as features of each token, the lemmas corresponding to each of the following: verbs for which the token acts as subject; verbs for which the token acts as object; nouns for which the token acts as modifier; and the modifiers of that token.

Features to reflect input and output dependencies are also added, considering inherent dependency, lemma, POS and chunk tags. For instance, regarding the sentence of Figure 6.1 and the token "regulation", the following features are added:

- Input dependencies:
  - "IN_DEP_LABEL=NMOD";
  - "IN_DEP_LEMMA=in";
  - "IN_DEP_POS=PP";
  - "IN_DEP_CHUNK=PP";
- Output dependencies:
  - "OUT_DEP_LABEL=OBJ";
  - "OUT_DEP_LEMMA=-";
  - "OUT_DEP_POS=HYPH";
  - "OUT_DEP_CHUNK=O".

By analyzing the dependency parse graph, we can find the shortest paths between two different tokens, by applying the Dijkstra's algorithm [315]. Since biomedical events and their triggers rely on entity names, it should be informative to extract features to reflect the relation between each token and the closest entity name. For instance, as illustrated in Figure 6.1, the shortest path between the token "regulation" and the closest entity "interferon regulatory factor 4", is "regulation-of-expression-4". Specific to shortest paths, we provide a feature to reflect the shortest distance between the current token and the closest entity name. Again, considering the token "regulation" on Figure 6.1, it should contain the feature "SPDistance=3", which is the number of hops between the token and the closest entity.

For both dependency and shortest paths, the following features are added (examples are based on the tokens "regulation" and "4" of Figure 6.1):

- Edge path: path of edge labels between two tokens (e.g., "NMOD-PMOD-NMOD");

- Edge type: reflect the type of path based on its size and first edge label (e.g., "NMOD_-3");
- Vertex path: path of features of tokens (vertexes) between two tokens (e.g., "regulation-of-expression-4", considering lemmas as features);
- Edge n-grams: n-grams of edge labels between two tokens (e.g., "NMOD_PMOD" and "PMOD_NMOD", considering 2-grams);
- Vertex n-grams: n-grams of features of tokens (vertexes) between two tokens (e.g., "regulation_of", "of_expression" and "expression_4", considering 2-grams and lemmas as features).

Context Higher-level relations between tokens and extracted features can be established through windows or conjunctions of features, reflecting the local context of each token. Conjunctions consist of creating new features by grouping together features of the surrounding tokens. For instance, considering the token "regulatory" in the sentence of Figure 6.1 and a {-1,1} window, the new conjunction feature "interferon-1_&_factor1" is created. The windows {-3,-1}, {-2,-1}, {-1,0}, {-1,1} and {0,1} are used with lemmas and POS tags, which have been shown to provide positive outcomes on biomedical concept recognition [28]. On the other hand, the application of windows consists of adding selected features from surrounding tokens, following two different interpretations of neighborhood: local and dependency. Local windows add features of preceding and succeeding tokens as features of the current token. The offset positions considered are the same as those applied for conjunctions, but using token, lemma, POS and chunk features. Regarding dependency windows, the tokens are selected following the linguistic dependencies provided by dependency parsing. For instance, considering the token "regulation" in the sentence of Figure 6.1 and a maximum of 1 hop, features of the tokens "of", "-" and "in" would be used. In the end, we consider a maximum of 3 hops and take the lemma, POS and chunk features of each token in that neighborhood.

Annotation

In order to annotate hundreds of documents using multiple ML models with different feature sets, we have to avoid generating the complete feature set for each ML model. Thus, a strategy must be applied to extract all the required features at once and filter them per model. To achieve this, a model configuration that results from the union of all model configurations is built and used to extract all the required features. Afterwards, the features are filtered per model, respecting the optimized requirements of each model, and the corpus is annotated using these models. By applying this strategy we considerably reduce the complexity of annotating a corpus with multiple ML models, since extracting some complex features may take considerable amounts of time and computational resources.

Post-processing

Post-processing tasks can be performed to further optimize and/or filter the identified event triggers. Three different approaches are implemented, based on:

- Parentheses: if the number of parentheses (round, square and curly) on each annotation is an odd number, the annotation is removed since it clearly indicates a mistake by the ML model;
- Concepts: the trigger annotation is removed if the sentence does not contain any concept annotation;

Output

The output can be generated in various formats, namely JSON, XML and A1, the default, which is the official format for the BioNLP challenges. A sample output is shown in the bottom of Figure 6.1, composed of a unique identifier, the event type, start and end character positions, and the chunk of text.

6.2.4 Optimization algorithm

Since triggers for different event types have different characteristics in terms of textual context and linguistic construction, we believe that training a CRF model focused on each event type will deliver improved results in terms of accuracy and speed. Thus, the optimization algorithm aims to find the feature set and model parameters that better reflect the characteristics of each event type. The proposed method considers the following optimization targets:

- Feature set: choose the features that better reflect the linguistic characteristics of the triggers for a particular event type;
- Context: choose the technique that provides a better representation of local context;
- Model orders: choose the model order that better fits the linguistic characteristics of the triggers;
- N-grams sizes: find the n-grams size that better reflects the common sub-structures of the triggers;
- Maximum hops on dependency parsing: choose the maximum number of hops used to extract dependency parsing-based features;
- Feature extracted from vertex on dependency parsing associated features: during the construction of dependency parsing-based features, optimize the information used from each vertex;

Table 6.1 presents the pseudo-code and processing pipeline of the optimization algorithm, assuming the following notation:

- D: data set;
  - D_T: train data set;
  - D_D: development data set.
- M: model;
- MC: model configurations;
- T: trigger types;
- F: feature set;
- O: model orders;
- N: n-grams;
- FN: features that use n-grams;
- C: contexts ;
- H: dependency hops;
- FH: features that use dependency hops;
- V: vertex feature type;
- FV: features that use vertex type.

Optimization algorithm arguments (T,F,O,N,C,H) are entirely configurable, allowing users to easily customize optimization goals, workflow and complexity. Additionally, default values are assumed unless others are provided. For instance, considering the array of contexts [None, Window, Conjunctions], None is considered the default value until further optimization is performed. The same approach is applied for n-gram sizes, maximum hops and vertex features. By analyzing the "TrainModels" method, which is used on every training task, we can see that a model is trained for each order, considering the various model orders O during the entire optimization process. Regarding the "Optimization" method, which considers each trigger type from T, it starts by iteratively choosing the best feature set from F, followed by the best local context technique selection from C. Afterwards, alternative optimizations are performed, choosing the best n-grams size for each feature in FN, selecting the best maximum number of hops for each dependency parsing feature in FH, and choosing the best vertex information for each vertex-based dependency parsing feature in FV. During this process, if a feature is not used in the feature set, it is skipped from further optimization. When the optimization process finishes, the final model configurations are obtained, with optimized feature set and parameters for each event type. In the end, the final model for each event type is trained using the obtained model configuration and the complete train data set, and stored.

6.3 Results

This section presents the performance results achieved on a manually annotated corpus. A detailed comparison with other existing approaches is performed, and the annotation and

Table 6.1: Pseudo-code of the optimization algorithm.

Optimization(D, T, F, O, C, N, H, V)

1) randomly split dataset D into train D_T and development D_D datasets
2) for each trigger type $T_i \in T$
   a) for each feature type $F_j \in F$
      i) activate feature F_j on model configuration MC_i
      ii) call TrainModels with D_T, D_D, MC_i and O
      iii) if no improvement, deactivate feature F_j on model configuration MC_i
   b) for each context type $C_j \in C$
      i) activate context C_j on model configuration MC_i
      ii) call TrainModels with D_T, D_D, MC_i and O
   c) store best performing context on model configuration MC_i
   d) for each feature with n-grams $FN_j \in FN$
      i) for each n-grams $N_k \in N$
         (1) activate n-gram N_k for feature FN_j on model configuration MC_i
         (2) call TrainModels with D_T, D_D, MC_i and O
      ii) store best performing n-gram of feature FN_j on model configuration MC_i
   e) for each feature with dependency hops $FH_j \in FH$
      i) for each dependency hop $H_k \in H$
         (1) activate hop H_k for feature FH_j on model configuration MC_i
         (2) call TrainModels with D_T, D_D, MC_i and O
      ii) store best performing hop of feature FH_j on model configuration MC_i
   f) for each feature with vertex feature type $FV_j \in FV$
      i) for each vertex feature type $V_k \in V$
         (1) activate vertex type V_k for feature FV_j on model configuration MC_i
         (2) call TrainModels with D_T, D_D, MC_i and O
      ii) store best performing vertex type of feature FV_j on model configuration MC_i
3) Return MC

TrainModels (D_T, D_D, MC_i, O)

1) for each $O_j \in O$
   a) train model M on dataset D_T using MC_i
   b) get performance of model M on dataset D_D
   c) store performance and model order if better
2) return better performance and order

optimization speeds are analyzed.

6.3.1 Corpus

To provide a fair comparison of the achieved performance results in terms of event trigger recognition, we used an annotated corpus with manually annotated triggers and events. As stated before, the BioNLP challenges have highly promoted the extraction of biomedical events, especially in the recognition of gene and protein-based events. Moreover, since the training and development data sets provided in the first two BioNLP GENIA challenges (2009 and 2011) are similar, we decided to use the corpus of the BioNLP 2009 GENIA shared task since more results were available for comparison. This corpus contains manual event annotations for nine biomedical events, categorized into three different groups:

- Simple events: gene expression, transcription, protein catabolism, phosphorylation and localization;
- Binding events: binding;
- Regulation events: regulation, positive regulation and negative regulation.

The corpus contains training and development parts, which we used to train the ML models and compare final performance results, respectively. Table 6.2 presents a detailed analysis of the corpus parts and the provided manual annotations, namely proteins, events and triggers.

Table 6.2: Statistics of the training and development data sets of the BioNLP 2009 GENIA shared task: number of abstracts, sentences, annotated proteins, events and triggers.

	Train	Development
Sentences	≈7449	≈1450
Proteins	9300	2080
Events	8615	1795
Triggers	7041	1476

6.3.2 Experiment

Figure 6.3 illustrates the workflow applied to perform optimization (1), train the final models (2), and annotate the development set (3) for evaluation and comparison. Here we split the training dataset into two parts in order to train and optimize the system. Moreover, the original development dataset is used as the test dataset. The optimization algorithm was executed with the following input arguments:

- Triggers (T): [Gene_expression, Transcription, Protein_catabolism, Phosphorylation, Localization, Binding, Regulation, Positive_regulation, Negative_regulation];
- Feature set (F): all features;
- Orders (O): [1,2,3];
- Contexts (C): [None, Window, Dependency Window, Conjunctions];
- N-grams (N): [2,3,4], [2,3], [3,4];
- Hops (H): [2,3];
- Vertex types (V): [Lemma, Token, POS, Chunk].

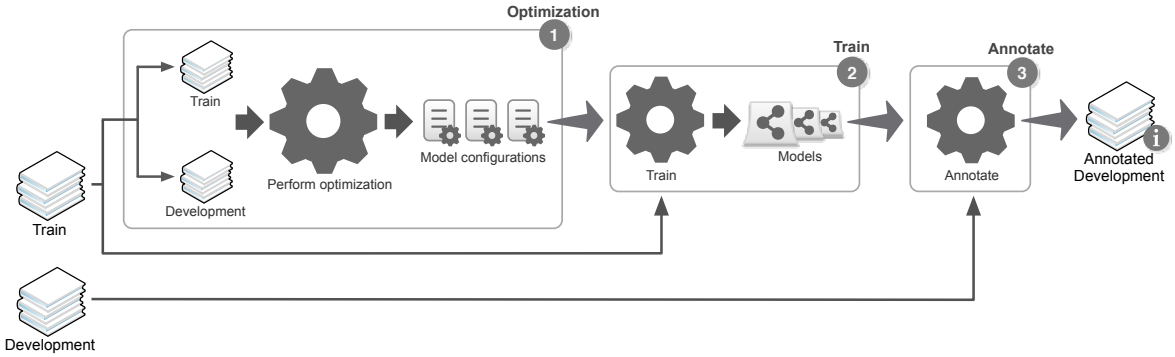


Figure 6.3: Illustration of the processing pipeline applied to perform optimization, train the final models and annotate the development corpus.

Appendix B presents the model configurations obtained after running the optimization algorithm. As can be observed, each event type requires a different feature set, reflecting the heterogeneous linguistic and context characteristics. As expected, simple events require simpler feature sets in comparison to regulatory events, whose feature sets include more token-based, concept-based and syntactic information, in order to properly model the higher complexity associated with their phrasal structure and linguistic contexts. An in-depth analysis shows that protein catabolism, phosphorylation and localization events require very simple feature sets. By contrast, the features sets to recognize gene expression, transcription and binding events require a considerable amount of context and dependency parsing information.

Overall, higher order CRF models are preferred, with seven out of nine event trigger types requiring CRFs of order three. This reflects a strong dependency on accurate sequence prediction, which we believe is directly associated with the inherent linguistic complexity of event descriptions. The low impact of local context features was unexpected, since they provide an important contribution in the case of biomedical concept recognition. However, we believe that this reduced contribution is a consequence of the deeper context description provided by dependency parsing features. Finally, we can observe that shortest path features have a much more relevant contribution than dependency path features, showing that, as expected, establishing a relation with concept names in the sentence is fundamental in the

recognition of event trigger words.

6.3.3 Results

Since more than 90% of trigger expressions are a single token, we believe that there is no need to apply fuzzy matching techniques for evaluation. Thus, only exact matching is applied, accepting an annotation as correct only if both left and right sides match. Standard evaluation metrics are used to analyze and compare the achieved results: Precision (the ability of a system to present only relevant items); Recall (the ability of a system to present all relevant items); and F-measure (the harmonic mean of precision and recall). Note that the presented results are micro-averaged, meaning that a general matrix of TP, FP and FN values is built from all documents to obtain final precision, recall and F-measure scores.

Figure 6.4 details the results of the proposed event trigger recognition method in the development set of the BioNLP 2009 GENIA shared task, and compares this with other existing systems. The data show that our approach achieves state-of-the-art results, with an F-measure of 74.5 on simple events and 52.5 on regulatory events. Overall, it achieves an F-measure of 62.7. Comparing with other existing systems, it achieves the best results on simple events, outperforming other solutions on gene expression, transcription, protein catabolism, phosphorylation and binding event triggers. Overall, our approach presents the second best results, due to the significant performance differences for regulation and negative regulation events, on which it is considerably outperformed by the best performing system. Nonetheless, the presented results are comparable to the best ones previously reported for this task and show the positive contribution of a simple automatic optimization approach.

	Our			**[238]**			**WSD [237]**						**Turku [234]**		
	CRF			CRF			CRF			CRF-VSM			SVM		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Gene expression	83.4	76.6	79.9							75.9	77.4	76.7	77.1	77.7	77.4
Transcription	77.4	60.3	67.8							64.0	61.5	62.7	66.2	66.2	66.2
Protein catabolism	95.0	100.0	97.4							100.0	84.6	91.7	94.4	89.5	91.9
Phosphorylation	86.1	77.5	81.6							82.8	70.6	76.2	77.3	85.0	81.0
Localization	76.5	65.0	70.3							72.7	61.5	66.7	85.3	75.5	78.4
Binding	71.6	58.9	64.6							78.7	52.9	63.3	68.6	53.3	60.0
EVT-TOTAL	79.8	69.8	74.5										74.8	70.0	72.3
Regulation	54.4	35.5	43.0							51.2	25.6	34.1	64.4	48.6	55.4
Positive regulation	62.7	50.9	56.2							64.9	42.2	51.2	66.5	54.1	59.7
Negative regulation	53.9	45.1	49.1							50.0	23.3	31.8	67.2	52.3	58.8
REG-TOTAL	59.5	46.9	52.5										66.3	52.7	58.7
TOTAL	**69.3**	**57.3**	**62.7**	**65.0**	**30.2**	**41.2**	**72.4**	**46.3**	**56.5**	**70.2**	**52.6**	**60.1**	**70.5**	**60.6**	**65.2**

Figure 6.4: Detailed performance results achieved by the proposed automatic approach compared with existing state-of-the-art systems.

Regarding the application of CRFs, our solution considerably outperforms previous systems, with an overall difference of more than 6 points of F-measure. This shows that CRFs are able to provide positive results in the recognition of event trigger words.

6.3.4 Speed

In order to analyze the applicability of our approach in large-scale problems, it is important to analyze the annotation processing speeds. There are various factors that add complexity to our system, namely dependency parsing, feature extraction and annotation with multiple ML models. However, the applied annotation algorithm together with multithreaded processing reduces the processing times significantly. Considering the complete processing pipeline presented on Figure 6.1 and the complexity associated with the previously obtained model configurations, the 1450 sentences of the development set of the BioNLP 2009 shared task were annotated in 40 seconds (excluding the time required to load processing models), using four processing threads running in a machine with 8 processing cores @ 2.67 GHz and 16GB of RAM. Thus, our system is able to process more than 36 sentences/second, corresponding to almost 4 abstracts/second. We believe that these results present a positive contribution, considering the inherent complexity and obtained performance results.

Regarding the optimization algorithm, this requires significant computational resources and may take a considerable amount of time, depending on the optimization algorithm configuration. In our case, which considered a high variety of complex features and parameters, the optimization process took almost 24 hours to find the best model configurations for nine event types. Thus, on average, about 2.6 hours were necessary to find the best model configuration for each event type.

6.4 Discussion

The solution presented in this chapter was built thinking on flexibility and configurability. Its architecture allows easy inclusion of new functionalities and modules, enabling easy development of new feature extraction algorithms and its integration in complex event extraction solutions. Additionally, considering the extracted linguistic information and its structured storage and access, and the amount of already implemented ML features, we believe that our solution is also a good starting point for the development of event extraction systems. Moreover, the approach and research presented in this article provides a new perspective of the linguistic and context complexity associated with each event trigger, providing a better perception of the associated requirements. This information is useful for the implementation of new event and trigger extraction solutions.

Regarding the optimization algorithm, it was developed to be completely configurable, allowing developers to easily specify the feature set, n-grams sizes, model orders and maximum dependency parsing hops. Such flexibility facilitates adapting the tool to new corpora, different domains and event triggers. Typically, the development of NER or trigger recognition solutions is performed by manually selecting the feature set and parameters that provide the best results, which is a very demanding and time-consuming task. The presented approach is

able to automatically find high-performance models in just a few hours, which we believe will save researchers' time. Since the optimization process only has to be executed once for any particular corpus, we consider the presented optimization times acceptable, in comparison with the time required to manually perform a similar process. Moreover, considering the variety of possible biomedical events, as can be seen from the new tasks emerging in the BioNLP challenges [192, 193], we can argue that the presented automatic optimization approach is an added value.

As previously shown, the automatic approach proposed here presents state-of-the-art results in the recognition of nine heterogeneous event triggers, outperforming existing solutions on simple event triggers. However, we believe there is still a margin to improve results on regulation events, which can be accomplished through the integration of new features for improved context description. By comparing the achieved performance results, we also showed that CRFs are able to perform as well as SVMs in the recognition of event triggers, considerably outperforming previous CRF-based approaches through appropriate context definition features. Additionally, our approach also presents positive annotation processing speeds, enabling its application in large-scale problems, such as annotating the entire MEDLINE.

6.5 Summary

This chapter presented TrigNER (`http://bioinformatics.ua.pt/trigner`), a new tool for biomedical event trigger recognition that takes advantage of a flexible and configurable optimization algorithm that allows the tool to adapt itself to corpora with different events and domains while maintaining high-performance results. It takes advantage of CRFs and feature sets optimized for the linguistic and context characteristics of each event type. The application of this automatic optimization algorithm delivered state-of-the-art performance results on the BioNLP 2009 shared task corpus with a total F-measure of 62.7 and outperformed existing solutions on various event trigger types, namely gene expression, transcription, protein catabolism, phosphorylation and binding.

We believe that the proposed tool represents a valuable contribution to the biomedical text mining community, by providing simplified event trigger recognition. Researchers can use it to replace or complement non-state-of-the-art dictionary-based approaches, taking advantage of a complex and high-performance solution and applying it as a simple and routine task, therefore leveraging their time to optimize and improve event argument extraction algorithms.

Chapter 7

Egas: biomedical interactive annotation

This chapter is based on:

- D. Campos, J. Lourenço, T. Nunes, R. Vitorino, P. Domingues, S. Matos, and J. L. Oliveira, "Egas-Collaborative Biomedical Annotation as a Service," in *Fourth BioCreative Challenge Evaluation Workshop*, Bethesda, Maryland, USA, Oct. 2013, pp. 254–259
- D. Campos, J. Lourenço, S. Matos, and J. L. Oliveira, "Egas: a web-based document curation platform," *Database (Oxford), Under Review*

Due to the complexity of the biomedical domain and the ambiguity of the associated scientific documents, the automatic extraction of information remains challenging, even if high-performance results have been reached in some particular tasks. For instance, in the CRAFT [52] corpus, Neji [31] achieved 95% of F-measure in the recognition of species names, and 76% of F-measure identifying gene and protein names. On the other hand, relation mining solutions present considerably inferior results, a direct consequence of the inherent task complexity. For instance, in the recognition of PPIs, the solution presented by Bui et al. [244] achieved F-measures results from 51% up to 84% in distinct corpora. When considering DDIs mining, the best solution [316] achieved 66% of F-mesure in the DDIExtraction corpus [205]. Overall, the most advanced solutions still produce many mistakes that must be taken into account when updating existing knowledge bases. Thus, one must carefully analyze the provided automatic information and correct the existing mistakes. In this perspective, various studies have shown that using automatic solutions to assist biocurators delivers improved curation times [317, 318]. Nevertheless, such solutions are still not being widely used by

biomedical research communities [319], which are the main target audience. This gap is related not only with the complexity and ambiguity of biocuration tasks, but also with the lack of standards and interaction between biocurators and developers. Moreover, Bolchini et al. [320] showed that usability of bioinformatics resources is fundamental to effectively support users in their daily research activities. Thus, it is important to develop interactive solutions that take advantage of automatic computational solutions and existing knowledge resources to assist expert curators in their daily tasks. To do so, the interface with the curator is an important aspect that needs to be considered for tool adoption. In the end, by taking advantage of such interactive solutions, biocurators can easily and more effectively keep current knowledge bases updated, and generate annotated data to develop and evaluate automatic solutions.

7.1 Background

Various research groups have developed solutions to assist biocurators, following different approaches, providing different features and targeting different tasks. Overall, two general tasks have been tackled: document triage and information annotation. Triage intends to retrieve and rank documents considering a specific goal. For instance, the BioCreative challenges organized a task [223, 224] to automatically classify documents as relevant for PPI curation. On the other hand, information annotation targets identifying information contained in documents. Many challenges were organized targeting the automatic extraction of concepts [90, 91, 95], relations [205, 223, 224] and events [192–194]. Brat [307] is one of the most used and complete web-based solutions for information curation, supporting inline annotation of documents. It provides concept normalization features, automatic services integration, search capabilities and documents comparison. However, annotation task configuration (e.g., target concepts and relations, normalization resources, and automatic services) is considerably difficult and non-accessible for non-advanced users, and document representation is considerably slow when full-text documents are used. MyMiner [321] is another complete web-based solution for biocuration, which supports concept tagging and normalization of a pre-defined set of concepts using a restrict set of previously processed resources. It also supports document triage, relation mining, automatic concept recognition, and document comparison. However, because it does not apply inline representation of annotations, understanding the inherent information may not be as clear as expected. Following a different approach, Argo [322] offers workflow design options with previously built and integrated components. Thus, users are able to create custom processing pipelines for concept and relation annotation with manual correction, supporting multiple import and export formats. Even though such approach is powerful, creating such workflows may require advanced expertise and provides a high-level

of flexibility that may not be required for biocurators. Other solutions, such as BioQRator[1], CellFinder[2], PubTator [323], RLIMS-P[3], tagtog[4] and Ontogene [324] follow typical web-based solutions with less usable interactions and annotation representation, using tabular listings of concept and/or relation annotations with simple highlighting and sorting/scoring capabilities. Nonetheless, some of those solutions incorporate interesting features. For instance, BioQRator integrates document triage for PPIs, tagtog integrates active-learning of concept names using annotated information, and PubTator features a PubMed-like interface with many state-of-the-art automatic solutions already integrated for concept recognition and normalization. There are other solutions that do not apply classic web-based approaches. For instance, SciKnowMine[5] is a desktop application for document triage that integrates active learning capabilities to obtain new models based on interactively annotated documents. On the other hand, MarkerRIF[6] is a web-browser extension that allows annotating concepts directly on documents from the Pubmed web-site, providing relevant sentences retrieval and supporting normalization of a restrict set of concepts.

Overall, in addition to the features of these tools, several desirable characteristics can be identified, that should facilitate the wider applicability and usability of this kind of tools by expert curators in their daily tasks:

- Architecture: flexible and ready to scale architecture to support new features and integrate of new services;
- Features: support for standard formats, integration with existing major services for document retrieval, integration with automatic annotation services, integration with existing state-of-the-art resources, flexible configuration of the annotation task, and real-time collaboration functionalities;
- Usability: easy to understand interfaces with inline annotations and interactions, and simple installation and configuration steps;
- Performance: fast document processing and representation.

In this chapter we present Egas, a web-based platform for interactive biomedical information curation that intends to address the aforementioned demands, delivering a highly flexible and easy to use solution. It supports manual and automatic annotation of concepts and relations, together with inline document representation and interaction. *De facto* standard knowledge bases are indexed and integrated to facilitate normalization of concept names. Real-time collaboration features are also provided to enhance curators communication and contribute to more consistent results. Moreover, Egas integrates on-demand configuration

[1] http://www.bioqrator.org
[2] http://141.20.31.85/cellfinder
[3] http://research.bioinformatics.udel.edu/rlimsp
[4] https://www.tagtog.net
[5] http://www.isi.edu/projects/sciknowmine/overview
[6] http://bws.iis.sinica.edu.tw/MarkerRIF

of the annotation task, namely annotators, concepts, relations and general annotation guidelines. Overall, based on the provided features and inherent characteristics, we strongly believe that Egas is a state-of-the-art solution to perform a large variety of biocuration tasks, ready to support information generation and keep current databases properly updated.

7.2 Methods

Egas is a web-based platform for biomedical text mining and collaborative curation. It allows users to annotate texts with occurrences of concepts and relations between these concepts. The annotation tool follows what we termed an "annotation-as-a-service" paradigm. Thus, document collections, users, configurations, annotations, back-end data storage, as well as the tools for document processing and text mining, are all managed centrally. This way, a curation team can use the service, configured according to their requisites, taking advantage of a centrally managed pipeline. Moreover, Egas was created and developed with a strong focus on usability and simplicity, applying clean and self-explanatory user interfaces and interactions. Overall, the main goal is to facilitate interactive information mining, making the tasks of data understanding and respective information extraction as simple as possible.

The tool is based on the idea of projects (Figure 7.1). A project consists of a curation task, performed by a team of curators on a collection of documents, and considering a pre-defined set of concept and relation types, as defined by the curation guidelines. The project manager is responsible for assigning users (curators) to the project for defining annotation guidelines, target concepts, relations, and project accessibility (private or public). Thus, users can only annotate a document if they are associated to the respective project. Egas keeps track of all users operations regarding annotations, namely adding, changing and removing concepts and relations. It also automatically registers curation times of each user per document, providing such statistics for further analysis.

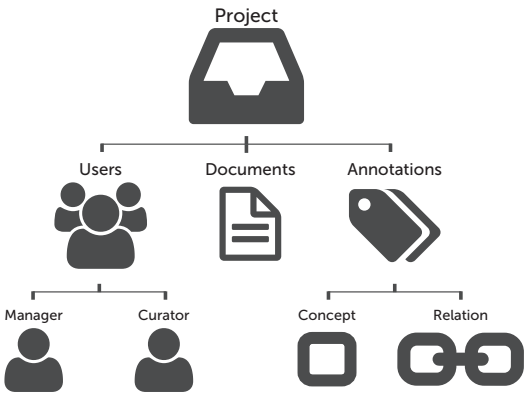


Figure 7.1: Egas organization based on projects, users, documents and annotations.

Figure 7.2 summarizes the features provided by Egas and illustrates the typical usage pipeline. At first, in order to associate a collection of documents to a project, users can import documents from their devices in standard formats, containing raw or previously annotated texts, or use remote resources to import documents, either by providing a list of identifiers, or by running remote searches on these resources. After importing documents to the project, they can be automatically annotated by using the available concept and relation annotation services. Afterwards, project administrators can freely define concept and relation types according to the requisites of the task. Additionally, each concept type can be associated to a knowledge base for normalization, and relations can be defined by specifying the types of the intervening concepts. Administrators can also upload documents describing the annotation guidelines, and specify the users that are associated with the project. After this step, curators are able to annotate the available documents by adding, editing and removing concept and relation annotations, taking advantage of real-time collaboration features for faster and easier communication. In the end of the annotation process, users are able to export annotated documents and respective concept and relation annotations to standard formats.

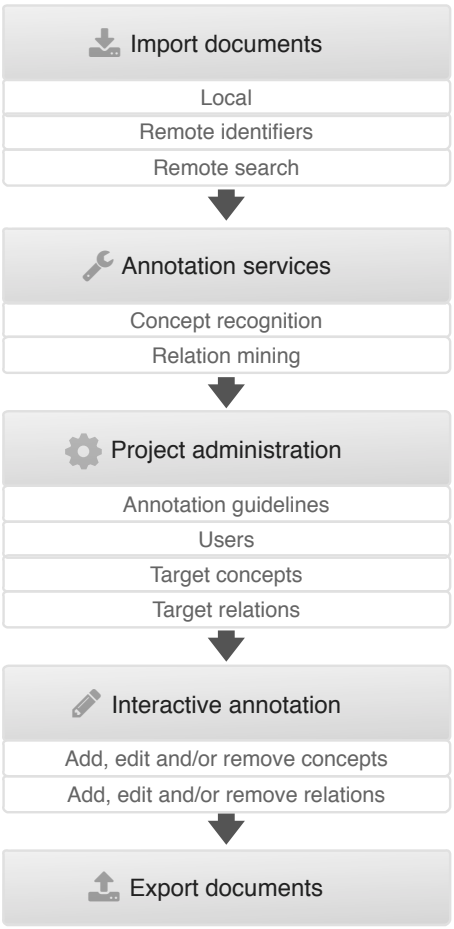


Figure 7.2: Typical usage pipeline of Egas.

7.2.1 User interface

Egas was designed to be simple and easy to use, taking advantage of user-friendly interactions highly focused on the document annotation task. Figure 7.3 presents the Egas workspace, which contains seven main action components for accessing the provided features:

1. Project management: manage and access project configurations, namely users, concepts, relations, annotation guidelines and statistics;
2. Project and document navigators: navigate through different projects and documents;
3. Processing tools: access the integrated automatic annotation services, as well as the importing and exporting functionalities;
4. Account management: manage user account settings;
5. Concept and relation type visualization filters: select concept and/or relation types to be highlighted in the document viewer.

Concepts and relations are represented inline, contributing to an improved annotation process by providing contextualized actions and rapid perception of the information added to the document. Concept annotations are highlighted with coloured boxes specific for each concept type, and to account for the complexity of the biomedical terminology, nested concept names are supported and carefully represented through overlayed boxes. On the other and, relations are displayed using directional lines below each sentence, tagged with the relation type and with boxes placed under the concepts that participate in the relation. The boxes have the same color as the respective concept, making it easy to identify the entire relation. In order to simplify the analysis of the annotated concepts and relations, users can use the corresponding visualization filters to select the concepts and relations that are shown in the document viewer. By unchecking the checkbox associated with a specific concept or relation type, the corresponding coloured boxes are removed from the document viewer, cleaning the document representation and making its analysis more focused.

Finally, as part of the workspace, it is also possible to enable real-time collaboration features. That way, Egas provides instant feedback of users' interactions within a document, such as adding, removing and/or changing concept and relation annotations. Thus, multiple users can change a document at the same time, showing exactly who changed what. A project chat is also available, which allows users to discuss details of the annotation task. Moreover, mouse pointer click position feedback is also provided, indicating where remote users clicked.

Concept and relation annotation

Information annotation is a key feature of Egas, which provides easy and interactive annotation of concepts and relations. Thus, to add a concept annotation, the user simply selects the chunk of text mentioning that concept, after which a menu is instantly shown allowing to select the concept type and the concept identifier from a knowledge base, if

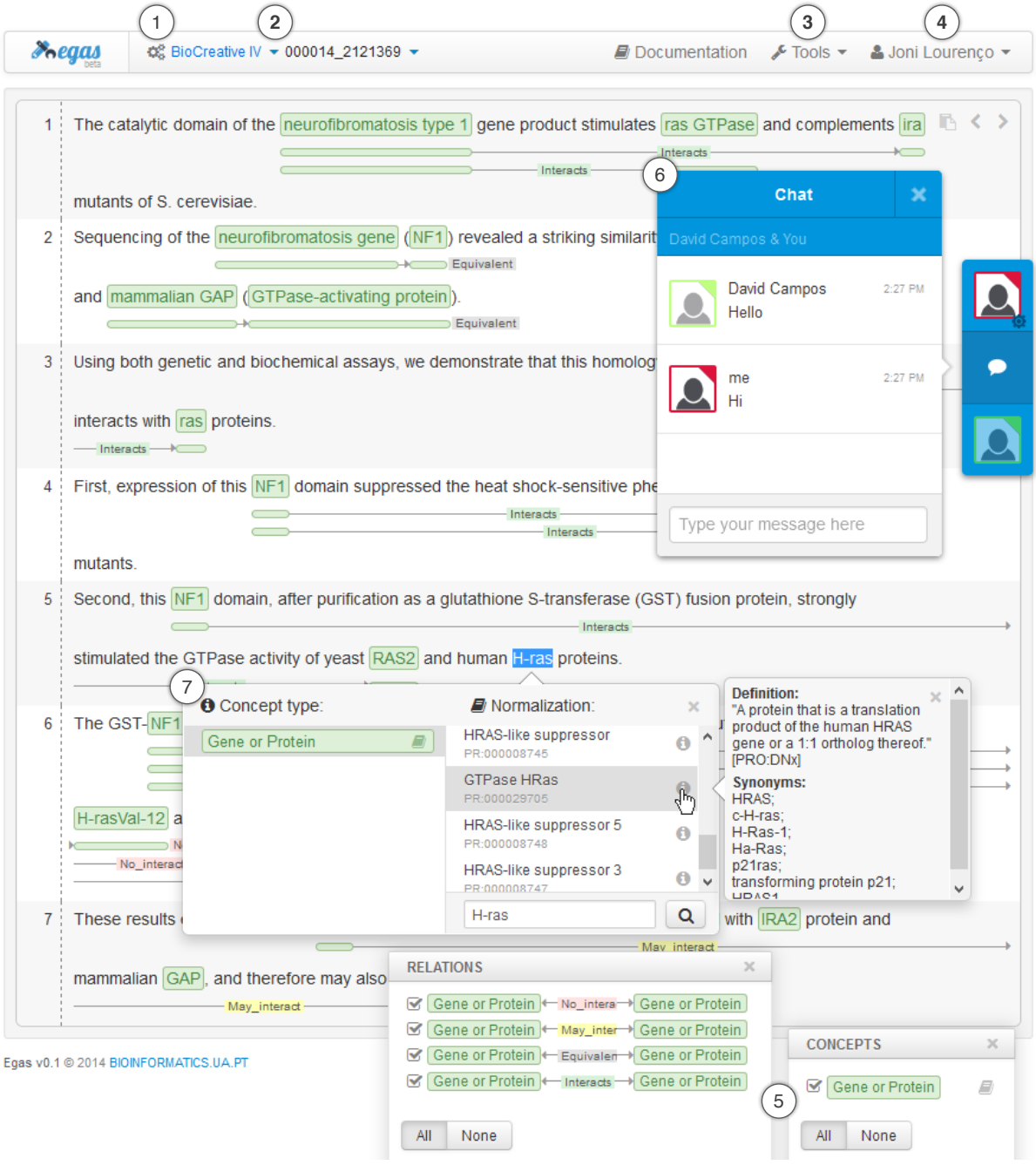


Figure 7.3: Egas main interface presenting a PubMed abstract (PMID 2121369) with annotated concepts and relations, and emphasizing relevant interaction components/features: *1)* project management; *2)* project and document navigators; *3)* processing tools; *4)* account management; *5)* concept and relation type visualization filters; *6)* real-time collaboration; and, *7)* concept annotation with normalization.

required. Adding relations is just as straightforward, simply by clicking the two concepts while pressing the "Alt" key and selecting the relation type in the pop-up menu. Right-clicking

an existing concept or relation allows removing that annotation or, in the case of relations, changing its type or direction.

Import and export documents

Import allows users to add documents to the currently selected project in three different ways. Local import allows users to select documents stored in their computer in three possible formats: raw text, A1[7] and BioC [325]. The two other options use remote servers to retrieve documents, either using lists of unique identifiers to select the documents, or by searching remote literature indexing services. Currently, both PubMed and PubMed Central are supported, allowing to import abstracts and full-text documents, respectively. User queries are executed directly in the remote services, allowing logic operators such as "AND" and "OR", as well as MeSH type queries. After submitting the query, Egas presents a list of documents, and allows the users to select the documents they want. On the other hand, export features are provided through a single interface, which allows users to select the documents to be exported and the output format. Egas currently supports two different formats: A1 and BioC.

Annotation services

The interface for calling automatic annotation services for specific documents was designed to be as flexible and adaptable as possible, in order to support services with different characteristics. Thus, Egas only requires the user to indicate the documents that should be annotated by the service. Afterwards, resulting annotations are loaded to Egas and presented in the document viewer.

Project management

Project management allows administrators to configure essential project characteristics, such as annotation guidelines, users, target concept and relation types, and access various statistics regarding the annotation process. The initial panel allows administrators to provide annotation guidelines for curators through inline text and/or attached documents in standard formats, such as Adobe PDF and Microsoft Word documents. Moreover, users management allows inviting and removing users from each project by taking advantage of an e-mail based invitation system. This panel also allows managing project administrators and pending issued invites. Besides the concept and relation types definition panels, Egas also provides a statistics panel, which allows administrators to collect detailed information regarding the annotation process per article and user, namely curation time and annotated concepts and relations. Exporting collected statistics for further analysis is also possible.

[7] http://brat.nlplab.org/standoff.html

7.2.2 Implementation

As a web-based platform, Egas intends to facilitate the access to an innovative and flexible solution for biomedical data curation, making it easily available for almost all internet-capable devices. Figure 7.4 illustrates the architecture of Egas, which is divided in two parts: client and server. The client-side is responsible for the direct interaction with users through their web-browsers, and the server-side is responsible for storing and processing all generated data. Both sides exchange data through a secured and encrypted channel using authenticated and authorized services.

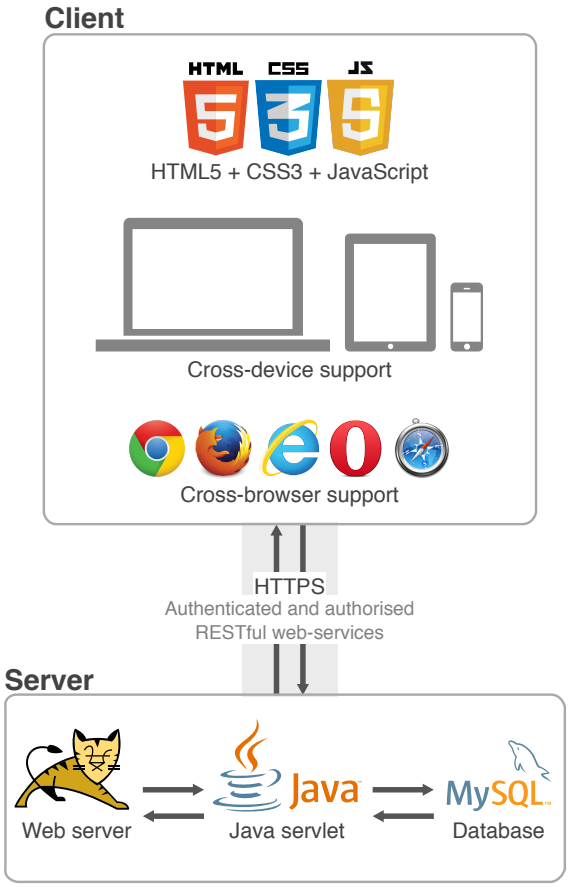


Figure 7.4: Egas architecture.

The client-side was developed targeting compatibility and performance, through the application of standard web technologies, i.e., HTML, CSS and JavaScript, which are supported by most commonly used web-browsers on both desktop and mobile devices. The application of such web standard technologies also deliver fast representation of information. Thus, together with simple and fast client-side algorithms, we enable loading and presenting full text documents with thousands of annotations in just a few seconds. For instance, considering one of the largest documents of the CRAFT corpus, which contains 3461 concept annotations,

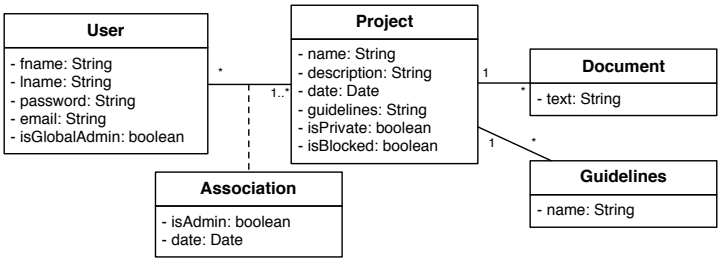
Egas only spent 3 seconds to present the document with respective annotations. On the other hand, a similar Scalable Vector Graphics (SVG) solution with inline annotations required 14 seconds to load the same exact document. Thus, our approach presents an improvement of more than 4.5 times in terms of document representation speed, which provides a smooth and sophisticated navigation and interaction with the system.

The server-side is responsible for storing all information in an unique resource, as well as providing the services to interact with that same data. All projects and respective users, documents, annotations and configurations are stored in a MySQL[8] relational database. Every processing task is available as a REST web-service, enabling easy and fast integration in any development platform, such as web, desktop and mobile. Moreover, those web-services are secured by requiring specific authentication and authorization per user. Additionally, in order to guarantee complete protection of exchanged data, the communication between client and server sides is performed through a secured and encrypted channel using Hypertext Transfer Protocol Secure (HTTPS).

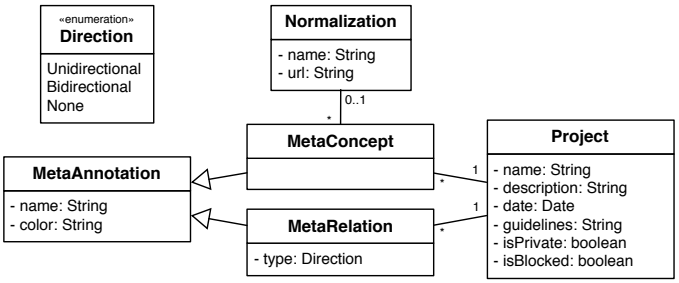
Data structure

By storing all information in a centralized and single resource, Egas enables easy setup of an annotation platform for any biocuration task. Thus, in order to store all information related with projects and users, Egas applies a data structure (Figure 7.5) designed targeting flexibility and scalability. It supports each project to contain multiple users (administrators or curators), documents with respective text, and description of annotation guidelines, provided as multiple files in attachment (Figure 7.5:a). Moreover, each project may contain multiple target concepts and relations for annotation, represented as meta concepts and relations that extend the idea of meta-annotation, which defines a specific name and representation color (Figure 7.5:b). Meta-relations have a direction type associated, which can be unidirectional, bidirectional or without any specific direction, in order to cover all possible cases. Moreover, each meta-concept may have an associated normalization knowledge base. Objectively, an annotation is an instance of a meta-annotation with specific information (Figure 7.5:c). For instance, a concept is annotated by a user in a specific document with start and end character positions, and if provided, an identifier from the normalization knowledge base. On the other hand, a relation is also annotated by a user in a document considering two target concepts. A relation can be further extended to support relations with more than two concepts. Finally, we also record the curation time of each user per document (Figure 7.5:d). This is applied by considering the time that each user spends in the Egas's tab of the web-browser. Thus, if the tab is open but the user is not working in the document annotation, that curation time is not considered.

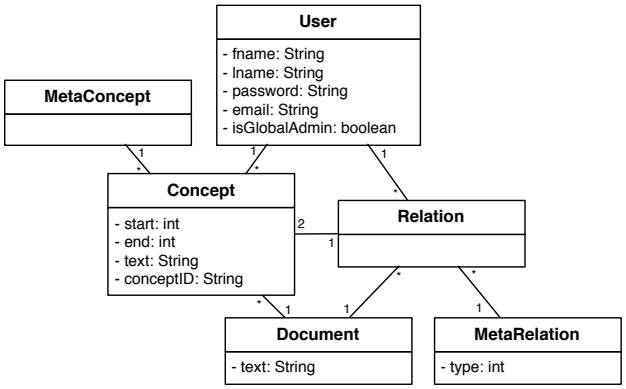
[8]http://www.mysql.com



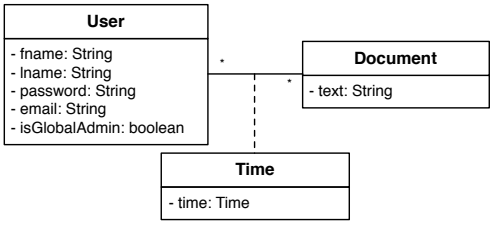
(a) **Project**



(b) **Meta-annotations**



(c) **Annotations**



(d) **Curation time**

Figure 7.5: Overview of the internal data structure to support projects and respective documents, users and annotations.

Import and export documents

As previously described, Egas supports importing documents and respective annotations (when available) from local and remote servers, as well as export features to locally store curated documents. Three formats are supported for import and export: A1, BioC and raw text. A1 converter was developed in-house, and BioC support takes advantage of the publicly available BioC Java library[9]. The integration with remote servers is performed using web-services, which already support retrieving specific documents by unique identifiers, or by submitting a search query. PubMed was integrated through the E-utility Simple Object Access Protocol (SOAP) web service [326], and PubMed Central using the Open Access (OA) REST web services[10].

Annotation services

Automatic annotation services allow performing identification of specific concepts and/or relations in a custom set of documents using state-of-the-art algorithms. That way, users can call an automatic annotation service and posteriorly manually correct the provided annotations and/or add missing ones. Such approach intends to considerably decrease the amount of time spent in the manual curation process. Egas supports automatic annotation services through an unique and simple REST web-services interface. To comply with this, web-services have to accept text as input and provide annotations following the A1 or BioC format as output. That way, it is straightforward to add new services to identify different concepts and/or relations. Moreover, Egas automatically adds concept and relation types provided by the service if they were not previously specified in the project configuration. Two different automatic annotation services for biomedical concept recognition and PPI mining and currently provided.

The concept identification service takes advantage of the BeCAS REST API [33] to provide annotations of genes and proteins, species, anatomical concepts, miRNAs, enzymes, chemicals, drugs, diseases, metabolic pathways, cellular components, biological processes and molecular functions. It was tested [31] on the CRAFT [107], AnEM [105] and NCBI Disease [308] corpora, achieving F-measure results for overlap matching of 76% for genes and proteins, 95% for species, 65% for chemicals, 83% for cellular components, 92% for cells, 63% for molecular functions and biological processes, 83% for anatomical entities, and 85% for diseases.

Regarding PPI extraction, since a state-of-the-art tool with fast processing times for real-time usage was not available as a service, we created a simple solution to provide relations between proteins and also indicate the possible presence of such relations, to support the manual annotation process. Thus, our PPIs service does not only provide relations between proteins, but also indicates the possible presence of such relations, supporting the manual

[9] http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/BioC
[10] http://www.ncbi.nlm.nih.gov/pmc/tools/oa-service

annotation process. Thus, the following annotations are provided by this service: *1)* protein concepts; *2)* relations between proteins; *3)* relations marking equivalent protein mentions (e.g., acronyms and long forms); and, *4)* trigger words that may indicate the presence of PPIs. The service was implemented on top of Neji [31], using Gimli [28] to perform ML-based protein name recognition. BioThesaurus is used to normalize recognized names, through the application of prioritized dictionary matching, as described in [31]. Equivalent protein relations are added using a simple abbreviation resolution technique, and PPIs are recognized through a rule-based approach using dependency-parsing trees. To do this, we first filter sentences by accepting only the ones that follow specific patterns, which have high probability of indicating PPIs:

- TRIGGER.*(of|between).*PRGE.*(by|to|through|with|on|and).*PRGE
- TRIGGER.*containing.*PRGE.*and.*PRGE
- PRGE.*TRIGGER.*PRGE
- PRGE.*PRGE.*TRIGGER
- TRIGGER.*TRIGGER.*between.*PRGE.*and.*PRGE
- PRGE.*TRIGGER.*TRIGGER.*with.*PRGE
- PRGE.*PRGE.*TRIGGER.*PRGE

Afterwards, considering the previously collected trigger words as reference, a relation is considered if there is a directional path between the trigger word and two proteins, allowing a maximum of four hops.

Normalization

In order to offer normalization features in the easiest and fastest way as possible for biocurators, we indexed and integrated a rich set of biomedical knowledge bases. Apache Solr[11] was used to index the identifier, preferred name, synonyms and definition (if available) of each concept in these resources. For added flexibility and robustness, a separate index is used for each knowledge base. Additionally, since knowledge bases are available in heterogeneous formats, we developed scripts to automatically index ontologies in OBO and OWL formats, and databases in SQL. Resources available in custom formats require the development of custom parsing algorithms. In order to cover the wide spectrum of biomedical knowledge, we decided to collect ontologies provided by OBO Foundry [327]. Thus, a total of 110 ontologies were indexed, including NCI thesaurus [328], NCBI taxonomy [87], Protein Ontology [85], Gene Ontology [47], ChEBI [83] and Disease Ontology [329]. Overall, more than 2 million entries are indexed and available for biocurators.

[11] http://lucene.apache.org/solr

Real-time collaboration

Real-time collaboration features were implemented by taking advantage of TogetherJS[12] from Mozilla, a JavaScript library built on top of Node.js[13] that simplifies the development of collaboration features. That way, all active users working in a document can observe the actions of adding, changing and removing concept and relations performed by other users. Additionally, every project has a dedicated chat, allowing users that are annotating different documents to discuss annotation guidelines, in order to minimize mistakes as much as possible.

7.3 Results

7.3.1 Experiment

Egas was tested in terms of applicability and user satisfaction in the BioCreative IV interactive annotation task [36], which intended to promote the development of useful text mining solutions to fill the gap between the biomedical text mining and biocuration communities, exploring the user-system interactions and hidden requirements. In that way, the task targeted the development of solutions to support interactive mining and/or triage of scientific documents.

The task organizers, together with a group of expert curators, defined a prioritized list of requirements that they considered more important to be available in such systems. The five more important system requirements were: *1)* highlighting of entities and relationships; *2)* processing of full texts; *3)* allowing manual mode for annotation; *4)* ability to edit results; and, *5)* ability to export curated results in standard formats. Each participating team developed and submitted their own approach to deal with the provided specifications. Moreover, each team had to propose a biocuration task to apply and test-drive the presented system. Our proposal consisted in the identification and extraction of biomolecular events described over PubMed abstracts related to neuropathological disorders, including PPI, protein expression and post-translational modifications. To create the corpus for this task, a collection consisting of more than 135 thousand PubMed abstracts was first obtained with the following query:

```
"Neurodegenerative Diseases"[MeSH Terms] OR "Heredodegenerative Disorders,
  Nervous System"[MeSH Terms] AND hasabstract[text] AND English[lang].
```

The documents were then ranked according to their relevance for extracting protein-protein interactions, using a SVM classifier [330] trained on the BioCreative III PPI Article Classification Task data [224]. Such approach achieved an F-measure of 62% and an Accuracy of 88%, when tested on the test part of the data. Finally, the top-ranked 100 documents were selected for the task.

[12] https://togetherjs.com
[13] http://nodejs.org

Four curators were selected, and each was assigned 50 documents from the corpus to curate. Curators were asked to annotate 25 of their assigned documents using the available PPI annotation service described above, and the remaining 25 documents without using this service, in order to assess its impact on curation effort. In the first case, curators had to revise the automatically generated annotations, correcting any erroneous concept or relation annotations and adding missing ones. In the second case, curators had to annotate all mentions of protein names and all protein interactions described in each document. The tool recorded the time taken by each curator to curate each document, as well as the number of annotated concepts and relations.

7.3.2 Results

Nine systems participated in the BioCreative IV IAT, targeting heterogeneous domains of application, and differing significantly in the followed approaches, in terms of design, implementation and usability. Overall, four systems provided integrated triage features, eight systems supported concept recognition (five of those with normalization), and six enabled relation/event mining.

To properly evaluate the behavior of the various systems, the BioCreative IV IAT organization committee built a detailed survey to subjectively rank and compare the different tools. Such survey covers various aspects of curators' satisfaction, such as: *1)* overall reaction; *2)* comparison with similar systems; *3)* ability to complete tasks; *4)* design; *5)* learning to use the application; and, *6)* usability. The answers to each of the 23 questions were scaled from 1 (very bad) to 5 (very good). The obtained evaluation results were averaged and grouped in three categories: recommendation, rating and experience. Egas presented very satisfying results in the three categories from the four curators, obtaining an average of 4.5 points in recommendation and 4.75 points in rating and experience.

Regarding the impact of automatic text mining services, the application of these annotation algorithms significantly contributed to reduced curation times: for 3 of the 4 curators, the curation times were reduced by 1.5 to 4 times. However, we also observed that automatic services may contribute to biased annotations, since curators tend to be influenced by automatic annotations, accepting or performing slight changes without throughout analysis and reflection. Thus, automatic tools should follow the same standards and assumptions as defined by the annotation guidelines, a fact that must be carefully considered in any annotation task. For instance, if the automatic tool provides species names as part of protein names, and the annotation guidelines indicate otherwise, the final corpus can be easily inconsistent and with serious annotation mistakes, seriously degrading the final IAA.

7.4 Discussion

We believe that Egas presents various advantages for biocurators, in terms of usability and simplicity. These advantages are an added value for the biomedical community, contributing to a faster and more accurate annotation of biomedical information from scientific literature. Thus, we discuss the contributions of delivering a platform-as-a-service solution and of integrating real-time collaboration features.

7.4.1 Biocuration-as-a-service

Following an application service solution, Egas enables on-demand creation and configuration of annotation projects, allowing supervisors to independently define target concepts and relations, invite curators and define annotation guidelines. Moreover, during the annotation process, supervisors can change any of the settings on-demand, obviously respecting consistency requirements. For instance, a user cannot delete a concept type if a relation type is using it. Additionally, the statistics dashboard allows administrators to actively supervise the performed work, providing valuable information regarding curation time and the amount of concepts and relations per article and user. Such active management and supervision is only possible by taking advantage of the integrated annotation task management features, which we believe is an added value for biocurators.

Egas also facilitates concept normalization by integrating and indexing a complete set of knowledge bases, offering heterogenous information targeting different domains of interest. That way, the presented platform positively responds to the needs of the most different curation tasks. The integration of such resources considerably facilitates biocurators tasks, since they do not have to acquire a deep understanding of knowledge bases, and/or develop any kind of scripts to process and integrate them. Thus, users can take advantage of such ontologies by simply associating a concept type with a specific normalization resource.

Finally, Egas also integrates annotation services to provide automatic identification of concepts and relations. As previously discussed, such integration may contribute to improved curation speeds, resulting in more time available to annotate more documents. Since the interaction with different automatic annotation services is performed through a single and self-explanatory interface, biocurators do not need any kind of expertise to take advantage of such advanced technologies. Overall, this simple integration of annotation services allows biocurators to easily take advantage of high-end and advanced biomedical text mining solutions, an approach that may streamline the communication and collaboration between text mining and biocuration communities.

By delivering a platform-as-a-service, Egas significantly facilitates the setup and on-demand configuration of annotation tasks. Additionally, since many curation tasks may work with sensitive data, we considered security as one of the most important characteristics of our

system. That way, all communications between clients and the server is performed through secured channels, using HTTPS. Moreover, all actions that interact with the centralized database, are carefully authorized and authenticated, considering the user permissions.

7.4.2 Real-time collaboration

In a classic annotation process, those responsible for the task start by specifying a target domain and by defining the annotation guidelines, where they describe target concepts, relations, and present examples of what to annotate. Afterwards, each curator has access to the annotation guidelines, interprets them, and starts annotating the set of documents that he/she was assigned to. During this process, frequent discussions among annotators to resolve and document ambiguous cases, and repeated verification of the annotated data against the guidelines are performed, in order to ensure annotation quality. In the end, IAA may be calculated to obtain a feedback regarding generated information consistency among curators. However, some research works [208] focused their efforts on annotating more documents with high quality, guaranteed by active supervision and correction of mistakes, rather than annotating repeated documents to obtain IAA scores. Based on this, we strongly believe that the definition of annotation guidelines, and the active discussion and iterative correction of annotations and respective guidelines is one of the most important aspects of the annotation process. Thus, through Egas, annotation task supervisors around the globe can work together to define the first version of annotation guidelines, taking advantage of the real-time feedback of concept and relation annotations, and of the chat to discuss mistakes and ideas. Additionally, since annotation guidelines are integrated in the platform, all supervisors can contribute to their improvement, and all participants have access to the most updated version. After starting the annotation process, curators can use real-time collaboration features to discuss with each other the interpretation of annotation guidelines using the chat, and supervisors can observe curators' work, correcting and discussing mistakes, and possibly improving the guidelines. In conclusion, through real-time features, we intend to promote the active involvement of both supervisors and curators in the annotation process, in order to deliver improved information consistency and quality.

7.5 Summary

This chapter presented Egas (`http://bioinformatics.ua.pt/egas`), a complete platform for scientific literature curation, focused on usability, simplicity, security and integration. It offers highly usable interfaces for manual and automatic in-line annotation of concepts and relations. A comprehensive set of knowledge bases are integrated and indexed to provide straightforward concept normalization features. Moreover, real-time collaboration and conversation functionalities allow discussing details of the annotation task as well as provid-

ing instant feedback of curators interactions. Egas also provides interfaces for on-demand management of the annotation task settings and guidelines, and supports standard formats and literature services to import and export documents. With Egas, we participated in the BioCreative IV interactive annotation task, targeting the assisted identification of protein-protein interactions described in PubMed abstracts related to neuropathological disorders. When evaluated by expert curators, it presented very good results regarding usability, reliability and performance. The application of automatic annotation services presented considerably reduced curation times. Moreover, Egas showed superior document processing and representation speeds, which is a significant added value and contribution to a smoother annotation process. Overall, Egas presents various advantages for the biomedical community, streamlining the collaboration between supervisors and curators, and simplifying the setup and on-demand configuration of the annotation task, using integrated knowledge bases and automatic annotation services. These contributions, together with the presented results, show that Egas is a state-of-the-art solution to perform a large variety of biocuration tasks, ready to grow and to be integrated with any major platform to support information generation and keep current databases properly updated in a consistent way.

Chapter 8

Conclusion and future work

This chapter presents some concluding remarks and research directions stemming from the research described in this thesis.

8.1 Conclusion

The goal of the research work presented in this thesis was to study, design and develop innovative solutions to support automatic mining of biomedical information from scientific literature, being mostly focused on the tasks of concept recognition (NER and normalization) and interactive curation, with contributions on event and relation mining. This work started with a careful study and analysis of the target research field, presented on Chapter 2, describing the motivation for the biomedical information extraction field, and detailing the tasks, respective approaches and existing solutions. Through this analysis we were able to carefully define and build a personal understanding and insight regarding previous work and possible research points of interest and future steps.

In Chapter 3 we present Gimli, our first and one of the most important contributions. Gimli is a machine learning-based solution for biomedical named entity recognition, which uses CRFs with a rich set of features, combining annotations of heterogenous models with a simple confidence-based harmonization technique. Moreover, two post-processing methods are applied to improve annotations quality. The recognition of different biomedical concepts is performed using different CRF models, whose feature set is optimized through an incremental approach. Such technique allowed an in-depth analysis of the best features required to recognize different concept types, providing a better understanding of their linguistic and complexity characteristics. Gimli was applied in two corpora to identify gene/protein, DNA, RNA, cell type and cell line concept names. It outperformed previously available open source tools. Gimli was further applied in the recognition of chemical compound and drug names [29], also achieving encouraging and positive performance results. Gimli is open source and publicly available at `http://bioinformatics.ua.pt/gimli` with detailed documentation for

users and developers.

In order to develop a more complex, complete, knowledge-based and accurate NER solution, we tackled the task of named entity harmonization. Chapter 4 presents Totum, a ML-based solution to harmonize gene and protein names provided by multiple heterogenous NER systems. It uses CRFs to combine annotations with different characteristics provided by four systems applying different recognition techniques, and using four corpora with different characteristics to perform training and evaluation. By doing so, Totum delivers an innovative cross-corpus solution to perform gene and protein names harmonization, which is not constrained to a specific corpus as previous systems. Moreover, the provided annotations take advantage of a rich knowledge base, creating unique and reasoned guidelines that respect as much as possible the heterogeneity of the various corpora. When evaluated on each corpus and on a merged corpus, Totum delivered significant improvements in comparison with state-of-the-art solutions. Moreover, we emphasized the differences between the various corpora and respective annotation guidelines.

Chapter 5 presents Neji, a modular framework to support the development of concept recognition solutions. Neji is specifically optimized for the biomedical domain, integrating dedicated and optimized modules and supported standards, and delivering high modularity together with fast processing speeds and high performance results. The extracted information is stored in an innovative concept tree, supporting structured ambiguity and multiple identifiers per concept. It also integrates a CLI annotation tool, which allows users to easily perform offline annotation of large amounts of documents with custom dictionaries and ML models with normalization dictionaries. The reliability of Neji was confirmed by annotating three different corpora with a total of nine concept types, outperforming existing solutions on heterogenous concept recognition. Neji is open source and publicly available at `http://bioinformatics.ua.pt/neji` with detailed documentation for users and developers. Chapter 5 also presents BeCAS, a web application, set of web-services and widget that takes advantage of Neji to deliver on-demand biomedical concept identification. Concept recognition features are provided through web-services, supporting selective annotation of eleven biomedical concepts. The web application applies intuitive annotations visualization and filtering interfaces, supporting inline nested concept names and providing link-outs to reference curated databases. Finally, its widget version allows easy integration of BeCAS features in any web page. BeCAS is available at `http://bioinformatics.ua.pt/becas` with detailed documentation for users and developers.

Taking advantage of the knowledge assimilated during the development of Gimli, and using the flexibility and speed delivered by Neji, we decided to tackle a challenging and more linguistic processing and knowledge intensive task: trigger recognition for biomedical event mining. Chapter 6 presents TrigNER, a ML-based solution to perform automatic and optimized recognition of biomedical event triggers. It applies CRFs with a rich feature-set

and post-processing modules, applying an innovative and automatic optimization method to obtain the best feature-set and model parameters for each event trigger, removing the hard task of manually optimizing the models characteristics. Such technique also allows to easily identify the complexity and linguistic characteristics of different triggers. When evaluated against manually annotated corpora, our solution outperformed existing solutions in the extraction of various gene-centric event triggers. TrigNER is open source and publicly available at `http://bioinformatics.ua.pt/trigner` with detailed documentation for users.

Finally, we tackled the task of interactive mining to take advantage of the previously developed state-of-the-art solutions and reduce the gap between the biomedical text mining and biocuration communities. Egas, presented in Chapter 7, is an innovative web-based platform for biomedical collaborative curation as a service, supporting manual and automatic annotation of concepts and relations. The user interface was developed targeting simplicity and intuitive interactions, through inline document visualization, filtering, insertion and deletion of annotations and relations. Moreover, it provides a rich set of features to support the complete workflow of knowledge curation, such as integrated project management, possibility to import and export documents to/from local and remote servers, automatic and state-of-the-art annotation services, and innovative real-time collaboration. Egas was developed on top of standard web technologies, in order to enable fast processing and visualization of documents in modern web-browsers. When evaluated by expert annotators, Egas obtained the best results in terms of usability, reliability and performance. Egas is available at `http://bioinformatics.ua.pt/egas` with detailed usage documentation.

Overall, this research work contributed with novel methods, applications, frameworks and libraries for the biomedical text mining community, helping simplifying complex processing steps through optimized solutions. Moreover, by promoting the application of biomedical information extraction methods on scientific literature, the work described on this thesis further contributes to: *a*) keeping current knowledge bases updated; and *b*) generating new hypothesis towards knowledge discovery.

8.2 Future work

Besides the fact that the performed research already incorporates multiple improvements on existing methods and solutions for biomedical information extraction, some aspects can be further explored. Overall, the research work conducted may be continued in terms of: *a*) applied techniques; *b*) performed tasks; and *c*) conceptual application.

Regarding the applied techniques, various methods can be explored and investigated to further improve systems' performance and behavior:

- Feature induction [331]: automatically extract informative features from texts in order to improve the feature set and obtain "hidden" characteristics of tokens and textual

context. For instance, McDonald and Pereira [138] applied a simple feature induction approach in the recognition of gene and protein names, achieving significant improvements of 1.3% of F-measure;

- Semi-supervised learning: use both annotated and unannotated data in order to extract characteristics of the unlabeled data that could contribute to a better identification of concepts names and/or relations. For instance, Ando [161] applied semi-supervised learning and achieved improvements of more than 2% of F-measure in the recognition of gene and protein names;
- Joint learning: take into account the interdependencies between candidate extractions during the learning process. Various IE tasks are trained together in the same corpus and focused on the same task, optimizing the various solutions and their dependencies considering an unique goal. For instance, Finkel and Manning [332] performed joint dependency parsing and NER, achieving improvements up to 1.4% and 9.0%, respectively;
- Improved knowledge input: provide more and improved domain knowledge as input of ML models and systems will further improve their performance, making the decisions even more reasoned and consistent. For instance, by building a graph representation of concepts and their interactions based on integrated heterogenous knowledge bases will provide an important input to any biomedical IE task, such as relation mining [273] and disambiguation [187];
- Improved linguistic parsing: improved linguistic information may further contribute to better biomedical IE, since dependency parsing solutions still need to be improved in terms of both accuracy and speed. The application of some relation and event mining solutions that strongly rely on linguistic parsing information is still far from real-time processing. Moreover, applying just chunking information to filter and/or extract relations between concepts [244] can also be a path to follow.

Besides the further investigation of previously described techniques, there are also various directions that can improve the developed tools in terms of usability and accessibility:

- Web services: make developed solutions available as simple and well documented web-services to disseminate their usage and integration in complex text mining workflows;
- Libraries and frameworks: provide developed solutions as development libraries and/or frameworks to simplify the integration in complex solutions, and to support the development of custom solutions;
- Web and desktop applications: the development of easy to use graphical interfaces will make such tools easily accessible for final users;
- Mobile applications: with the proliferation of mobile devices, such as tablets and smartphones, it is essential to follow that trend and deliver solutions compatible with such usage patterns, satisfying the users' needs and requirements.

Even though we already follow such directions in most of the developed tools, we strongly believe that this is an important requirement to enable the application of such solutions by biocuration experts, giving an important contribution to minimize the gap between curators and complex computerized solutions.

8.2.1 Research directions

By taking advantage of the developed solutions and acquired know-how, the work presented in this thesis serves as a baseline to further explore other biomedical information extraction tasks, studying innovative techniques that we believe will deliver improved results:

- Knowledge-based disambiguation: we consider disambiguation to be an important next step in this work, due to the complexity of the biomedical domain and to the levels of ambiguity that we already observed in our experiments. Thus, in order to build a general approach, not strictly optimized to specific terms, we intend to contribute to the development of improved knowledge-based disambiguation approaches, which still have to reach the levels of accuracy achieved by ML-based solutions;
- Machine learning and knowledge-based relation and event mining: we intend to explore the tasks of relation and event mining by applying hybrid solutions taking advantage of both ML and knowledge bases. The main idea is to train ML models using features from a knowledge base and/or applying post-processing techniques using the same knowledge base;
- Improved interactive mining: we plan to further explore and improve our solution for interactive mining, minimizing the gap between scientific articles and knowledge bases;
- Semantic web, indexing and searching: we intend to work on semantic indexing and search, in order to deliver the extracted information to final users in the best way as possible and considering the meaning, relations and events of "things". Moreover, we also have to work on semantic integration technologies to associate the extracted information with other heterogeneous resources;
- Question answering: as the Holy Grail of the field, we plan to explore solutions to retrieve and synthesize relevant information from both textual and structured data, in order to provide real-time and reasoned answers to natural language questions.

Finally, the developed solutions may be applied and/or adapted to different domains, taking advantage of the acquired know-how and exploring new areas of application, such as:

- Domains: apply the developed solutions in different domains in order to integrate new research projects with closed defined and focused goals. There are many areas that may take advantage of TM solutions to improve their research, achieving better and faster results. We are currently applying our solutions in a project related with neurodegenerative diseases;

- Documents: explore different types of documents, such as clinical records and patents, which present different challenges and goals. For instance, clinical records contain a large amount of acronyms and specific terminology that is difficult to map with existing knowledge bases;
- Concepts: consider the recognition and normalization of different types of biomedical concepts, such as chemicals, drug dosage administration, and gene variations;
- Relations: investigate the extraction of different types of biomedical relations to collect new information that may deliver new hypothesis and conclusions. For instance, gene-drug and chemical-protein relations still have to be properly explored, which deliver important information regarding pharmacogenetics;
- Events: explore the identification of biomedical events to help understanding general and specific biological processes and molecular functions. There is a current interest in extracting events associated with cancer genetics and general pathway curation;
- Challenges: keep participating in domain challenges which typically define new tasks and provide innovative and useful information to go beyond existing solutions, allowing a fair comparison of the achieved performed results with different approaches.

Overall, we strongly believe that there is much work to do in biomedical information extraction, which is a continuation of the research presented in this thesis. The world just started its journey exploring biomedical information from unstructured data sources, in order to create an unique resource that reflects human knowledge and support new discoveries. The future is promising and interesting for any biomedical researcher and text miner, there is much to explore, learn and discover.

Appendix A

Detailed results of biomedical named entity harmonization

This appendix presents detailed results of biomedical named entity harmonization, considering Precision (P), Recall (R) and F-measure (F1) results of exact, cosine 0.98, cosine 0.90 and nested matching alignments of the annotations provided by the four systems and the four hamonization approaches. The shaded boxes (73.87%) highlight the F-measure of each system, where the bold font (**86.24%**) indicates the best system of the partners and harmonisation solutions. The arrow (⇧) indicates the harmonisation solution with highest precision, and the circle (●) the one with better recall.

			FSUPRGE				JNLPBA				PENNBIOIE			
			Exact	Cos98	Cos90	Nested	Exact	Cos98	Cos90	Nested	Exact	Cos98	Cos90	Nested
Systems	System 1	R	52.74%	56.34%	57.02%	53.19%	32.53%	43.15%	44.04%	32.64%	52.08%	55.89%	56.78%	52.93%
		P	71.40%	76.27%	77.19%	72.00%	42.78%	56.75%	57.92%	42.93%	59.36%	63.70%	64.71%	60.33%
		F1	**60.67%**	**64.80%**	**65.59%**	61.18%	36.96%	49.02%	50.04%	37.09%	55.48%	59.54%	60.48%	56.39%
	System 2	R	36.85%	38.62%	38.79%	37.68%	23.85%	30.35%	30.63%	24.06%	33.48%	35.11%	35.35%	35.20%
		P	83.72%	87.75%	88.14%	85.60%	50.53%	64.30%	64.88%	50.98%	83.88%	87.96%	88.57%	88.19%
		F1	51.17%	53.64%	53.87%	52.32%	32.41%	41.23%	41.61%	32.70%	47.86%	50.18%	50.53%	50.32%
	System 3	R	45.82%	51.75%	52.41%	54.40%	44.64%	50.88%	51.07%	45.52%	53.67%	58.27%	59.03%	59.90%
		P	76.51%	86.41%	87.50%	90.84%	59.62%	67.95%	68.21%	60.80%	84.76%	92.02%	93.22%	94.59%
		F1	57.32%	64.74%	65.55%	**68.05%**	**51.06%**	**58.19%**	**58.41%**	**52.06%**	65.72%	71.36%	72.28%	**73.35%**
	System 4	R	43.90%	45.71%	45.97%	44.16%	35.17%	44.97%	45.64%	35.23%	57.38%	60.88%	61.32%	60.98%
		P	83.41%	86.85%	87.34%	83.90%	50.60%	64.70%	65.66%	50.69%	82.77%	87.82%	88.45%	87.96%
		F1	57.53%	59.90%	60.23%	57.86%	41.49%	53.06%	53.85%	41.57%	**67.78%**	**71.91%**	**72.43%**	72.02%
Harmonisation	Union	R	66.49%	71.59%	72.46%	71.15%	48.26%	62.89%	63.55%	48.70%	72.37%	77.52%	78.65%	77.52%
		P	69.14%	74.44%	75.34%	73.98%	45.97%	59.92%	60.54%	46.39%	70.04%	75.02%	76.12%	75.02%
		F1	67.79%	72.99%	73.87%	72.54%	47.09%	61.37%	62.01%	47.52%	71.18%	76.25%	77.36%	76.25%
	Intersection	R	53.52%	56.90%	57.20%	53.93%	38.02%	50.05%	50.52%	38.11%	62.32%	66.16%	66.43%	63.19%
		P	⇧83.70%	⇧88.99%	⇧89.45%	84.33%	49.45%	65.10%	65.71%	49.58%	⇧86.54%	⇧91.88%	⇧92.25%	87.75%
		F1	65.29%	69.41%	69.78%	65.79%	42.99%	56.59%	57.12%	43.10%	72.46%	76.93%	77.24%	73.47%
	TotumID	R	66.50%	70.95%	71.50%	70.93%	51.81%	62.08%	62.80%	52.33%	71.01%	74.34%	74.92%	77.01%
		P	78.87%	84.14%	84.79%	84.12%	58.75%	70.40%	71.21%	59.34%	83.32%	87.23%	87.92%	90.36%
		F1	72.16%	76.98%	77.58%	76.96%	55.06%	65.98%	66.74%	55.62%	76.67%	80.27%	80.90%	83.15%
	Totum	R	●68.15%	●76.26%	●77.09%	●83.09%	● 69.29%	● 76.16%	● 76.93%	● 72.61%	● 74.03%	● 79.41%	● 80.20%	● 85.71%
		P	73.51%	82.26%	83.15%	⇧89.63%	⇧ 65.81%	⇧ 72.34%	⇧ 73.06%	⇧ 68.97%	81.21%	87.11%	87.98%	⇧ 94.02%
		F1	**70.73%**	**79.15%**	**80.00%**	**86.24%**	**67.51%**	**74.20%**	**74.95%**	**70.74%**	**77.46%**	**83.08%**	**83.91%**	**89.67%**

			GENETAG				MERGED			
			Exact	Cos98	Cos90	Nested	Exact	Cos98	Cos90	Nested
Systems	System 1	R	35.88%	47.22%	49.00%	36.47%	46.21%	52.39%	53.32%	46.69%
		P	50.82%	66.88%	69.41%	51.66%	60.93%	69.07%	70.30%	61.56%
		F1	42.06%	55.35%	57.45%	42.76%	52.56%	59.58%	60.65%	53.10%
	System 2	R	22.94%	30.03%	30.91%	24.20%	31.69%	35.17%	35.49%	32.62%
		P	60.36%	79.01%	81.31%	63.67%	73.78%	81.88%	82.62%	75.93%
		F1	33.25%	43.52%	44.79%	35.07%	44.33%	49.21%	49.65%	45.63%
	System 3	R	38.92%	43.19%	43.70%	41.70%	45.67%	51.18%	51.74%	51.55%
		P	81.61%	90.57%	91.63%	87.45%	74.79%	83.80%	84.72%	84.40%
		F1	**52.71%**	**58.49%**	**59.18%**	**56.48%**	**56.71%**	**63.54%**	**64.24%**	**64.00%**
	System 4	R	26.04%	35.07%	36.28%	29.05%	41.44%	46.14%	46.66%	42.64%
		P	55.38%	74.58%	77.15%	61.78%	72.34%	80.55%	81.45%	74.42%
		F1	35.42%	47.71%	49.35%	39.52%	52.69%	58.67%	59.33%	54.21%
Harmonisation	Union	R	48.27%	63.53%	64.81%	52.29%	61.10%	69.60%	70.54%	64.97%
		P	52.70%	69.37%	70.76%	57.10%	62.29%	70.96%	71.92%	66.24%
		F1	50.39%	66.32%	67.65%	54.59%	61.69%	70.28%	71.22%	65.60%
	Intersection	R	34.65%	45.24%	46.39%	35.32%	48.96%	55.16%	55.62%	49.42%
		P	62.56%	⇧81.67%	⇧83.76%	63.76%	⇧ 74.09%	⇧ 83.48%	⇧ 84.18%	74.80%
		F1	44.60%	58.23%	59.71%	45.46%	58.96%	66.43%	66.99%	59.52%
	TotumID	R	43.17%	55.99%	57.60%	50.75%	59.42%	66.23%	67.04%	64.16%
		P	57.76%	74.90%	77.05%	67.90%	73.18%	81.58%	82.57%	79.03%
		F1	49.41%	64.08%	65.92%	58.09%	65.58%	73.11%	74.00%	70.83%
	Totum	R	●53.76%	●63.58%	●64.93%	●66.71%	● 66.86%	● 74.62%	● 75.51%	● 78.89%
		P	⇧65.45%	77.40%	79.04%	⇧81.21%	71.93%	80.27%	81.23% ⇧	84.87%
		F1	**59.03%**	**69.81%**	**71.29%**	**73.25%**	**69.30%**	**77.34%**	**78.27%**	**81.77%**

Appendix B

Biomedical event trigger recognition feature sets

This appendix presents a detailed description of the feature sets obtained after running the automatic optimization algorithm in the training data of event trigger recognition. Configurations presented as "Lemma, [2,3,4], 3", indicate the applicability of [2,3,4] n-grams to combine lemmas of each vertex until a maximum number of 3 dependency hops.

		Gene expression	Transcription	Protein catabolism	Phosphorylation	Localization	Binding	Regulation	Positive regulation	Negative regulation
Basic linguistic	Token	X	X	X	X	X	X	X	X	X
	Lemma	X	X	X	X		X	X	X	
	POS		X		X					
	Chunk		X							X
Orthographic	Capitalization		X		X	X				
	Counting							X	X	
	Symbols							X		
Morphological	Char n-grams	X	X	X		X	X	X	X	X
		[2,3]	[2,3,4]	[2,3,4]		[2,3,4]	[2,3,4]	[3,4]	[2,3,4]	[2,3,4]
	Suffix								X	
									[2,3,4]	
	Prefix	X					X			X
		[2,3,4]					[2,3,4]			[2,3]
	Word shape						X			
Concepts	Tags								X	
	Names								X	
	Heads					X				X
	Counting	X	X	X	X	X	X		X	X
Sentences	Tokens counting						X		X	X
External resources	Triggers	X	X				X		X	
Dependency path	Modifiers		X				X		X	
	In and out dependencies		X							
			3							
	Vertex path									
	Edge path		X						X	X
			3						3	2
	Edge type									
	Vertex n-grams	X								
		Lemma, [2,3,4], 3								
	Edge n-grams									
Shortest path	Distance	X	X				X	X	X	X
	Vertex path		X				X			X
			Word				Lemma			Word
	Edge path	X	X					X		X
	Edge type	X						X		
	Vertex n-grams	X	X					X	X	
		Lemma, [2,3,4]	Lemma,[2,3,4]					Lemma,[3,4]	Lemma,[2,3,4]	
	Edge n-grams	X	X			X		X		X
		[2,3,4]	[2,3,4]			[2,3,4]		[2,3,4]		[2,3,4]
Context	Window	X								
	Dependency window							X		
	Conjunctions									
Model	Order	3	3	1	3	3	3	2	3	3
# Occurrences		1379	482	99	140	208	673	716	2149	761

Appendix C

Annotation guidelines of protein-protein interactions in neurodegenerative diseases

This task requires the annotation of protein-protein interactions (PPI) in a corpus of abstracts related to neurodegenerative diseases. A PPI is defined as any mention of a direct (physical) interaction between two proteins, as well as when a protein changes or regulates another protein's physical/chemical/dynamical properties or function. In terms of annotation guidelines, no distinction is made between these types of interactions. Three annotation tags are used: *a*) "Interacts", meaning that a protein is referred to definitely interact with another protein; *b*) "May_interact", meaning that the text points to an interaction but is not conclusive; *c*) "No_interaction", meaning that the text mentions that an interaction was not present. Note that the second case only applies to speculative mentions in the text (e.g. as suggested by words such as "may", "indicating", "apparent") and not to cases where the curator is unsure about an interaction being mentioned or not in a given sentence. In the latter case, do not provide an annotation. A fourth annotation tag ("Equivalent") is used to annotate mentions of synonyms of the same protein within a sentence. An example is when a protein's long name is mentioned, followed by its symbol.

C.1 What to annotate?

1. Mentions of PPIs described within a single sentence;
2. Conclusive mentions of PPIs. These should be marked with the "Interacts" relation;
3. Mentions of "speculative" PPIs (e.g., "may" and "appears to"). These should be marked with the "Possible_interaction" relation;
4. Mentions of negated PPIs (e.g., "did not affect"). These should be marked with the

"Negated_interaction" relation;

5. If a protein is mentioned by its long name and the respective symbol (e.g., "Activator protein 2 (AP2)"), annotate both forms as "Equivalent", and use only the long name to annotate any interaction to another protein in that sentence;
6. Annotate all mentions of an interaction within a document, i.e., if two proteins are repeatedly mentioned as interacting in two or more separate sentences, all mentions should be annotated;
7. Whenever possible, assign relation directionality according to the text (e.g., "A regulates B" vs. "A is regulated by B");
8. Consider all proteins, irrespective of organism;
9. Mentions of protein names as part of a pathway should be annotated (e.g. "mTOR signaling");
10. Mentions of protein complexes and protein families should be annotated, and mentions of interactions involving a protein complex or protein family should be annotated as PPI.

C.1.1 Example sentences

For reference, proteins are shown in bold, action words are shown underlined, and speculative/negation words are shown in italics.

- "…we show that extracellular **-synuclein** released from neuronal cells is an endogenous agonist for **Toll-like receptor 2 (TLR2)**, which activates inflammatory responses in microglia." [PMID:23463005]
- "We identified transcription factors that are likely to bind the **PRE**, using competition gel shift and gel supershift: **Activator protein 2 (AP2)**, **nm23 nucleoside diphosphate kinase/metastatic inhibitory protein (PuF)**, and **specificity protein 1 (SP1)**." [PMID:23368879]
- "**Pin1** deficiency is suggested to cause **Tau** hyperphosphorylation in Alzheimer disease." [PMID:23362255]

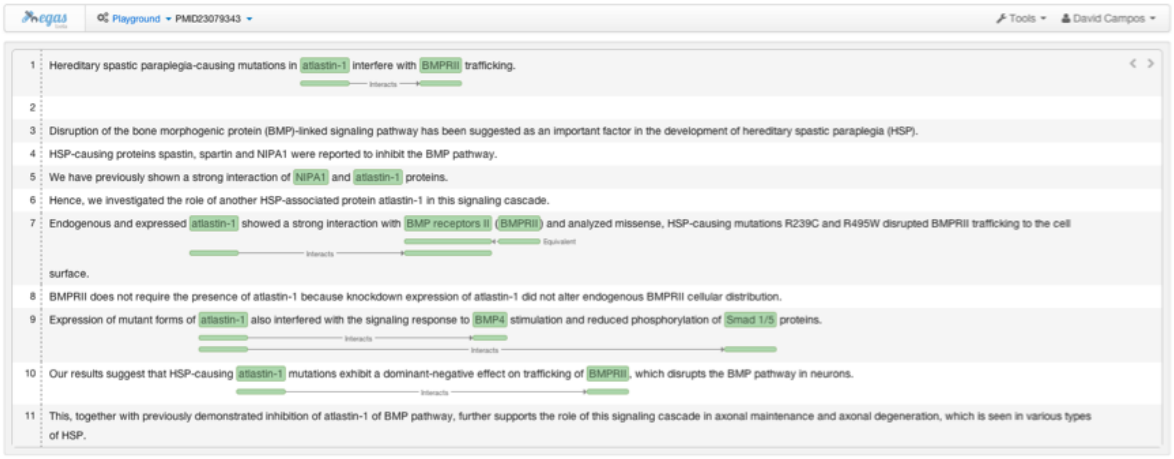
C.2 What to not annotate?

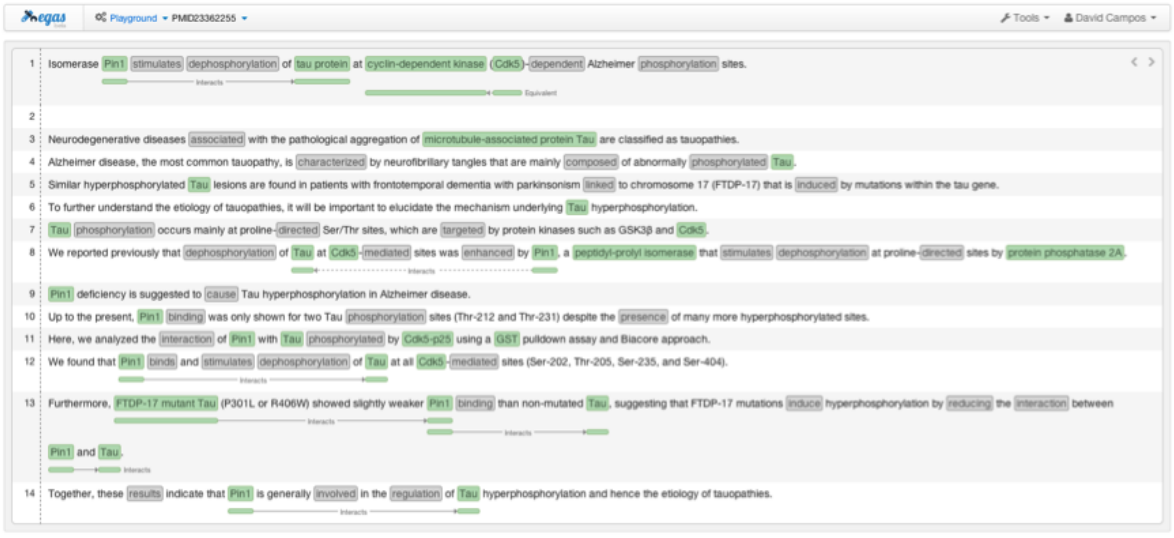
1. PPIs described over two sentences, that is, if one or both proteins are mentioned implicitly or through an anaphoric expression (e.g., "It interacts with…");
2. References to PPIs described in previous works (e.g., "…has been shown to interact with…");
3. Self-interactions, i.e., mentions of a protein interacting with itself.
4. Other bio-molecular events involving a single protein.

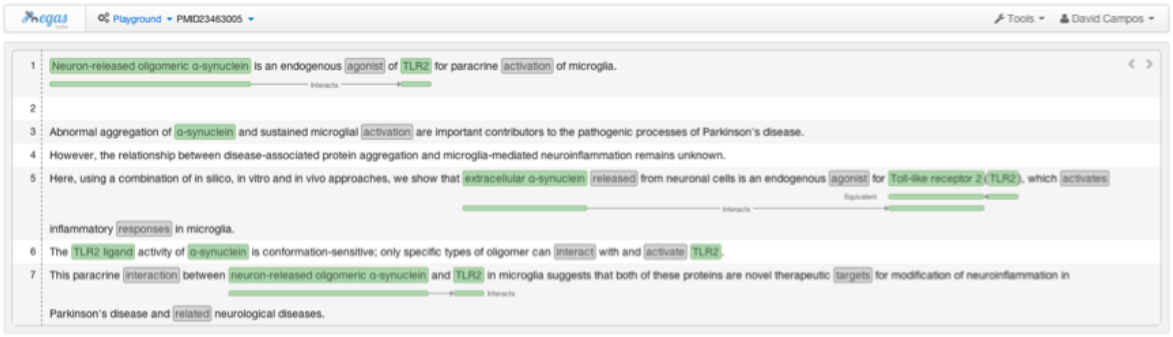
C.2.1 Example sentences

- "We have previously shown a strong interaction of **NIPA1** and **atlastin-1** proteins." [PMID:23079343]
- "**DJ-1** has multiple functions that include transcriptional regulation, anti-oxidative reaction and chaperone and mitochondrial regulation." [PMID:23326576]

C.3 Example documents






Playground
PMID23463005
Tools
David Campos
1 Neuron-released oligomeric α-synuclein is an endogenous agonist of TLR2 for paracrine activation of microglia.
Interacts
2
3 Abnormal aggregation of α-synuclein and sustained microglial activation are important contributors to the pathogenic processes of Parkinson's disease.
4 However, the relationship between disease-associated protein aggregation and microglia-mediated neuroinflammation remains unknown.
5 Here, using a combination of in silico, in vitro and in vivo approaches, we show that extracellular α-synuclein released from neuronal cells is an endogenous agonist for Toll-like receptor 2 (TLR2), which activates
Equivalent
Interacts
inflammatory responses in microglia.
6 The TLR2 ligand activity of α-synuclein is conformation-sensitive; only specific types of oligomer can interact with and activate TLR2.
7 This paracrine interaction between neuron-released oligomeric α-synuclein and TLR2 in microglia suggests that both of these proteins are novel therapeutic targets for modification of neuroinflammation in
Interacts
Parkinson's disease and related neurological diseases.

Bibliography

[1] M. Hilbert and P. López, "The world's technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[2] R. Blumberg and S. Atre, "The problem with unstructured data," *DM REVIEW*, vol. 13, pp. 42–49, 2003.

[3] R. L. Ackoff, "From data to wisdom," *Journal of Applied Systems Analysis*, vol. 16, no. 1, pp. 3–9, 1989.

[4] K. Franzen, "Protein names and how to find them," *International journal of medical informatics*, vol. 67, no. 1-3, pp. 49–61, 2002.

[5] R. A. B. Yates and B. R. Neto, *Modern information retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[6] L. Hirschman, "The evolution of evaluation: Lessons from the message understanding conferences," *Computer Speech & Language*, vol. 12, pp. 281–305, 1998.

[7] F. S. Collins, "The Human Genome Project: Lessons from Large-Scale Biology," *Science*, vol. 300, no. 5617, pp. 286–290, Apr. 2003.

[8] N. Siva, "1000 Genomes project." *Nature biotechnology*, vol. 26, no. 3, p. 256, Mar. 2008.

[9] M. Weeber, H. Klein, a. R. Aronson, J. G. Mork, L. T. de Jong-van den Berg, and R. Vos, "Text-based discovery in biomedicine: the architecture of the DAD-system." in *AMIA Annual Symposium Proceedings*, Los Angeles, CA, USA, Nov. 2000, pp. 903–907.

[10] K. B. C. Lawrence Hunter, "Biomedical Language Processing: Perspective What's Beyond PubMed?" *Molecular cell*, vol. 21, no. 5, p. 589, Mar. 2006.

[11] W. a. Baumgartner, K. B. Cohen, L. M. Fox, G. Acquaah-Mensah, and L. Hunter, "Manual curation is not sufficient for annotation of genomic databases," *Bioinformatics (Oxford, England)*, vol. 23, no. 13, pp. i41–i48, 2007.

[12] A. M. Cohen and W. R. Hersh, "A survey of current work in biomedical text mining." *Briefings in bioinformatics*, vol. 6, no. 1, pp. 57–71, 2005.

[13] A. Vlachos, "Semi-supervised learning for biomedical information extraction," Ph.D. dissertation, University of Cambridge, Computer Laboratory, Nov. 2010.

[14] W. J. Wilbur, A. Rzhetsky, and H. Shatkay, "New directions in biomedical text annotation: definitions, guidelines and corpus construction." *BMC bioinformatics*, vol. 7, p. 356, 2006.

[15] Y. Sasaki, S. Montemagni, P. Pezik, D. Rebholz-Schuhmann, J. McNaught, and S. Ananiadou, "Biolexicon: A lexical resource for the biology domain," in *Proceedings of the Third International Symposium on Semantic Mining in Biomedine (SMBM)*, Turku, Finland, 2008, pp. 109–116.

[16] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering, and L. J. Jensen, "STRING v9.1: protein-protein interaction networks, with increased coverage and integration." *Nucleic acids research*, vol. 41, no. Database issue, pp. D808–15, Jan. 2013.

[17] D. R. Swanson, "Complementary structures in disjoint science literatures," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 1991, pp. 280–289.

[18] R. Hoffmann and A. Valencia, "Implementing the iHOP concept for navigation of biomedical literature," *Bioinformatics (Oxford, England)*, vol. 21 Suppl 2, pp. ii252–8, Sep. 2005.

[19] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "FACTA: a text search engine for finding associated biomedical concepts," *Bioinformatics (Oxford, England)*, vol. 24, no. 21, p. 2559, 2008.

[20] Y. Tsuruoka, M. Miwa, K. Hamamoto, J. Tsujii, and S. Ananiadou, "Discovering and visualizing indirect associations between biomedical concepts." *Bioinformatics (Oxford, England)*, vol. 27, no. 13, pp. i111–9, Jul. 2011.

[21] D. R. Swanson, "Medical literature as a potential source of new knowledge." *Bull Med Libr Assoc*, vol. 78, no. 1, pp. 29–37, Jan. 1990.

[22] R. A. Digiacomo, J. M. Kremer, and D. M. Shah, "Fish-oil dietary supplementation in patients with Raynaud's phenomenon: A double-blind, controlled, prospective study," *The American journal of medicine*, vol. 86, no. 2, pp. 158–164, Jan. 1989.

[23] P. Schiapparelli, G. Allais, I. Castagnoli Gabellari, S. Rolando, M. G. Terzi, and C. Benedetto, "Non-pharmacological approach to migraine prophylaxis: part II." *Neurological sciences : official journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology*, vol. 31 Suppl 1, pp. S137–9, Jun. 2010.

[24] G. Trifiro, A. Fourrier-Reglat, M. C. J. M. Sturkenboom, C. Díaz Acedo, and J. Van Der Lei, "The EU-ADR project: preliminary results and perspective." *Studies in health technology and informatics*, vol. 148, pp. 43–49, 2009.

[25] J. L. Oliveira, P. Lopes, T. Nunes, D. Campos, S. Boyer, E. Ahlberg, E. M. van Mulligen, J. a. Kors, B. Singh, L. I. Furlong, F. Sanz, A. Bauer-Mehren, M. C. Carrascosa, J. Mestres, P. Avillach, G. Diallo, C. Díaz Acedo, and J. Van Der Lei, "The EU-ADR Web Platform: delivering advanced pharmacovigilance tools." *Pharmacoepidemiology and drug safety*, vol. 22, no. 5, pp. 459–467, May 2013.

[26] D. Campos, S. Matos, and J. L. Oliveira, "Current methodologies for biomedical Named Entity Recognition," in *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, M. Elloumi and A. Y. Zomaya, Eds. John Wiley & Sons, Inc., Dec. 2013, pp. 839–868.

[27] D. Campos, S. Matos, and J. L. Oliveira, "Biomedical Named Entity Recognition: A Survey of Machine-Learning Tools," in *Theory and Applications for Advanced Text Mining*, S. Sakurai, Ed. InTech, 2012, pp. 175–195.

[28] D. Campos, S. Matos, and J. L. Oliveira, "Gimli: open source and high-performance biomedical name recognition." *BMC bioinformatics*, vol. 14, p. 54, 2013.

[29] D. Campos, S. Matos, and J. L. Oliveira, "Chemical name recognition with harmonized feature-rich conditional random fields," *Fourth BioCreative Challenge Evaluation Workshop*, vol. 2, pp. 82–87, 2013.

[30] D. Campos, S. Matos, I. Lewin, J. L. Oliveira, and D. Rebholz-Schuhmann, "Harmonization of gene/protein annotations: towards a gold standard MEDLINE." *Bioinformatics (Oxford, England)*, vol. 28, no. 9, pp. 1253–1261, May 2012.

[31] D. Campos, S. Matos, and J. L. Oliveira, "A modular framework for biomedical concept recognition." *BMC bioinformatics*, vol. 14, no. 1, p. 281, Sep. 2013.

[32] D. Campos, S. Matos, and J. L. Oliveira, "Neji: a tool for heterogeneous biomedical concept identification," in *BioLINK SIG, ISMB/ECCB*, Berlin, Germany, Jul. 2013, pp. 28–31.

[33] T. Nunes, D. Campos, S. Matos, and J. L. Oliveira, "BeCAS: biomedical concept recognition services and visualization." *Bioinformatics (Oxford, England)*, vol. 29, no. 15, pp. 1915–1916, Jun. 2013.

[34] D. Campos, Q.-C. Bui, S. Matos, and J. L. Oliveira, "TrigNER: automatically optimized biomedical event trigger recognition on scientific documents," *Source code for biology and medicine*, vol. 9, no. 1, Jan. 2014.

[35] D. Campos, J. Lourenço, T. Nunes, R. Vitorino, P. Domingues, S. Matos, and J. L. Oliveira, "Egas-Collaborative Biomedical Annotation as a Service," in *Fourth BioCreative Challenge Evaluation Workshop*, Bethesda, Maryland, USA, Oct. 2013, pp. 254–259.

[36] S. Matis-Mitchell, P. Roberts, C. O. Tudor, and C. N. Arighi, "BioCreative IV Interactive Task," in *Fourth BioCreative Challenge Evaluation Workshop*, Bethesda, MD, USA, Oct. 2013, pp. 190–203.

[37] D. Campos, J. Lourenço, S. Matos, and J. L. Oliveira, "Egas: a web-based document curation platform," *Database (Oxford), Under Review.*

[38] Z. Lu, H.-Y. Kao, C.-H. Wei, M. Huang, J. Liu, C.-J. Kuo, C.-N. Hsu, R. T.-H. Tsai, H.-J. Dai, N. Okazaki, H.-C. Cho, M. Gerner, I. Solt, S. Agarwal, F. Liu, D. Vishnyakova, P. Ruch, M. Romacker, F. Rinaldi, S. Bhattacharya, P. Srinivasan, H. Liu, M. Torii, S. Matos, D. Campos, K. Verspoor, K. M. Livingston, and W. J. Wilbur, "The gene normalization task in BioCreative III." *BMC bioinformatics*, vol. 12 Suppl 8, p. S2, 2011.

[39] Q. C. Bui, E. M. Van Mulligen, D. Campos, and J. A. Kors, "A fast rule-based approach for biomedical event extraction," in *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, Aug. 2013, pp. 104–108.

[40] D. Campos, D. Rebholz-Schuhmann, S. Matos, and J. L. Oliveira, "A CRF-based approach to harmonize heterogeneous gene/protein annotations," in *Second CALBC Workshop*, Cambridge, UK, Mar. 2011, pp. 17–18.

[41] D. Campos, S. Matos, and J. L. Oliveira, "Annotating the CALBC corpus with a machine learning harmonization approach," in *Second CALBC Workshop*, Cambridge, UK, Mar. 2011, pp. 43–45.

[42] P. Lopes, D. Campos, and J. L. Oliveira, "A tagging system for bioinformatics resources," *2010 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB 2010)*, pp. 1–4, 2010.

[43] D. Campos, S. Matos, and J. L. Oliveira, "Recognition of Gene/Protein names using Conditional Random Fields," in *International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, Valencia, Spain, Oct. 2010, pp. 275–280.

[44] S. Matos, D. Campos, and J. L. Oliveira, "Vector-space models and terminologies in gene normalization and document classification," in *Proceedings of the Third BioCreative Challenge Workshop*, Bethesda, Maryland, USA, 2010, pp. 119–124.

[45] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: are they so different? A comparison," *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, 2012.

[46] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, and A. Bairoch, "UniProtKB/Swiss-Prot." *Methods in molecular biology (Clifton, N.J.)*, vol. 406, pp. 89–112, 2007.

[47] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, and others, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nature genetics*, vol. 25, no. 1, p. 25, 2000.

[48] O. Lassila and R. R. Swick, "Resource description framework (RDF) model and syntax specification," *W3C*, Feb. 1999.

[49] F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette, "Bio2RDF: towards a mashup to build bioinformatics knowledge systems." *Journal of Biomedical Informatics*, vol. 41, no. 5, pp. 706–716, Oct. 2008.

[50] M. J. Schuemie, M. Weeber, B. J. A. Schijvenaars, E. M. Van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons, and J. A. Kors, "Distribution of information in biomedical abstracts and full-text publications." *Bioinformatics (Oxford, England)*, vol. 20, no. 16, pp. 2597–2604, Nov. 2004.

[51] Y. He and M. Kayaalp, "A Comparison of 13 Tokenizers on MEDLINE," Bethesda, MD, Tech. Rep. LHNCBC-TR-2006-003, 2006.

[52] K. Verspoor, K. B. Cohen, A. Lanfranchi, C. Warner, H. L. Johnson, C. Roeder, J. D. Choi, C. Funk, Y. Malenkiy, M. Eckert, N. Xue, W. a. Baumgartner, M. Bada, M. Palmer, and L. E. Hunter, "A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools." *BMC bioinformatics*, vol. 13, p. 207, 2012.

[53] R. Sætre, K. Yoshida, A. Yakushiji, Y. Miyao, Y. Matsubyashi, and T. Ohta, "AKANE system: protein-protein interaction pairs in BioCreAtIvE2 challenge, PPI-IPS subtask," in *Proceedings of the Second BioCreative Challenge Workshop*. Madrid, Spain: Citeseer, Apr. 2007, pp. 209–212.

[54] K. Tomanek, J. Wermter, and U. Hahn, "A reappraisal of sentence and token splitting for life sciences documents," *Studies in health technology and informatics*, vol. 129, no. Pt 1, pp. 524–528, 2007.

[55] Y. Tsuruoka, Y. Tateishi, J. D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, "Developing a robust part-of-speech tagger for biomedical text," *Advances in informatics*, pp. 382–392, 2005.

[56] H. Liu, T. Christiansen, W. a. Baumgartner, and K. Verspoor, "BioLemmatizer: a lemmatization tool for morphological processing of biomedical text." *Journal of biomedical semantics*, vol. 3, p. 3, 2012.

[57] N. Kang, E. M. Van Mulligen, and J. A. Kors, "Comparing and combining chunkers of biomedical text," *Journal of Biomedical Informatics*, vol. 44, no. 2, pp. 354–360, Apr. 2011.

[58] K. Sagae, "Dependency parsing and domain adaptation with LR models and parser ensembles," in *Eleventh Conference on Computational Natural Language Learning*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 1044–1050.

[59] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, Aug. 2013, pp. 455–465.

[60] Y. Miyao and J. Tsujii, "Feature forest models for probabilistic HPSG parsing," *Computational Linguistics*, vol. 34, no. 1, pp. 35–80, 2008.

[61] S. Petrov and D. Klein, "Improved Inference for Unlexicalized Parsing." in *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Rochester, NY, USA, Apr. 2007, pp. 404–411.

[62] M. Lease and E. Charniak, "Parsing biomedical literature," in *Natural Language Processing–IJCNLP*, Jeju Island, Korea, Oct. 2005, pp. 58–69.

[63] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan, "Recognizing names in biomedical texts: a machine learning approach," *Bioinformatics (Oxford, England)*, vol. 20, no. 7, pp. 1178–1190, May 2004.

[64] S. Ananiadou and J. McNaught, *Text Mining for Biology and Biomedicine*. Artech House, 2006.

[65] Y. Tsuruoka, J. McNaught, J. Tsujii, and S. Ananiadou, "Learning string similarity measures for gene/protein name dictionary look-up using logistic regression." *Bioinformatics (Oxford, England)*, vol. 23, no. 20, pp. 2768–2774, Oct. 2007.

[66] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez Gene: gene-centered information at NCBI," *Nucleic acids research*, vol. 33, no. suppl 1, pp. D54–D58, 2005.

[67] S. Povey, R. Lovering, E. Bruford, M. Wright, M. Lush, and H. Wain, "The HUGO Gene Nomenclature Committee (HGNC)." *Human genetics*, vol. 109, no. 6, pp. 678–680, Dec. 2001.

[68] S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin, "dbSNP: the NCBI database of genetic variation." *Nucleic acids research*, vol. 29, no. 1, pp. 308–311, Jan. 2001.

[69] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[70] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch, "ExPASy: The proteomics server for in-depth protein knowledge and analysis." *Nucleic acids research*, vol. 31, no. 13, pp. 3784–3788, Jul. 2003.

[71] P. Tomasulo, "ChemIDplus-super source for chemical and drug information." *Medical reference services quarterly*, vol. 21, no. 1, pp. 53–59, 2002.

[72] D. S. Wishart, D. Tzur, C. Knox, R. Eisner, A. C. Guo, N. Young, D. Cheng, K. Jewell, D. Arndt, S. Sawhney, C. Fung, L. Nikolai, M. Lewis, M.-A. Coutouly, I. Forsythe, P. Tang, S. Shrivastava, K. Jeroncic, P. Stothard, G. Amegbey, D. Block, D. D. Hau, J. Wagner, J. Miniaci, M. Clements, M. Gebremedhin, N. Guo, Y. Zhang, G. E. Duggan, G. D. Macinnis, A. M. Weljie, R. Dowlatabadi, F. Bamforth, D. Clive, R. Greiner, L. Li, T. Marrie, B. D. Sykes, H. J. Vogel, and L. Querengesser, "HMDB: the Human Metabolome Database." *Nucleic acids research*, vol. 35, no. Database issue, pp. D521–6, Jan. 2007.

[73] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey, "DrugBank: a comprehensive resource for in silico drug discovery and exploration." *Nucleic acids research*, vol. 34, no. Database issue, pp. D668–72, Jan. 2006.

[74] M. Hewett, D. E. Oliver, D. L. Rubin, K. L. Easton, J. M. Stuart, R. B. Altman, and T. E. Klein, "PharmGKB: the pharmacogenetics knowledge base," *Nucleic acids research*, vol. 30, no. 1, p. 163, 2002.

[75] S. Liu, W. Ma, R. Moore, V. Ganesan, and S. Nelson, "RxNorm: prescription for electronic drug information exchange," *IT professional*, vol. 7, no. 5, pp. 17–23, 2005.

[76] M. Kanehisa and S. Goto, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, p. 27, 2000.

[77] L. Y. Geer, A. Marchler-Bauer, R. C. Geer, L. Han, J. He, S. He, C. Liu, W. Shi, and S. H. Bryant, "The NCBI BioSystems database," *Nucleic acids research*, vol. 38, no. Database, pp. D492–D496, Dec. 2009.

[78] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders," *Nucleic acids research*, vol. 33, no. suppl 1, pp. D514–D517, 2005.

[79] K. A. Spackman, K. E. Campbell, and R. A. Côté, "SNOMED RT: a reference terminology for health care." in *Proceedings of the AMIA Annual Fall Symposium*, Nashville, TN, USA, Oct. 1997, pp. 640–644.

[80] C. E. Lipscomb, "Medical Subject Headings (MeSH)," *Bull Med Libr Assoc*, vol. 88, no. 3, pp. 265–266, Jul. 2000.

[81] C. J. Mattingly, G. T. Colby, J. N. Forrest, and J. L. Boyer, "The Comparative Toxicogenomics Database (CTD)." *Environmental health perspectives*, vol. 111, no. 6, pp. 793–795, May 2003.

[82] E. G. Brown, L. Wood, and S. Wood, "The medical dictionary for regulatory activities (MedDRA)." *Drug safety : an international journal of medical toxicology and drug experience*, vol. 20, no. 2, pp. 109–117, Feb. 1999.

[83] K. Degtyarenko, P. De Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, and M. Ashburner, "ChEBI: a database and ontology for chemical entities of biological interest," *Nucleic acids research*, vol. 36, no. suppl 1, pp. D344–D350, 2008.

[84] J. Bard, S. Y. Rhee, and M. Ashburner, "An ontology for cell types." *Genome Biology*, vol. 6, no. 2, p. R21, 2005.

[85] D. A. Natale, C. N. Arighi, W. C. Barker, J. A. Blake, C. J. Bult, M. Caudy, H. J. Drabkin, P. D'Eustachio, A. V. Evsikov, H. Huang, J. Nchoutmboube, N. V. Roberts, B. Smith, J. Zhang, and C. H. Wu, "The Protein Ontology: a structured representation of protein forms and complexes." *Nucleic acids research*, vol. 39, no. Database issue, pp. D539–45, Jan. 2011.

[86] K. Eilbeck, S. E. Lewis, C. J. Mungall, M. Yandell, L. Stein, R. Durbin, and M. Ashburner, "The Sequence Ontology: a tool for the unification of genome annotations." *Genome Biology*, vol. 6, no. 5, p. R44, 2005.

[87] S. Federhen, "The NCBI Taxonomy database." *Nucleic acids research*, vol. 40, no. Database issue, pp. D136–43, Jan. 2012.

[88] M. A. Haendel, F. Neuhaus, D. Osumi-Sutherland, P. M. Mabee, J. L. Mejino, Jr, C. J. Mungall, and B. Smith, "CARO–the common anatomy reference ontology," in *Anatomy Ontologies for Bioinformatics: Principles and Practice*, A. Burger, D. Davidson, and R. Baldock, Eds. Springer, 2008, pp. 327–349.

[89] A. T. McCray, "The UMLS semantic network," in *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care*, Washington DC, USA, Nov. 1989, pp. 503–507.

[90] L. Smith, L. K. Tanabe, R. J. n. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, W. a. Baumgartner, L. Hunter, B. Carpenter, R. T.-H. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Maña-lópez, J. Mata, and W. J. Wilbur, "Overview of BioCreative II gene mention recognition." *Genome Biology*, vol. 9 Suppl 2, no. Suppl 2, p. S2, 2008.

[91] J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier, "Introduction to the bio-entity recognition task at JNLPBA," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. Geneva, Switzerland: Association for Computational Linguistics, 2004, pp. 70–75.

[92] L. Tanabe, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur, "GENETAG: a tagged corpus for gene/protein named entity recognition." *BMC bioinformatics*, vol. 6 Suppl 1, p. S3, 2005.

[93] U. Hahn, E. Beisswanger, E. Buyko, M. Poprat, K. Tomanek, and J. Wermter, "Semantic Annotations for Biology—A Corpus Development Initiative at the Jena University Language & Information Engineering (JULIE) Lab," in *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May 2008, pp. 28–30.

[94] S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, L. Ungar, S. Winters, and P. White, "Integrated annotation for biomedical information extraction," in *Proceedings of the Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, Boston, Massachusetts, USA, May 2004, pp. 61–68.

[95] A. A. Morgan, Z. Lu, X. Wang, A. M. Cohen, J. Fluck, P. Ruch, A. Divoli, K. Fundel, R. Leaman, J. Hakenberg, C. Sun, H.-h. Liu, R. Torres, M. Krauthammer, W. W. Lau, H. Liu, C.-N. Hsu, M. Schuemie, K. B. Cohen, and L. Hirschman, "Overview of BioCreative II gene normalization." *Genome Biology*, vol. 9 Suppl 2, p. S3, 2008.

[96] N. Naderi, T. Kappler, C. J. Baker, and R. Witte, "OrganismTagger: detection, normalization and grounding of organism entities in biomedical documents," *Bioinformatics (Oxford, England)*, vol. 27, no. 19, pp. 2721–2729, Oct. 2011.

[97] M. Gerner, G. Nenadic, and C. M. Bergman, "LINNAEUS: a species name identification system for biomedical literature." *BMC bioinformatics*, vol. 11, p. 85, 2010.

[98] H. Gurulingappa, R. Klinger, M. Hofmann-Apitius, and J. Fluck, "An Empirical Evaluation of Resources for the Identification of Diseases and Adverse Effects in Biomedical Literature," in *2nd Workshop on Building and evaluating resources for biomedical text mining (7th edition of the Language Resources and Evaluation Conference)*, Valetta, Malta, 2010, p. 15.

[99] A. Jimeno, E. Jimenez-Ruiz, V. Lee, S. Gaudan, R. Berlanga, and D. Rebholz-Schuhmann, "Assessment of disease named entity recognition on a corpus of annotated sentences." *BMC bioinformatics*, vol. 9 Suppl 3, p. S3, 2008.

[100] R. Leaman, C. Miller, and G. Gonzalez, "Enabling recognition of diseases in biomedical text with machine learning: Corpus and benchmark," in *3rd International Symposium on Languages in Biology and Medicine*, Jeju Island, South Korea, 2009, pp. 82–89.

[101] B. Rosario and M. A. Hearst, "Classifying semantic relations in bioscience texts," in *42nd annual meeting of the Association for Computational Linguistics*. Barcelona, Spain: Association for Computational Linguistics, 2004, p. 430.

[102] R. Klinger, C. Kolárik, J. Fluck, M. Hofmann-Apitius, and C. M. Friedrich, "Detection of IUPAC and IUPAC-like chemical names," *Bioinformatics (Oxford, England)*, vol. 24, no. 13, pp. i268–76, Jul. 2008.

[103] C. Kolárik, R. Klinger, C. M. Friedrich, M. Hofmann-Apitius, and J. Fluck, "Chemical names: terminological resources and corpora annotation," in *Workshop on Building and evaluating resources for biomedical text mining (Language Resources and Evaluation Conference)*, Marrakech, Morocco, May 2008, pp. 51–58.

[104] M. Krallinger, F. Leitner, O. Rabal, and M. Vazquez, "Overview of the chemical compound and drug name recognition (CHEMDNER) task," in *Fourth BioCreative Challenge Evaluation Workshop*, Washington, DC, USA, Oct. 2013, pp. 2–33.

[105] T. Ohta, S. Pyysalo, J. Tsujii, and S. Ananiadou, "Open-domain Anatomical Entity Mention Detection," in *Workshop on Detecting Structure in Scholarly Discourse*, Jeju Island, Korea, 2012, pp. 27–36.

[106] M. Neves, A. Damaschun, A. Kurtz, and U. Leser, "Annotating and evaluating text for stem cell research," in *Third Workshop on Building and Evaluation Resources for Biomedical Text Mining*, Istanbul,Turkey, May 2012, pp. 16–23.

[107] M. Bada, M. Eckert, D. Evans, K. Garcia, K. Shipley, D. Sitnikov, W. a. Baumgartner, K. B. Cohen, K. Verspoor, J. A. Blake, and L. E. Hunter, "Concept annotation in the CRAFT corpus." *BMC bioinformatics*, vol. 13, p. 161, 2012.

[108] D. Rebholz-Schuhmann, A. J. J. Yepes, E. M. van Mulligen, N. Kang, J. Kors, D. Milward, P. Corbett, E. Buyko, E. Beisswanger, and U. Hahn, "CALBC silver standard corpus." *Journal of bioinformatics and computational biology*, vol. 8, no. 1, pp. 163–179, Feb. 2010.

[109] H. Yu, "Towards answering biological questions with experimental evidence: automatically identifying text that summarize image content in full-text articles," in *Proceedings of the AMIA Annual Symposium*. Washington, D.C., WA, USA: American Medical Informatics Association, 2006, pp. 834–838.

[110] K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi, "Toward information extraction: identifying protein names from biological papers." in *Pacific Symposium on Biocomputing*, Big Island, Hawaii, USA, Jan. 1998, pp. 707–718.

[111] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett, "Protein structures and information extraction from biological texts: the PASTA system." *Bioinformatics (Oxford, England)*, vol. 19, no. 1, pp. 135–143, Jan. 2003.

[112] A. S. Schwartz and M. A. Hearst, "A simple algorithm for identifying abbreviation definitions in biomedical text." in *Pacific Symposium on Biocomputing*. Hawaii, HI, USA: Computer Science Division, University of California, Berkeley, Berkeley, CA 94720, USA, 2003, pp. 451–462.

[113] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[114] W. E. Winkler, "The state of record linkage and current research problems," Statistical Research Division, U.S. Census Bureau, Tech. Rep., 1999.

[115] W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg, "A Comparison of String Distance Metrics for Name-Matching Tasks." in *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb), AAAI,*, Acapulco, Mexico, Aug. 2003, pp. 73–78.

[116] H. Liu, Z. Z. Hu, J. Zhang, and C. Wu, "BioThesaurus: a web-based thesaurus of protein and gene names," *Bioinformatics (Oxford, England)*, vol. 22, no. 1, pp. 103–105, Jan. 2006.

[117] P. Thompson, J. McNaught, S. Montemagni, N. Calzolari, R. Del Gratta, V. Lee, S. Marchi, M. Monachini, P. Pezik, V. Quochi, C. Rupp, Y. Sasaki, G. Venturi, D. Rebholz-Schuhmann, and S. Ananiadou, "The BioLexicon: a large-scale terminological resource for biomedical text mining," *BMC bioinformatics*, vol. 12, p. 397, 2011.

[118] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl 1, p. D267, 2004.

[119] J. Arrais, J. E. Pereira, J. Fernandes, and J. L. Oliveira, "GeNS: a biological data integration platform," *Proceedings of the International Conference on Bioinformatics and Biomedicine (ICBB 2009): 26-29 October 2009; Venice, Italy*, pp. 850–855, 2009.

[120] K. M. Hettne, R. H. Stierum, M. J. Schuemie, P. J. M. Hendriksen, B. J. a. Schijvenaars, E. M. v. Mulligen, J. Kleinjans, and J. a. Kors, "A dictionary to identify small molecules and drugs in free text." *Bioinformatics (Oxford, England)*, vol. 25, no. 22, pp. 2983–2991, Nov. 2009.

[121] A. M. Cohen, W. R. Hersh, C. Dubay, and K. A. Spackman, "Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts," *BMC bioinformatics*, vol. 6, p. 103, 2005.

[122] A. M. Cohen, "Unsupervised gene/protein named entity normalization using automatically extracted dictionaries," in *Proceedings of the acl-ismb workshop on linking biological literature, ontologies and databases: Mining biological semantics*. Michigan, MI, USA: Association for Computational Linguistics, Jun. 2005, pp. 17–24.

[123] M. J. Schuemie, B. Mons, M. Weeber, and J. a. Kors, "Evaluation of techniques for increasing recall in a dictionary approach to gene and protein name identification." *Journal of Biomedical Informatics*, vol. 40, no. 3, pp. 316–324, 2007.

[124] A. T. McCray, S. Srinivasan, and A. C. Browne, "Lexical methods for managing variation in biomedical terminologies." in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, Bethesda, Washington, USA, Nov. 1994, pp. 235–239.

[125] D. Hanisch, J. Fluck, H. T. Mevisssen, and R. Zimmer, "Playing Biology's Name Game: Identifying Protein Names in Scientific Text," *Symposium A Quarterly Journal In Modern Foreign Literatures*, vol. 414, pp. 403–414, 2003.

[126] D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, and A. Jimeno, "Text processing through Web services: calling Whatizit," *Bioinformatics (Oxford, England)*, vol. 24, no. 2, pp. 296–298, Jan. 2008.

[127] K. Fundel, D. Güttler, R. Zimmer, and J. Apostolakis, "Exact versus approximate string matching for protein name identification," in *BioCreative Challenge Evaluation Workshop*, Granada, Spain, Mar. 2004, pp. 1–5.

[128] E. Fredkin, "Trie memory," *Communications of the ACM*, vol. 3, no. 9, pp. 490–499, 1960.

[129] U. Manber and G. Myers, "Suffix arrays: a new method for on-line string searches," in *Suffix arrays: a new method for on-line string searches.* Society for Industrial and Applied Mathematics, 1990, pp. 319–327.

[130] D. R. Morrison, "PATRICIA—practical algorithm to retrieve information coded in alphanumeric," *Journal of the ACM (JACM)*, vol. 15, no. 4, pp. 514–534, 1968.

[131] H. Shang, T. H. Merrettal, R. S. Eng, S. Inc, and C. A. Emeryville, "Tries for approximate string matching," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 4, pp. 540–547, 1996.

[132] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943.

[133] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics." *Bioinformatics (Oxford, England)*, vol. 23, no. 19, pp. 2507–2517, 2007.

[134] J. Hakenberg, S. Bickel, C. Plake, U. Brefeld, H. Zahn, L. Faulstich, U. Leser, and T. Scheffer, "Systematic feature evaluation for gene name recognition." *BMC bioinformatics*, vol. 6 Suppl 1, p. S9, 2005.

[135] S. Della Pietra, V. Della Pietra, J. Lafferty, R. Technol, and S. Brook, "Inducing features of random fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[136] C. J. Kuo, Y. M. Chang, H. S. Huang, K. T. Lin, B. H. Yang, Y. S. Lin, C. N. Hsu, and I. F. Chung, "Rich feature set, unification of bidirectional parsing and dictionary filtering for high F-score gene mention tagging," in *Second BioCreative Challenge Evaluation Workshop*, Madrid, Spain, 2007, pp. 105–107.

[137] R. T.-H. Tsai, C.-L. Sung, H.-J. Dai, H.-C. Hung, T.-y. Sung, and W.-l. Hsu, "NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition." *BMC bioinformatics*, vol. 7 Suppl 5, no. Suppl 5, p. S11, 2006.

[138] R. McDonald and F. Pereira, "Identifying gene and protein mentions in text using conditional random fields." *BMC bioinformatics*, vol. 6 Suppl 1, p. S6, 2005.

[139] J. Lafferty, A. Mccallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, Williamstown, MA, USA, Jul. 2001, pp. 282–289.

[140] H. M. Wallach, "Conditional random fields: An introduction," *Technical Reports (CIS)*, p. 22, 2004.

[141] S. S. Keerthi and S. Sundararajan, "CRF versus SVM-struct for sequence labeling," Yahoo Research, Tech. Rep., 2007.

[142] C. Lee and M. G. Jang, "Fast training of structured SVM using fixed-threshold sequential minimal optimization," *ETRI journal*, vol. 31, no. 2, pp. 121–128, 2009.

[143] G. Hoefel and C. Elkan, "Learning a two-stage SVM/CRF sequence classifier," in *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*. Napa Valley, California, USA: ACM Press, Oct. 2008, pp. 271–278.

[144] V. Cherkassky, "The nature of statistical learning theory," *IEEE Trans Neural Netw*, vol. 8, no. 6, p. 1564, 1997.

[145] O. Ivanciuc, "Applications of support vector machines in chemistry," *Reviews in computational chemistry*, vol. 23, p. 291, 2007.

[146] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," *Advances in Neural Information processing systems*, vol. 17, pp. 1185–1192, 2004.

[147] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans, "Semi-supervised conditional random fields for improved sequence segmentation and labeling," *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pp. 209–216, 2006.

[148] J. Suzuki and H. Isozaki, "Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data," *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 665–673, Jun. 2008.

[149] K. Bennett and A. Demiriz, "Semi-supervised support vector machines," *Advances in Neural Information processing systems*, pp. 368–374, 1999.

[150] S. Clark, J. R. Curran, and M. Osborne, "Bootstrapping POS taggers using unlabelled data," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, W. Daelemans and M. Osborne, Eds., Edmonton, Canada, Jun. 2003, pp. 49–55.

[151] Z. Kozareva, "Bootstrapping named entity recognition with automatically generated gazetteer lists," in *Proceedings of the eleventh conference of the European chapter of the association for computational linguistics: student research workshop*. Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 15–21.

[152] M. Becker, B. Hachey, B. Alex, and C. Grover, "Optimising Selective Sampling for Bootstrapping Named Entity Recognition," in *Proceedings of the ICML-2005 Workshop on Learning with Multiple Views*, S. Ruping and T. Scheffer, Eds., Bonn, Germany, Aug. 2005, pp. 5–11.

[153] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. Madison, Wisconsin, USA: ACM, Jul. 1998, pp. 92–100.

[154] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, P. Fung and J. Zhou, Eds., College Park, MD, USA, Jun. 1999, pp. 189–196.

[155] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," in *Proceedings of the National Conference on Artificial Intelligence*. Orlando, Florida, USA: JOHN WILEY & SONS LTD, Jul. 1999, pp. 474–479.

[156] A. P. Dempster, N. M. Laird, D. B. Rubin, and others, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[157] S. Borman, "The expectation maximization algorithm-a short tutorial," 2004.

[158] B. Merialdo, "Tagging English text with a probabilistic model," *Computational Linguistics*, vol. 20, no. 2, pp. 155–171, 1994.

[159] R. K. Ando and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Michigan, MI, USA: Association for Computational Linguistics, Jun. 2005, pp. 1–9.

[160] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," in *Proceedings of HLT-NAACL*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2004, pp. 337–342.

[161] R. K. Ando, "BioCreative II Gene Mention Tagging System at IBM Watson," in *Proceedings of the Second Biocreative Challenge Evaluation Workshop*, Madrid, Spain, Apr. 2007, pp. 101–103.

[162] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *The Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[163] G. Zhou, D. Shen, J. Zhang, J. Su, and S. Tan, "Recognition of protein/gene names from text using an ensemble of classifiers." *BMC bioinformatics*, vol. 6 Suppl 1, p. S7, 2005.

[164] S. Mika and B. Rost, "Protein names precisely peeled off free text." *Bioinformatics (Oxford, England)*, vol. 20 Suppl 1, pp. i241–7, 2004.

[165] R. Leaman and G. Gonzalez, "BANNER: an executable survey of advances in biomedical named entity recognition." in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. Big Island of Hawaii, HI, USA: Department of Computer Science and Engineering, Arizona State University, USA., 2008, pp. 652–663.

[166] S. Egorov, A. Yuryev, and N. Daraselia, "A simple and practical dictionary-based approach for identification of proteins in MEDLINE abstracts," *Journal of the American Medical Informatics Association*, vol. 11, no. 3, pp. 174–178, 2004.

[167] A. J. Jimeno-Yepes and A. R. Aronson, "Knowledge-based biomedical word sense disambiguation: comparison of approaches." *BMC bioinformatics*, vol. 11, no. 1, p. 569, 2010.

[168] M. Weeber, J. G. Mork, and a. R. Aronson, "Developing a test collection for biomedical word sense disambiguation." in *Proceedings AMIA Symposium*, S. Bakken, Ed., Washington, DC, USA, Nov. 2001, pp. 746–750.

[169] P. Edmonds and E. Agirre, Eds., *Word Sense Disambiguation: Algorithms And Applications*. Springer, Nov. 2006.

[170] R. Navigli, "Word sense disambiguation," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, Feb. 2009.

[171] J. Pustejovsky, J. Castano, R. Saurí, A. Rumshinsky, J. Zhang, and W. Luo, "Medstract: creating large-scale information servers for biomedical libraries," in *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, S. Johnson, Ed. Philadelphia, PA, USA: Association for Computational Linguistics, Jul. 2002, pp. 85–92.

[172] D. Raileanu, P. Buitelaar, S. Vintar, and J. Bay, "Evaluation Corpora for Sense Disambiguation in the Medical Domain," in *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands, Spain, Jun. 2002, pp. 609–612.

[173] M. J. Schuemie, J. a. Kors, and B. Mons, "Word sense disambiguation in the biomedical domain: an overview." *Journal of computational biology : a journal of computational molecular cell biology*, vol. 12, no. 5, pp. 554–565, 2005.

[174] G. K. Savova, A. R. Coden, I. L. Sominsky, R. Johnson, P. V. Ogren, P. C. d. Groen, and C. G. Chute, "Word sense disambiguation across two domains: Biomedical literature and clinical notes," *Journal of Biomedical Informatics*, vol. 41, no. 6, pp. 1088–1100, Dec. 2008.

[175] M. Stevenson, Y. Guo, R. Gaizauskas, and D. Martinez, "Disambiguation of biomedical text using diverse sources of information." *BMC bioinformatics*, vol. 9 Suppl 11, p. S7, 2008.

[176] M. Joshi, T. Pedersen, and R. Maclin, "A Comparative Study of Support Vector Machines Applied to the Supervised Word Sense Disambiguation Problem in the Medical Domain," *Artificial Intelligence*, pp. 3449–3468, 2005.

[177] A. Jimeno-Yepes and a. R. Aronson, "Self-training and co-training in biomedical word sense disambiguation," in *Proceedings of BioNLP 2011 Workshop*, Portland, Oregon, USA, Jun. 2011, pp. 182–183.

[178] H. Liu, V. Teller, and C. Friedman, "A multi-aspect comparison study of supervised word sense disambiguation." *Journal of the American Medical Informatics Association*, vol. 11, no. 4, pp. 320–331, Jul. 2004.

[179] H. Schütze, "Automatic word sense discrimination," *Computational Linguistics*, vol. 24, no. 1, pp. 97–123, Mar. 1998.

[180] B. Andreopoulos, D. Alexopoulou, and M. Schroeder, "Word Sense Disambiguation in biomedical ontologies with term co-occurrence analysis and document clustering." *International journal of data mining and bioinformatics*, vol. 2, no. 3, pp. 193–215, 2008.

[181] J. Hakenberg, C. Plake, R. Leaman, M. Schroeder, and G. Gonzalez, "Inter-species normalization of gene mentions with GNAT." *Bioinformatics (Oxford, England)*, vol. 24, no. 16, pp. i126–132, 2008.

[182] H. Xu, J.-W. Fan, G. Hripcsak, E. a. Mendonça, M. Markatou, and C. Friedman, "Gene symbol disambiguation using knowledge-based profiles." *Bioinformatics (Oxford, England)*, vol. 23, no. 8, pp. 1015–1022, 2007.

[183] X. Wang and M. Matthews, "Distinguishing the species of biomedical named entities for term identification." *BMC bioinformatics*, vol. 9 Suppl 11, p. S6, 2008.

[184] B. T. McInnes, "An unsupervised vector approach to biomedical term disambiguation: integrating UMLS and Medline," in *HLT-SRWS '08 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*. Columbus, Ohio: Association for Computational Linguistics, 2008, pp. 49–54.

[185] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *SIGDOC '86 Proceedings of the 5th annual international conference on Systems documentation*, V. DeBuys, Ed. New York, NY, USA: Association for Computational Linguistics, 1986, pp. 24–26.

[186] E. Agirre, A. Soroa, and M. Stevenson, "Graph-based word sense disambiguation of biomedical documents." *Bioinformatics (Oxford, England)*, vol. 26, no. 22, pp. 2889–2896, 2010.

[187] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *EACL '09 Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, Apr. 2009, pp. 33–41.

[188] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, Apr. 1998.

[189] S. Ananiadou, S. Pyysalo, J. Tsujii, and D. B. Kell, "Event extraction for systems biology by text mining the literature." *Trends in biotechnology*, vol. 28, no. 7, pp. 381–390, Jul. 2010.

[190] K. B. Cohen and L. Hunter, "A critical review of PASBio's argument structures for biomedical verbs." *BMC bioinformatics*, vol. 7 Suppl 3, p. S5, 2006.

[191] A. Rzhetsky, I. Iossifov, T. Koike, M. Krauthammer, P. Kra, M. Morris, H. Yu, P. A. Duboué, W. Weng, W. J. Wilbur, V. Hatzivassiloglou, and C. Friedman, "GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data." *Journal of Biomedical Informatics*, vol. 37, no. 1, pp. 43–53, Feb. 2004.

[192] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of BioNLP'09 shared task on event extraction," in *BioNLP Shared Task 2009 Workshop*. Boulder, Colorado, USA: Association for Computational Linguistics, Jun. 2009, pp. 1–9.

[193] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of BioNLP Shared Task 2011," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 1–6.

[194] C. Nédellec, R. Bossy, J. D. Kim, J. Kim, T. Ohta, and S. Pyysalo, "Overview of BioNLP Shared Task 2013," in *Proceedings of BioNLP Shared Task 2013 Workshop*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1–7.

[195] J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus-a semantically annotated corpus for bio-textmining," *Bioinformatics (Oxford, England)*, vol. 19, pp. i180–i182, Jul. 2003.

[196] G. D. Bader, I. Donaldson, C. Wolting, B. F. Ouellette, T. Pawson, and C. W. Hogue, "BIND–The Biomolecular Interaction Network Database." *Nucleic acids research*, vol. 29, no. 1, pp. 242–245, Jan. 2001.

[197] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: a Molecular INTeraction database," *Febs Letters*, vol. 513, no. 1, pp. 135–140, 2002.

[198] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler, "IntAct: an open source molecular interaction database." *Nucleic acids research*, vol. 32, no. Database issue, pp. D452–5, Jan. 2004.

[199] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: a general repository for interaction datasets." *Nucleic acids research*, vol. 34, no. Database issue, pp. D535–9, Jan. 2006.

[200] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, A. Kothari, A. Kubo, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, P. Subhraveti, D. S. Weaver, D. Weerasinghe, P. Zhang, and P. D. Karp, "The MetaCyc database of metabolic pathways

and enzymes and the BioCyc collection of Pathway/Genome Databases." *Nucleic acids research*, vol. 42, no. 1, pp. D459–71, Jan. 2014.

[201] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein, "Reactome: a knowledgebase of biological pathways." *Nucleic acids research*, vol. 33, no. Database issue, pp. D428–32, Jan. 2005.

[202] A. R. Pico, T. Kelder, M. P. van Iersel, K. Hanspers, B. R. Conklin, and C. Evelo, "WikiPathways: pathway editing for the people." *PLoS biology*, vol. 6, no. 7, p. e184, Jul. 2008.

[203] E. Wingender, P. Dietze, H. Karas, and R. Knüppel, "TRANSFAC: a database on transcription factors and their DNA binding sites." *Nucleic acids research*, vol. 24, no. 1, pp. 238–241, Jan. 1996.

[204] S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski, "Comparative analysis of five protein-protein interaction corpora," *BMC bioinformatics*, vol. 9, no. Suppl 3, p. S6, 2008.

[205] I. Segura-Bedmar and P. Martınez, "The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts," in *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, Sep. 2011, pp. 1–9.

[206] I. Segura-Bedmar, P. Martınez, and M. Herrero-Zazo, "SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013)," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 341–350.

[207] R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong, "Comparative experiments on learning information extractors for proteins and their interactions," *Artificial intelligence in medicine*, vol. 33, no. 2, pp. 139–155, Feb. 2005.

[208] S. Pyysalo, F. Ginter, J. Heimonen, J. Bjorne, J. Boberg, J. Jarvinen, and T. Salakoski, "BioInfer: a corpus for information extraction in the biomedical domain," *BMC bioinformatics*, vol. 8, p. 50, 2007.

[209] K. Fundel, R. Küffner, and R. Zimmer, "RelEx–relation extraction using dependency parse trees." *Bioinformatics (Oxford, England)*, vol. 23, no. 3, pp. 365–371, 2007.

[210] J. Ding, D. Berleant, and D. Nettleton, "Mining MEDLINE: abstracts, sentences, or phrases," in *Pacific Symposium on Biocomputing*, Lihue, Hawaii, USA, 2002, pp. 326–337.

[211] C. Nédellec, "Learning language in logic-genic interaction extraction challenge," in *Proceedings of the 4th Learning Language in Logic Workshop*, 2005, pp. 31–37.

[212] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, and T. Declerck, "The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions." *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 914–920, Oct. 2013.

[213] E. M. van Mulligen, A. Fourrier-Reglat, D. Gurwitz, M. Molokhia, A. Nieto, G. Trifiro, J. a. Kors, and L. I. Furlong, "The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships." *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 879–884, Oct. 2012.

[214] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 885–892, 2012.

[215] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou, "Event extraction across multiple levels of biological organization." *Bioinformatics (Oxford, England)*, vol. 28, no. 18, pp. i575–i581, Sep. 2012.

[216] J.-D. Kim, Y. Wang, T. Takagi, and A. Yonezawa, "Overview of genia event task in bionlp shared task 2011," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 7–15.

[217] J. D. Kim, Y. Wang, and Y. Yasunori, "The Genia Event Extraction Shared Task, 2013 Edition-Overview," in *Proceedings of BioNLP Shared Task 2013 Workshop*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 8–15.

[218] T. Ohta, S. Pyysalo, and J. Tsujii, "Overview of the epigenetics and post-translational modifications (EPI) task of BioNLP Shared Task 2011," in *BioNLP Shared Task '11: Proceedings of the BioNLP Shared Task 2011 Workshop*, Association for Computational Linguistics. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 16–25.

[219] S. Pyysalo, T. Ohta, R. Rak, D. Sullivan, C. Mao, C. Wang, B. Sobral, J. Tsujii, and S. Ananiadou, "Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, Portland, Oregon, Jun. 2011, pp. 26–35.

[220] S. Pyysalo, T. Ohta, and S. Ananiadou, "Overview of the Cancer Genetics (CG) task of BioNLP Shared Task 2013," in *Proceedings of BioNLP Shared Task 2013 Workshop*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 58–66.

[221] T. Ohta, S. Pyysalo, R. Rak, A. Rowley, H.-W. Chun, S.-J. Jung, S.-P. Choi, and S. Ananiadou, "Overview of the pathway curation (PC) task of bioNLP shared task 2013," in *Proceedings of BioNLP Shared Task 2013 Workshop*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 67–75.

[222] I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalickova, T. Pawson, and C. W. V. Hogue, "PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine." *BMC bioinformatics*, vol. 4, p. 11, 2003.

[223] M. Krallinger, F. Leitner, C. Rodriguez-penagos, and A. Valencia, "Overview of the protein-protein interaction annotation extraction task of BioCreative II." *Genome Biology*, vol. 9 Suppl 2, p. S4, 2008.

[224] M. Krallinger, M. Vazquez, and F. Leitner, "The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text," *BMC bioinformatics*, vol. 12, no. Suppl 8, p. S3, 2011.

[225] B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, R. Tobin, and X. Wang, "Automating curation using a natural language processing pipeline." *Genome Biology*, vol. 9 Suppl 2, p. S10, 2008.

[226] S. Kim and W. J. Wilbur, "Classifying protein-protein interaction articles using word and syntactic features." *BMC bioinformatics*, vol. 12 Suppl 8, p. S9, 2011.

[227] M. Lan, C.-L. Tan, and J. Su, "Feature generation and representations for protein-protein interaction classification." *Journal of Biomedical Informatics*, vol. 42, no. 5, pp. 866–872, Oct. 2009.

[228] A. Abi-Haidar, J. Kaur, A. Maguitman, P. Radivojac, A. Rechtsteiner, K. Verspoor, Z. Wang, and L. M. Rocha, "Uncovering protein interaction in abstracts and text using a novel linear model and word proximity networks." *Genome Biology*, vol. 9 Suppl 2, p. S11, 2008.

[229] B. P. Suomela and M. A. Andrade, "Ranking the whole MEDLINE database according to a large training set using text indexing." *BMC bioinformatics*, vol. 6, p. 75, 2005.

[230] J.-F. Fontaine, A. Barbosa-Silva, M. Schaefer, M. R. Huska, E. M. Muro, and M. a. Andrade-Navarro, "MedlineRanker: flexible ranking of biomedical literature." *Nucleic acids research*, vol. 37, no. Web Server issue, pp. W141–6, Jul. 2009.

[231] A. Casillas, A. D. de Ilarraza, K. Gojenola, M. Oronoz, and G. Rigau, "Using Kybots for extracting events in biomedical texts," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 138–142.

[232] Q. Le Minh, S. N. Truong, and Q. H. Bao, "A pattern approach for biomedical event annotation," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 149–150.

[233] H. Kilicoglu and S. Bergler, "Syntactic dependency based heuristics for biological event extraction," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*. Boulder, Colorado, USA: Association for Computational Linguistics, Jun. 2009, pp. 119–127.

[234] J. Björne, F. Ginter, and T. Salakoski, "University of Turku in the BioNLP'11 Shared Task." *BMC bioinformatics*, vol. 13 Suppl 11, p. S4, 2012.

[235] M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii, "Event extraction with complex event classification using rich features." *Journal of bioinformatics and computational biology*, vol. 8, no. 1, pp. 131–146, Feb. 2010.

[236] Y. Zhang, H. Lin, Z. Yang, J. Wang, and Y. Li, "Biomolecular event trigger detection using neighborhood hash features." *Journal of theoretical biology*, vol. 318, pp. 22–28, Feb. 2013.

[237] D. Martinez and T. Baldwin, "Word sense disambiguation for event trigger word detection in biomedicine," *BMC bioinformatics*, vol. 12, no. Suppl 2, p. S4, 2011.

[238] A. MacKinlay, D. Martinez, and T. Baldwin, "Biomedical event annotation with CRFs and precision grammars," in *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Boulder, Colorado, USA: Association for Computational Linguistics, 2009, pp. 77–85.

[239] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting complex biological events with rich graph-based feature sets," in *BioNLP Shared Task 2009 Workshop*. Boulder, Colorado, USA: Association for Computational Linguistics, 2009, pp. 10–18.

[240] H. Chen and B. M. Sharp, "Content-rich biological network constructed by mining PubMed abstracts," *BMC bioinformatics*, vol. 5, no. 1, p. 147, 2004.

[241] C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia, "Automatic extraction of biological information from scientific text: protein-protein interactions." *Proc Int Conf Intell Syst Mol Biol*, pp. 60–67, 1999.

[242] C. Plake, J. Hakenberg, and U. Leser, "Optimizing syntax patterns for discovering protein-protein interactions," in *the 2005 ACM symposium*. New York, New York, USA: ACM Press, 2005, pp. 195–201.

[243] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics (Oxford, England)*, vol. 17, no. 2, pp. 155–161, 2001.

[244] Q.-C. Bui, S. Katrenko, and P. M. A. Sloot, "A hybrid approach to extract protein-protein interactions." *Bioinformatics (Oxford, England)*, vol. 27, no. 2, pp. 259–265, Jan. 2011.

[245] M. Huang, X. Zhu, and M. Li, "A hybrid method for relation extraction from biomedical literature." *International journal of medical informatics*, vol. 75, no. 6, pp. 443–455, Jun. 2006.

[246] F. Rinaldi, G. Schneider, and S. Clematide, "Relation mining experiments in the pharmacogenomics domain." *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 851–861, Oct. 2012.

[247] Q. C. Bui and P. Sloot, "Extracting biological events from text using simple syntactic patterns," in *BioNLP Shared Task 2011 Workshop*, Portland, Oregon, USA, 2011, pp. 143–146.

[248] E. Buyko, E. Faessler, and J. Wermter, "Event extraction from trimmed dependency graphs," in *BioNLP Shared Task 2009 Workshop*, Boulder, Colorado, USA, 2009, pp. 19–27.

[249] K. Kaljurand, G. Schneider, and F. Rinaldi, "UZurich in the BioNLP 2009 shared task," in *BioNLP Shared Task 2009 Workshop*. Boulder, Colorado, USA: Association for Computational Linguistics, Jun. 2009, pp. 28–36.

[250] H. Kilicoglu and S. Bergler, "Adapting a general semantic interpretation approach to biological event extraction," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: BioNLP Shared Task 2011 Workshop, Jun. 2011, pp. 173–182.

[251] H. Jung, S.-P. Choi, S. Lee, and S.-K. Song, "Survey on Kernel-Based Relation Extraction," in *Theory and Applications for Advanced Text Mining*, S. Sakurai, Ed. InTech, 2012.

[252] S. Kim, J. Yoon, and J. Yang, "Kernel approaches for genic interaction extraction." *Bioinformatics (Oxford, England)*, vol. 24, no. 1, pp. 118–126, Jan. 2008.

[253] R. Bunescu and R. J. Mooney, "Subsequence kernels for relation extraction," in *Advances in Neural Information processing systems*, Y. Weiss, B. Scholkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, Dec. 2005, pp. 171–178.

[254] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *The Journal of Machine Learning Research*, vol. 3, pp. 1083–1106, 2003.

[255] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Human Language Technology Conference Conference on Empirical Methods in Natural Language Processing*. Vancouver, Canada: Association for Computational Linguistics, Oct. 2005, pp. 724–731.

[256] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," *Learning Theory and Kernel Machines*, vol. 2777, pp. 129–143, 2003.

[257] T. Fayruzov, M. De Cock, C. Cornelis, and V. Hoste, "Linguistic feature analysis for protein interaction extraction." *BMC bioinformatics*, vol. 10, p. 374, 2009.

[258] S. Van Landeghem, Y. Saeys, B. De Baets, and Y. Van de Peer, "Extracting protein-protein interactions from text using rich feature vectors and feature selection," *SMBM '08 : proceedings of the third symposium on semantic mining in biomedicine*, pp. 77–84, 2008.

[259] Y. Miyao, K. Sagae, R. Sætre, T. Matsuzaki, and J. Tsujii, "Evaluating contributions of natural language parsers to protein-protein interaction extraction." *Bioinformatics (Oxford, England)*, vol. 25, no. 3, pp. 394–400, 2009.

[260] R. Sætre, K. Yoshida, M. Miwa, T. Matsuzaki, Y. Kano, and J. Tsujii, "Extracting protein interactions from text with the unified AkaneRE event extraction system." *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 3, pp. 442–453, Jul. 2010.

[261] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski, "All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning." *BMC bioinformatics*, vol. 9 Suppl 11, p. S2, 2008.

[262] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "Protein-protein interaction extraction by leveraging multiple kernels and parsers." *International journal of medical informatics*, vol. 78, no. 12, pp. e39–46, Dec. 2009.

[263] E. Buyko, E. Beisswanger, and U. Hahn, "The extraction of pharmacogenetic and pharmacogenomic relations - a case study using pharmgkb." in *Pacific Symposium on Biocomputing*, The Big Island of Hawaii, Hawaii, USA, Jan. 2012, pp. 376–387.

[264] M. Miwa, P. Thompson, and S. Ananiadou, "Boosting automatic event extraction from the literature using domain adaptation and coreference resolution." *Bioinformatics (Oxford, England)*, vol. 28, no. 13, pp. 1759–1765, Jul. 2012.

[265] J. Björne and T. Salakoski, "Generalizing biomedical event extraction," in *BioNLP Shared Task 2011 Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 183–191.

[266] S. Riedel and A. McCallum, "Robust biomedical event extraction with dual decomposition and minimal domain adaptation," in *BioNLP Shared Task 2011 Workshop*, Portland, Oregon, USA, 2011.

[267] A. Vlachos and M. Craven, "Biomedical event extraction from abstracts and full papers using search-based structured prediction." *BMC bioinformatics*, vol. 13 Suppl 11, p. S5, 2012.

[268] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *The Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 951–991, 2003.

[269] H. H. H. B. M. van Haagen, P. a. C. 't Hoen, A. Botelho Bovo, A. de Morrée, E. M. van Mulligen, C. Chichester, J. a. Kors, J. T. den Dunnen, G.-J. B. van Ommen, S. M. van der Maarel, V. M. Kern, B. Mons, and M. J. Schuemie, "Novel protein-protein interactions inferred from literature context." *PloS one*, vol. 4, no. 11, p. e7894, 2009.

[270] R. Jelier, M. J. Schuemie, P.-J. Roes, E. M. van Mulligen, and J. a. Kors, "Literature-based concept profiles for gene annotation: the issue of weighting." *International journal of medical informatics*, vol. 77, no. 5, pp. 354–362, May 2008.

[271] L. Tari, S. Anwar, S. Liang, J. Cai, and C. Baral, "Discovering drug-drug interactions: a text-mining and reasoning approach based on properties of drug metabolism," *Bioinformatics (Oxford, England)*, vol. 26, no. 18, pp. i547–i553, Sep. 2010.

[272] J. D. Wren and H. R. Garner, "Shared relationship analysis: ranking set cohesion and commonalities within a literature-derived relationship network," vol. 20, no. 2, pp. 191–198, 2004.

[273] N. Kang, "Using natural language processing to improve biomedical concept normalization and relation mining," Ph.D. dissertation, Erasmus Medical Center, University of Rotterdam, 2013.

[274] T. C. Rindflesch and M. Fiszman, "The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text." *Journal of Biomedical Informatics*, vol. 36, no. 6, pp. 462–477, Dec. 2003.

[275] B. Settles, "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text." *Bioinformatics (Oxford, England)*, vol. 21, no. 14, pp. 3191–3192, 2005.

[276] Y. Song, E. Kim, G. G. Lee, and B. Yi, "POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, N. Collier, P. Ruch, and A. Nazarenko, Eds. Geneva, Switzerland: Association for Computational Linguistics, Aug. 2004, pp. 100–103.

[277] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning, and G. Sinclair, "Exploiting context for biomedical entity recognition: From syntax to the web," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, N. Collier, P. Ruch, and A. Nazarenko, Eds. Geneva, Switzerland: Association for Computational Linguistics, Aug. 2004, pp. 88–91.

[278] C.-N. Hsu, Y.-M. Chang, C.-J. Kuo, Y.-S. Lin, H.-S. Huang, and I.-F. Chung, "Integrating high dimensional bi-directional parsing models for gene mention tagging." *Bioinformatics (Oxford, England)*, vol. 24, no. 13, pp. i286–94, 2008.

[279] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[280] M. Collins, "Ranking algorithms for named-entity extraction: Boosting and the voted perceptron," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 489–496.

[281] A. Vlachos, "Tackling the BioCreative2 gene mention task with conditional random fields and syntactic parsing," in *Proceedings of the Second BioCreative Challenge Evaluation Workshop; 23 to 25 April 2007; Madrid, Spain*, 2007, pp. 85–87.

[282] M. Neves, M. Chagoyen, J. M. Carazo, and A. Pascual-Montano, "CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*. Columbus, Ohio: Association for Computational Linguistics, Jun. 2008, pp. 108–109.

[283] B. Carpenter, "LingPipe for 99.99% recall of gene mentions," *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, vol. 23, pp. 307–309, 2007.

[284] M. Vazquez, M. Krallinger, and F. Leitner, "Text Mining for Drugs and Chemical Compounds: Methods, Tools and Applications," *Molecular Informatics*, vol. 30, no. 6-7, pp. 506–519, 2011.

[285] A. P. Davis, C. G. Murphy, C. A. Saraceni-Richards, M. C. Rosenstein, T. C. Wiegers, and C. J. Mattingly, "Comparative Toxicogenomics Database: a knowledgebase and discovery tool for chemical-gene-disease networks." *Nucleic acids research*, vol. 37, no. Database issue, pp. D786–92, Jan. 2009.

[286] M. Colosimo, A. Morgan, A. Yeh, J. Colombe, and L. Hirschman, "Data preparation and interannotator agreement: BioCreAtIvE task 1B," *BMC bioinformatics*, vol. 6, no. Suppl 1, p. S12, 2005.

[287] H. M. Wain, M. J. Lush, F. Ducluzeau, V. K. Khodiyar, and S. Povey, "Genew: the human gene nomenclature database, 2004 updates," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D255–D257, 2004.

[288] S. I. Letovsky, R. W. Cottingham, C. J. Porter, and P. W. D. Li, "GDB: the human genome database," *Nucleic acids research*, vol. 26, no. 1, p. 94, 1998.

[289] I. Mani, Z. Hu, S. B. Jang, K. Samuel, M. Krause, J. Phillips, and C. H. Wu, "Protein name tagging guidelines: lessons learned," *Comp Funct Genomics*, vol. 6, no. 1-2, pp. 72–76, 2005.

[290] D. Rebholz-Schuhmann, H. Kirsch, and G. Nenadic, "IeXML: towards an annotation framework for biomedical semantic types enabling interoperability of text processing modules," in *Joint BioLINK and Bio-Ontologies SIG Meeting*, Fortaleza, Brazil, 2006.

[291] M. Torii, Z. Hu, C. H. Wu, and H. Liu, "BioTagger-GM: a gene/protein name recognition system." *Journal of the American Medical Informatics Association*, vol. 16, no. 2, pp. 247–255, Mar. 2009.

[292] L. Li, R. Zhou, D. Huang, and W. Liao, "Integrating divergent models for gene mention tagging," in *International Conference on Natural Language Processing and Knowledge Engineering*. Dalian, China: Ieee, Sep. 2009, pp. 1–7.

[293] J. Wilbur, L. Smith, and L. Tanabe, "Biocreative 2. gene mention task," in *Second BioCreative Challenge Evaluation Workshop*, Madrid, Spain, Apr. 2007, pp. 7–16.

[294] H. Kirsch, S. Gaudan, and D. Rebholz-Schuhmann, "Distributed modules for text annotation and IE applied to the biomedical domain." *International journal of medical informatics*, vol. 75, no. 6, pp. 496–500, 2006.

[295] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 142–147.

[296] D. Ferrucci and A. Lally, "UIMA: an architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering*, vol. 10, no. 3-4, pp. 327–348, 2004.

[297] H. Cunningham, "GATE, a general architecture for text engineering," *Computers and the Humanities*, vol. 36, pp. 223–254, 2002.

[298] Y. Kano, W. a. Baumgartner, L. McCrohon, S. Ananiadou, K. B. Cohen, L. Hunter, and J. Tsujii, "U-Compare: share and compare text mining tools with UIMA." *Bioinformatics (Oxford, England)*, vol. 25, no. 15, pp. 1997–1998, Aug. 2009.

[299] U. Hahn, E. Buyko, R. Landefeld, M. Mühlhausen, M. Poprat, K. Tomanek, and J. Wermter, "An overview of JCoRe, the JULIE lab UIMA component repository," in *Proceedings of the LREC Workshop: Towards Enhanced Interoperability for Large HLT Systems*, Marrakech, Morocco, 2008, pp. 1–8.

[300] E. Loper and S. Bird, "NLTK: the Natural Language Toolkit," in *Proceedings of the ACL Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*. Philadelphia, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70.

[301] J. Wermter, K. Tomanek, and U. Hahn, "High-performance gene name normalization with GeNo." *Bioinformatics (Oxford, England)*, vol. 25, no. 6, pp. 815–821, 2009.

[302] C. Jonquet, N. Shah, C. Youn, C. Callendar, M. Storey, and M. Musen, "NCBO annotator: semantic annotation of biomedical data," in *International Semantic Web Conference, Poster and Demo session*, Washington, D.C., WA, USA, 2009.

[303] a. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program." in *Proceedings of the AMIA Annual Symposium*, Washington, D.C., WA, USA, 2001, pp. 17–21.

[304] Y. Tateisi and J. Tsujii, "Part-of-speech annotation of biology research abstracts," in *4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004, pp. 1267–1270.

[305] N. Elhadad, *User-sensitive text summarization: Application to the medical domain.* PhD Thesis. Columbia University, Graduate School of Arts and Sciences, 2006.

[306] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," Internet Engineering Task Force, 2006.

[307] P. Stenetorp, S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, and J. Tsujii, "BRAT: a Web-based Tool for NLP-Assisted Text Annotation," in *European Chapter of the Association for computational Linguistics*, Avignon, France, 2012, p. 102.

[308] R. I. Doğan and Z. Lu, "An improved corpus of disease mentions in PubMed citations," in *Workshop on Biomedical Natural Language Processing, Association for Computational Linguistics*, Montréal, Canada, 2012, pp. 90–99.

[309] E. Pafilis, S. I. O'Donoghue, L. J. Jensen, H. Horn, M. Kuhn, N. P. Brown, and R. Schneider, "Reflect: augmented browsing for the life scientist." *Nature biotechnology*, vol. 27, no. 6, pp. 508–510, Jun. 2009.

[310] J.-J. Kim, P. Pezik, and D. Rebholz-Schuhmann, "MedEvi: retrieving textual evidence of relations between biomedical concepts from Medline." *Bioinformatics (Oxford, England)*, vol. 24, no. 11, pp. 1410–1412, 2008.

[311] T. Ohta, T. Matsuzaki, N. Okazaki, M. Miwa, R. Sætre, S. Pyysalo, and J. Tsujii, "Medie and Info-pubmed: 2010 update," *BMC bioinformatics*, vol. 11, no. Suppl 5, p. P7, 2010.

[312] P. Coppernoll-Blach, "Quertle: the conceptual relationships alternative search engine for pubmed," *Journal of the Medical Library Association*, vol. 99, no. 2, pp. 176–177, 2011.

[313] S. Matos, J. P. Arrais, J. Maia-Rodrigues, and J. L. Oliveira, "Concept-based query expansion for retrieving gene related publications from MEDLINE." *BMC bioinformatics*, vol. 11, p. 212, 2010.

[314] J. Hakenberg, D. Voronov, V. H. Nguyên, S. Liang, S. Anwar, B. Lumpkin, R. Leaman, L. Tari, and C. Baral, "A SNPshot of PubMed to associate genetic variants with drugs, diseases, and adverse reactions." *Journal of Biomedical Informatics*, vol. 45, no. 5, pp. 842–850, Oct. 2012.

[315] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[316] P. Thomas, M. Neves, I. Solt, D. Tikk, and U. Leser, "Relation extraction for drug-drug interactions using ensemble learning," *Training*, vol. 4, no. 2,402, pp. 21–425–23–827, 2011.

[317] B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, S. Roebuck, R. Tobin, and X. Wang, "Assisted curation: does text mining really help?" in *Pacific Symposium on Biocomputing*, Big Island, Hawaii, USA, Jan. 2008, pp. 556–567.

[318] N. Karamanis, R. Seal, I. Lewin, P. McQuilton, A. Vlachos, C. Gasperin, R. Drysdale, and T. Briscoe, "Natural language processing in aid of FlyBase curators." *BMC bioinformatics*, vol. 9, p. 193, 2008.

[319] A. Bairoch, "The future of annotation/biocuration," *Nature Precedings*, no. 713, 2009.

[320] D. Bolchini, A. Finkelstein, V. Perrone, and S. Nagl, "Better bioinformatics through usability analysis." *Bioinformatics (Oxford, England)*, vol. 25, no. 3, pp. 406–412, Feb. 2009.

[321] D. Salgado, M. Krallinger, M. Depaule, E. Drula, A. V. Tendulkar, F. Leitner, A. Valencia, and C. Marcelle, "MyMiner: a web application for computer-assisted biocuration and text annotation." *Bioinformatics (Oxford, England)*, vol. 28, no. 17, pp. 2285–2287, Sep. 2012.

[322] R. Rak, A. Rowley, W. Black, and S. Ananiadou, "Argo: an integrative, interactive, text mining-based workbench supporting curation." *Database : the journal of biological databases and curation*, vol. 2012, p. bas010, 2012.

[323] C.-H. Wei, H.-Y. Kao, and Z. Lu, "PubTator: a web-based text mining tool for assisting biocuration." *Nucleic acids research*, vol. 41, no. Web Server issue, pp. W518–22, Jul. 2013.

[324] F. Rinaldi, S. Clematide, G. Schneider, and M. Romacker, "ODIN: An Advanced Interface for the Curation of Biomedical Literature," 2010.

[325] D. C. Comeau, R. Islamaj Doğan, P. Ciccarese, K. B. Cohen, M. Krallinger, F. Leitner, Z. Lu, Y. Peng, F. Rinaldi, M. Torii, A. Valencia, K. Verspoor, T. C. Wiegers, C. H. Wu, and W. J. Wilbur, "BioC: a minimalist approach to interoperability for biomedical text processing." *Database : the journal of biological databases and curation*, vol. 2013, no. 0, p. bat064, 2013.

[326] E. Sayers and V. Miller, "Overview of the E-utility Web service (SOAP)," 2010.

[327] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, OBI Consortium, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. H. Scheuermann, N. Shah, P. L. Whetzel, and S. Lewis, "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration." *Nature biotechnology*, vol. 25, no. 11, pp. 1251–1255, Nov. 2007.

[328] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright, "NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information." *Journal of Biomedical Informatics*, vol. 40, no. 1, pp. 30–43, Feb. 2007.

[329] L. M. Schriml, C. Arze, S. Nadendla, Y.-W. W. Chang, M. Mazaitis, V. Felix, G. Feng, and W. A. Kibbe, "Disease Ontology: a backbone for disease semantic integration." *Nucleic acids research*, vol. 40, no. Database issue, pp. D940–6, Jan. 2012.

[330] S. Matos and J. L. Oliveira, "Classification methods for finding articles describing protein-protein interactions in PubMed." *Journal of Integrative Bioinformatics*, vol. 8, no. 3, p. 178, 2011.

[331] A. Mccallum, "Efficiently inducing features of conditional random fields," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, C. Meek and U. Kjaerulff, Eds. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., Aug. 2002, pp. 403–410.

[332] J. R. Finkel and C. D. Manning, "Joint parsing and named entity recognition," *North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pp. 326–334, 2009.