DISCOURSE CAUSALITY RECOGNITION IN THE BIOMEDICAL DOMAIN

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2014

By

Claudiu Mihăilă

School of Computer Science

Contents

Word Count: 55779

List of Tables

List of Figures

Acronyms and abbreviations

ADV adverb

CRF Conditional Random Field

DA dependent argument

DS different sentence

EDU Elementary Discourse Unit

EM Expectation Maximisation

FN false negative

FOL first order logic

FP false positive

GO Gene Ontology

IA independent argument

IAA inter-annotator agreement

idf inverse document frequency

IE information extraction

ILP inductive logic programming

LDA Latent Dirichlet Analysis

ME Maximum Entropy

MEMM Maximum Entropy Markov model

MK meta-knowledge

ML machine learning

NB Naïve Bayes

NE named entity

NER named entity recognition

NLM National Library of Medicine

NLP natural language processing

NP noun phrase

PAS predicate-argument structure

PDTB Penn Discourse Treebank

PMC PubMed Central

PMI pointwise mutual information

PoS part-of-speech

RBF radial basis function

RF Random Forest

RST Rhetorical Structure Theory

S sentence

SBAR clause

SBML Systems Biology Markup Language

SS same sentence

SSL semi-supervised learning

SVM Support Vector Machine

TM text mining

TP true positive

TSVM transductive support vector machine

UMLS Unified Medical Language System

V verb

VP verb phrase

Abstract

DISCOURSE CAUSALITY RECOGNITION
IN THE BIOMEDICAL DOMAIN
Claudiu Mihăilă
A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2014

With the advent of online publishing of scientific research came an avalanche of electronic resources and repositories containing knowledge encoded in some form or another. In the domain of biomedical sciences, research is now being published at a faster-than-ever pace, with several thousand articles per day. It is impossible for any human being to process that amount of information in due time, let alone apply it to their own needs. Thus appeared the necessity of being able to automatically retrieve relevant documents and extract useful information from text. Although it is now possible to distil essential factual knowledge from text, it is difficult to interpret the connections between the extracted facts. These connections, also known as *discourse relations*, make the text coherent and cohesive, and their automatic discovery can lead to a better understanding of the conveyed knowledge. One fundamental discourse relation is causality, as it is the one which explains reasons and allows for inferences to be made. This thesis is the first comprehensive study which focusses on recognising discourse causality in biomedical scientific literature. We first construct a manually annotated corpus of discourse causality and analyse its characteristics. Then, a methodology for automatically recognising causal relations using text mining and natural language processing techniques is presented. Furthermore, we investigate the automatic identification of additional information about the polarity, certainty, knowledge type and source of causal relations. The entire methodology is evaluated by empirical experiments, whose results show that it is possible to successfully extract causal relations from biomedical literature. Finally, we provide an example of a direct application of our research and offer ideas for further research directions and possible improvements to our methodology.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Sophia Ananiadou, who has provided me with both guidance and criticism throughout the most challenging times.

The friendship of the various members of the research group at NaCTeM, past and present, has made a difference to my PhD studies. Special thanks to Riza, George and Paul, the company of whom has made my journey more pleasant.

I need to say thank you to my family and friends and apologise for having neglected them recently. Mother, father, Claudia, thank you for your continuous support and encouragement, especially during the last few years. Alexandra, thank you for cheering me up and staying by me through the good times and bad. Monica, Khalil, Urgup, I appreciate the stimulating discussions, sleepless nights, and fun we have had over the last four years.

And last but not least, I am grateful for the three years of financial support offered jointly by EPSRC and the School of Computer Science, University of Manchester.

Publications

Intermediate results from this research have been presented and published in the following conference and journal articles.

1. **Claudiu Mihăilă**, Riza Theresa B. Batista-Navarro (2012). *What's in a Name? Entity Type Variation across Two Biomedical Subdomains*. In Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, pp. 38–45, Association for Computational Linguistics.

2. **Claudiu Mihăilă**, Riza Theresa B. Batista-Navarro, Sophia Ananiadou (2012). *Analysing Entity Type Variation across Biomedical Subdomains*. In Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2012), Istanbul, Turkey, pp. 1–7.

3. **Claudiu Mihăilă**, Tomoko Ohta, Sampo Pyysalo, Sophia Ananiadou (2013). *BioCause: Annotating and Analysing Causality in the Biomedical Domain*. BMC Bioinformatics, **14**:2.

4. **Claudiu Mihăilă**, Sophia Ananiadou (2013). *What Causes a Causal Relation? Detecting Causal Triggers in Biomedical Scientific Discourse*. In Proceedings of the Student Research Workshop at the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 38–45, Association for Computational Linguistics.

5. **Claudiu Mihăilă**, Sophia Ananiadou (2013). *Recognising Discourse Causality Triggers in the Biomedical Domain*. Journal of Bioinformatics and Computational Biology, **11**(6):1343008.

6. **Claudiu Mihăilă**, Sophia Ananiadou (2013). *A Hybrid Approach to Recognising Discourse Causality in the Biomedical Domain*. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2013, Shanghai, China, pp. 361–366, IEEE.

7. **Claudiu Mihăilă**, Sophia Ananiadou (2014). *The Meta-knowledge of Causality in Biomedical Scientific Discourse*. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. and Piperidis, S., ed.: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, pp. 1984-1991, European Language Resources Association (ELRA), 2014.

8. **Claudiu Mihăilă** and Sophia Ananiadou (In Press). *Semi-supervised learning of causal relations in biomedical discourse*. In BMC Medical Informatics and Decision Making.

Other work carried out as indirect part of this research has been presented and published in the conference and journal articles.

1. Riza Theresa B. Batista-Navarro, Georgios Kontonatsios, **Claudiu Mihăilă**[1], Paul Thompson, Rafal Rak, Raheel Nawaz, Ioannis Korkontzelos, Sophia Ananiadou (2013). *Facilitating the Analysis of Discourse Phenomena in an Interoperable NLP Platform*. In Computational Linguistics and Intelligent Text Processing, Springer Berlin Heidelberg, pp. 559–571.

[1] joint first author

2. Georgios Kontonatsios, Paul Thompson, Riza Theresa B. Batista-Navarro, **Claudiu Mihăilă**[1], Ioannis Korkontzelos, Sophia Ananiadou (2013). *Extending an interoperable platform to facilitate the creation of multilingual and multimodal NLP applications*. In Butt, M. and Hussain, S., ed.: Proceedings of the System Demonstration Session at The 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 43–48, Association for Computational Linguistics.

3. **Claudiu Mihăilă**[1], Georgios Kontonatsios, Riza Theresa B. Batista-Navarro, Paul Thompson, Ioannis Korkontzelos, Sophia Ananiadou (2013). *Towards a Better Understanding of Discourse: Integrating Multiple Discourse Annotation Perspectives Using UIMA*. In: Proceedings of the the 7th Linguistic Annotation Workshop and Interoperability with Discourse (LAW VII & ID), Sofia, Bulgaria, pp. 79–88, Association for Computational Linguistics.

Chapter 1

Introduction

This chapter provides an introduction to the research project. It begins with a description of the motivation and definition of the problem for this research. This is followed by an outline of research aims and objectives.

1.1 Motivation and problem definition

Human language is a complex, extremely powerful communication system. Whilst people have an amazing ability to communicate with one another, understanding natural language is a daunting task for computers. The main difficulty is that although natural language provides the ability to signal, it also enables its users to express an infinite number of new meanings. Natural languages are inherently ambiguous, and thus become a problem for computers which are not able to manage complex contextual situations.

Since the beginning of computational linguistics and natural language processing (NLP) it has been a known fact that scientific sublanguages exhibit specific properties that differentiate them from general language (Harris, 1968). These differences can be observed at various levels, such as vocabulary, semantic relationships and,

in some cases, even syntax (Grishman, 2001), and often require domain-specialised knowledge sources to aid in the performed analysis. For instance, Biber (1988) describes two distinctive syntactic characteristics of academic writing which distinguish it from general English. Firstly, in academic writing, supplementary information is most commonly integrated by modification of phrases rather than by the addition of extra clauses. Secondly, academic writing demands a greater effort from the reader by omitting non-essential information, through the frequent use of passivisation, nominalisation and noun compounding. They also show that these tendencies towards ”less elaborate and less explicit” language have become more pronounced in recent history.

Language is the medium in which, amongst others, health sciences education, research and practice operate. The language used in this domain, usually referred to as *biomedical language*, has also been studied from the sublanguage point of view. Some researchers focussed on differences at the semantic and syntactic levels, using predicate-argument structures (PASs) to formally describe *frames* for predicates (usually verbs, but also their participial and nominalised forms) and the roles of their arguments (parts of the sentence surrounding it) (Wattarujeekrit et al., 2004; Thompson et al., 2011a). For instance, Wattarujeekrit et al. (2004) analysed the PASs of a large number of verbs used in biomedical articles, and their results suggest that in some cases a significant difference exists in the predicate frames compared to those obtained from analysing news articles by the PropBank project (Palmer et al., 2005). Other studies address the differences between the language used in different biomedical subdomains (Lippincott et al., 2011; Mihăilă and Batista-Navarro, 2012; Mihăilă et al., 2012) to discover that detectable differences exist between them, varying according to the employed features. Their conclusion is that biomedical researchers need be aware of the importance of subdomain variation when considering the practical use of NLP applications. A similar study closely examines the differences not between biomedical and general languages, but between the abstracts and the body texts of scientific

biomedical research articles (Cohen et al., 2010). The two genres were found to differ significantly from structural, morpho-syntactic, semantic and discourse points of view, as well as in the performance of various text mining tools when applied on them.

Another problem is that, due to the rapid advances in biomedical research, scientific literature in this domain is being published at an ever-increasing rate (Verspoor et al., 2006). Without the aid of automated means, keeping abreast of recent developments within biomedicine would become difficult for researchers (Ananiadou and McNaught, 2006). Thus, it has become more and more important to be able to provide such automated, efficient and accurate means of retrieving and extracting user-oriented biomedical knowledge (Cohen and Hunter, 2008). Based on this need, biomedical text mining has seen significant recent advancements in the last years (Zweigenbaum et al., 2007), including named entity recognition (Fukuda et al., 1998), coreference resolution (Batista-Navarro and Ananiadou, 2011; Savova et al., 2011) and relation (Miwa et al., 2009) and event extraction (Miwa et al., 2012b,a). Additionally, bio-text mining tools have been included in specifically designed frameworks and systems, in which biomedical researchers can easily build workflows to extract information, e.g., U-Compare (Kano et al., 2009; Kontonatsios et al., 2013) and Argo (Rak et al., 2012), or create and curate pathways and link them to the literature, e.g., PathText (Kemper et al., 2010). Using biomedical text mining technology, text can now be enriched via the addition of semantic metadata and thus can support tasks such as analysing molecular pathways (Rzhetsky et al., 2004) and semantic searching (Miyao et al., 2006). On the one hand, text mining gives extra power to the searching mechanism, thus reducing the number of separate searches that have to be performed. On the other hand, it increases the relevance of the results that are returned by the search. In contrast to traditional search engines, text mining systems do not simply view documents as sequences of words, but rather they try to structure this information automatically. More specifically, instead of applying bag-of-words approaches, they try to find relationships

(or events) between words and phrases (or entities) within sentences, a process called event extraction.

Biomolecular events have recently received considerable attention as an important source of information in biomedical text mining (Tsuruoka et al., 2011; Ananiadou et al., 2010; Miwa et al., 2010), especially with three shared tasks dedicated to their extraction (Tsujii, 2009; Tsujii et al., 2011; Nédellec et al., 2013a). Once extracted, biomedical events can be used not only for returning appropriate results in response to searches, but also for the discovery of unknown facts.

Although event-based searching can retrieve many more relevant documents and distil essential factual knowledge from text than is possible using traditional keyword searches, the typical event representations (and the event extraction systems based on such representations) do not take into account all available information pertaining to the interpretation of the event, making it difficult to interpret the connections between extracted facts. New knowledge can be obtained by connecting the newly extracted events with already existing information. These connections, also known as *discourse relations*, make the text coherent and cohesive, and their automatic discovery can lead to a better understanding of the conveyed knowledge. They can be either explicit or implicit, depending on whether or not they are expressed in text using overt *discourse connectives* (also known as *triggers*). One of the fundamental discourse relations is causality, as it explains the functioning of ourselves, our environment and our interaction with it. This information can be leveraged by epidemiologists to identify patterns and predict disease outbreaks, health care professionals to provide personalised treatments based on patient history, etc. Nevertheless, causal relations pose two main difficulties when trying to recognise them, one regarding causal triggers, and the other regarding their arguments.

Firstly, causal triggers are both highly ambiguous and highly variable. Take, for

instance, the example (1.1) below, where the token *and* expresses causality[1]. However, in most other contexts, the same token has a non-causal meaning, denoting only a conjunction.

(1.1) SsrB binds within SPI-2 *and* activates SPI-2 genes for transcription.

This is the usual case with most closed-class part-of-speech words, such as conjunctions and adverbials. Other examples of trigger types more commonly used as causal triggers and belonging to open-class parts-of-speech are *suggesting*, *indicating* and *resulting in*. For instance, example (1.2) contains two mentions of *indicating*, but neither of them implies discourse causality.

(1.2) Buffer treated control cells showed intense green staining with syto9 (*indicating* viability) and a lack of PI staining (*indicating* no dead/dying cells or DNA release).

Furthermore, their variability leads to numerous ways of expressing the same causal trigger, due to the open-class properties of nouns and verbs. Take example (1.3), where the trigger *this result suggests that* indicates a causal relation.

(1.3) The hilE mRNA level measured by real-time PCR also revealed that hilE expression was increased in SR1304 by about 2-fold (Figure 3A).
This result suggests that Mlc can act as a negative regulator of hilE.

[1] All examples provided in this thesis are excerpts form the BioCause corpus, unless otherwise stated.

The same idea can be conveyed using synonyms of these words, such as *observation*, *experiment*, *indicate*, *show*, *prove* etc. The high variability reflects in obtaining a low recall, since there will be many false negatives (FNs).

With respect to the two arguments of a causal relation, cause and effect, they are more difficult to recognise than causal triggers, as we have previously reported (Mihăilă et al., 2013). Firstly, the spans of text that make up the arguments are of arbitrary length, varying significantly from one case to another. Arguments can go up to 100 tokens in length in the case of Cause, and up to 70 in the case of Effect.

Secondly, the position of the two arguments around the trigger can change. Although most of the relations follow a Cause-Trigger-Effect pattern, there is an important percentage of relations, 20%, which do not obey this rule. Furthermore, we showed that almost half of all relations have one argument in a different sentence than that of the trigger. Thus, the search space increases significantly and the difficulty of a correct recognition increases too.

This leads to the third issue, which concerns the distance between the trigger and the arguments. We illustrated the number of sentences between that of the trigger and that of the separate argument, when it is located in a different sentence. About half of the cases have the argument located in the previous sentence, but the rest spread up to the tenth previous sentence.

Considering the shortcomings that have been identified above, one possible solution is to analyse the characteristics of discourse causal relations in the biomedical domain and to create a methodology for their automatic recognition and extraction. This can be achieved by employing deep linguistic analysis methods specific to biomedicine.

In this thesis, we report on the research undertaken to evaluate the feasibility of

identifying discourse causality relations in biomedical text. We first present our annotation effort for enriching biomedical text with causality information. We then describe our approaches to train systems to recognise discourse causality automatically. Finally, we give one example from the many applications of discourse causality by creating natural language questions from causal relations.

1.2 Aims and Objectives

In what follows, we provide a brief description of the research project by outlining the main aims, objectives, hypothesis and evaluation measures.

1.2.1 Research Aims

The main aim of this thesis is as follows:

A_0 to investigate the use of NLP techniques in the task of recognising discourse causality in biomedical scientific literature.

In order to accomplish the main aim A_0, we have split it into more specific aims, marked as A_{S_n}, and define them as follows:

A_{S_1} to develop a methodology for the automatic recognition of biomedical discourse causality.

A_{S_2} to produce a ranking of the most relevant features which can recognise biomedical discourse causality.

A_{S_3} to develop a methodology for the automatic classification of meta-knowledge information about causal relations.

At this moment, it is necessary to emphasise two aspects which are not the subject of the current research study:

1. This study does not propose a new definition of causality, nor does it debate which definition is better. The literature contains multiple definitions of causality, depending on the field under study. Nevertheless, most of them have some overlapping core characteristics. We consider this set of characteristics as causality.

2. This research is limited to the biomedical scientific literature. Although there exists a large amount of work for open-domain discourse analysis, the biomedical domain (as well as many other specific domains) has been rather ignored. Due to the specificity of biomedical language, direct applications of open-domain methodologies are not suitable and vice versa.

Thus, considering the three specific aims of this study, the research questions addressed by this thesis are as follows:

RQ_1 how can natural language processing techniques be organised into a methodology that can identify the characteristics of biomedical discourse causality?

RQ_2 to what extent can discourse causality be automatically recognised in biomedical scientific literature?

RQ_3 to what extent can linguistic features be used to recognise discourse causality?

(a) to what extent can domain-independent features be used to recognise biomedical discourse causality?

(b) to what extent can domain-dependent features be used to recognise biomedical discourse causality?

(c) which are the features that are the most relevant to the task of automatically recognising biomedical discourse causality?

RQ_4 to what extent can the meta-knowledge of biomedical discourse causality be recognised?

RQ_5 can the task of recognising biomedical discourse causality improve on other NLP tasks, such as question generation?

1.2.2 Hypothesis

The research effort is being driven by the following main hypothesis:

H_0 Discourse causality in biomedical scientific literature exhibits significant and measurable differences, which can be captured through statistical and linguistic indicators.

To test the hypothesis, a framework which incorporates deep NLP techniques along with existing shallower techniques is proposed to improve the identification of biomedical discourse causality.

1.2.3 Research Objectives

To answer the research questions stated above, the following research objectives need to be met:

O_1 to develop a manually annotated corpus of biomedical scientific literature with relevant discourse causality information.

O_2 to develop a methodology that can recognise discourse causality in biomedical literature.

O_3 to identify useful features for recognising biomedical discourse causality.

O_4 to develop a manually annotated corpus of biomedical discourse causality with meta-knowledge information.

O_5 to investigate the automatic recognition of the meta-knowledge information of biomedical causal relations.

1.2.4 Research Evaluation

Three traditional evaluation methodologies will be followed for assessing the quality of resources produced in the course of this research:

1. the manual annotations performed by human experts are evaluated using inter-annotator agreement (IAA).
2. the performance of the models is evaluated using precision, recall and F-score.
3. the methodologies are evaluated based on the performance of their corresponding models.

1.3 Summary of Contributions

To summarise, the main original contributions of this thesis are as follows:

1. a methodology to investigate biomedical discourse causality;
2. a novel resource for discourse relation studies in the biomedical domain created as a by-product of this research: BioCause;
3. novel knowledge regarding biomedical discourse causality, being the first computational study which addresses this aim;
4. the first study which provides an analysis and ranking of the features able to recognise biomedical causal relations;
5. the first study which investigates the meta-knowledge of biomedical discourse causal relations;

To achieve this, an extensive review of the current research studies and approaches to the recognition of discourse causality in biomedical language has been undertaken.

The **first contribution**, the biomedical discourse causality investigation methodology, is situated in an inter-disciplinary context. It is a mixture of three main research areas: biomedicine, natural language processing, and machine learning. In Chapter 3, we present the necessary information with respect to the resources and tools that are employed in this research. The pipeline for processing the literature from raw text to extracted causal relations is described at an abstract level.

The **second contribution** represents the resource that has been compiled for these experiments: the Biomedical discourse causality corpus, BioCause. The scarcity of discourse relation resources for the biomedical domain is overcome by the compilation of a new corpus, assembled according to the needs of this project. The compilation process is detailed in Chapter 4, together with an in-depth evaluation of inter-annotator agreement.

Starting from the manually annotated BioCause, we were able to gain new insights into how causality is expressed in biomedical scientific literature. The **third contribution** provides detailed discussions on the characteristics of the corpus, causal triggers and causal arguments (in Chapter 4), as well as the empirical experiments undertaken to investigate the feasibility of recognising causal triggers (Chapter 5) and their arguments (Chapter 6).

From the output of these experiments follows the **fourth contribution**. Based on the results obtained from the numerous models created, we analysed all features independently to investigate their usefulness towards our task. The performance of each feature is presented in Chapters 5 and 6, according to which step in our pipeline they belong, showing their interaction with other features and the change in performance brought by their addition.

The **fifth contribution** regards the identification of additional information concerning causal relations. Knowing the polarity, certainty, knowledge type and source of causal relations is an important aspect that can dramatically change their interpretation. Chapter7 describes the manual annotation process, as well as the experiments performed to prove the viability of extracting such information.

1.4 Structure of the thesis

This thesis comprises nine chapters, in which the objectives of the research are followed systematically, and is grouped in three parts. Chapter 2 provides the background for the remaining chapters, offering an overview of causality and various perspectives of studying it. Chapters 3, 4, 5, 6 and 7 constitute the original contributions: guidelines for causality and meta-knowledge annotation, an annotated corpus, a corpus analysis, and the experimental results obtained. Chapter 8 represents the third part, evaluating an application of the work completed in the previous four chapters.

Chapter 2 gives an introduction to *causality*. This is achieved via a discussion of causality as studied in philosophy and general-domain natural language processing. The chapter then addresses causality in the context of biomedical text mining and natural language processing. The difference between the various definitions of causality is established, as each of the described efforts brings a new variation or focusses on a different aspect.

Chapter 3 reports on the methodology adopted in this thesis. It describes the pipeline we have devised for the creation of a full discourse causality parser. More specifically, it provides the high-level steps in recognising causal triggers, their arguments and disambiguating between the cause and the effect, as well as the core concepts of machine learning and evaluation measures employed in this research.

Chapter 4 describes the process of creating BioCause, the corpus that will be exploited in Chapters 5, 6, 7 and 8. BioCause is a new resource containing manual annotations of discourse causality across 19 full-text journal articles on infectious diseases. All details regarding the annotation, as well as detailed analyses of causal triggers, their arguments, and annotator disagreements, are included.

Chapter 5 describes the work on the identification of discourse causal triggers. The chapter begins with motivating the task, after which follows a detailed description of the employed features. We then present the experimental results, and discuss the impact of the engineered features, training corpus and algorithms.

Following the detection of triggers, in Chapter 6 we undertake the task of recognising their arguments. We split this task into three steps, each of them following a similar structure as the previous chapter. We describe the features, present the results and discuss various factors that influence them for each step individually.

Chapter 8 details one of multiple threads of research that can be extended from this work, involving both the biomedical discourse causality area and the natural language processing and generation areas. It contains a description of the rules that are used to create natural language questions generated from causal relation annotations, as well as a manual evaluation for these questions.

In Chapter 9, the concluding remarks of this research are reported. The chapter revisits the aims and the objectives of the thesis, and discusses to what extent these have been accomplished in the experiments that have been conducted. The thesis finishes by suggesting future directions of research.

Chapter 2

Causality

Causality, as a general semantic relationship, has been studied for many millennia, and in a multitude of completely different topics, such as Philosophy, Psychology and Linguistics. In contrast, causality in the biomedical domain has been scarcely studied. Most studies limit themselves to establish causal relations between entities in fine-grained contexts, such as disease-treatment or gene-protein. In this chapter we first provide details concerning causality and various theories that attempt to define it, both in the general and biomedical domains. This is followed by a review of the various efforts in the biomedical domain which capture some aspects of causality. Finally, we analyse the resources and methods that are currently used for automatic causality recognition, emphasising on their strengths and weaknesses.

2.1 Definitions of causality

Causality is the relationship that stands between one event (which is named the *cause*) and a second event (known as the *effect*), where the second event is understood as a consequence of the first.

Studies on causality have been performed in multiple fields. Therefore, different

theories exist in parallel in, e.g., Philosophy, Psychology, and Biomedicine. However, a consensus has not yet emerged between researchers regarding the definition of causality or its perception by humans. These theories are briefly described in the following subsections.

2.2 Causality in the general domain

Causality has been in the focus of philosophers for millennia, dating back as far as Aristotle in the Western philosophical tradition. Although several theories have been developed over the years, it still remains a recurring topic in contemporary Philosophy, Psychology, Linguistics etc.

Hobbs (1990), for instance, suggests that a large amount of contextual and external knowledge is required to interpret discourse. His proposed discourse relations are defined in terms of various types of inferences that have to be made in order to understand the text. There are four types of inferences, and, correspondingly, four coherence relations. Most relevant to our research is that which mentions that a discourse can be coherent since it describes coherent events in the world. The notion of coherent events refers to the fact that if one event is known, the other can be inferred provided that appropriate background knowledge is available. The relation that exists between these events is an OCCASION relation, which can be subdivided in either CAUSE or ENABLEMENT. Hobbs also proposes that these relations that exist in text must be organised into a tree structure in order to make the text coherent. Thus, the relations are defined recursively.

Mann and Thompson (1988) develop the Rhetorical Structure Theory (RST), which describes the organisation of natural text by characterising its structure in terms of relations that hold between two different parts of the text, named *nucleus* and *satellite*.

There are 78 fine-grained relations which are used in the annotation of the RST Discourse Treebank (RSTDT) corpus (Carlson et al., 2001). Usually, the elements of this very large set are grouped into 18 coarse-grained relations in order to reduce the complexity. However, each of the 18 relations can exist with several possible configurations for its arguments (or *nuclearity*), which results in 41 possible combinations. With regard to CAUSALITY, this relation has five subtypes: VOLITIONAL CAUSE, NON-VOLITIONAL CAUSE, VOLITIONAL RESULT, NON-VOLITIONAL RESULT, and PURPOSE. The differentiation between the first four subtypes is performed with the purpose of including both situations that are and are not intended outcomes of some action, as well as showing which roles are played by the nucleus and its satellite. The last subtype refers to yet unrealised situations, presented in the satellite, that can become reality through the activity described in the nucleus.

An interesting and important feature of RST, which also made it popular, is the fact that the relations are not mapped directly onto texts. Instead, they are part of more abstract structures, called schema applications, which are then connected to the text. By doing so, schema applications can be organised into a hierarchical system, a rhetorical structure tree, with the textual Elementary Discourse Units (EDUs) being located at the leaf level of the trees. This hierarchy is similar to that proposed by Hobbs (1990).

Regularity theories (Hume and Selby-Bigge, 1896) provide a widely used modern definition of causality. According to these theories, causality is a constant conjunction between events, associated with priority in time and contiguity in time and space. Basically, this theory states that one event causes another event if the latter follows the former and it is usual that the first event is followed by the second event. In this context, causality is thought to be asymmetric, imperfect, and indeterminate, and therefore is treated using probabilities.

In contrast, in the counterfactual theory (Lewis, 2001), causation is defined as *what*

would have happened if something were the case that in fact is not the case. Alternatively, $event_1$ causes $event_2$ only in the case when it is true that if $event_1$ had not occurred, then $event_2$ would not have occurred either.

Another theory, which is largely adopted, regards causality as a condition for the occurrence of an event (Sosa, 1975). Thus, causes are split into three categories:

1. Sufficient cause – if $event_1$ causes $event_2$, then the existence of $event_1$ implies the existence of $event_2$. Nonetheless, another $event_3$ may cause $event_2$ as well;

2. Necessary cause – if $event_1$ causes $event_2$, then the existence of $event_2$ implies the existence of $event_1$. However, the existence of $event_1$ does not guarantee the existence of $event_2$;

3. Insufficient but Necessary part of an Unnecessary but Sufficient (INUS) cause – $event_1$ contributes to the cause of $event_2$. However, the existence of $event_1$ does not guarantee the existence of $event_2$, and vice-versa. An example of this type of causation is a conglomerate of events, including $event_1$, which then implies the existence of $event_2$. Nevertheless, $event_2$ can be caused by other events or groups of events excluding $event_1$, and the existence of $event_1$ by itself does not guarantee the existence of $event_2$.

Knott and Sanders (1998) take another approach to define discourse relations in terms of cognitive primitives. There are four such basic notions, each with two values, which can be combined to form twelve classes of discourse relations.

1. BASIC OPERATION: discourse relations are either CAUSAL, where a 'relevant' causal connection exists between the spans, or ADDITIVE otherwise.

2. SOURCE OF COHERENCE: relations are either SEMANTIC, if the two spans are related in terms of their propositional content, or PRAGMATIC if they are related by their illocutionary force.

3. POLARITY: relations can be either NEGATIVE, if the operation links the content of one span to the negation of the content of the other span, or POSITIVE otherwise.

4. ORDER OF SEGMENTS: this is applicable only to *Causal* relations, where they are BASIC if the antecedent is on the left and NON-BASIC if it is on the right.

As can be noticed, CAUSALITY does not have a clear definition, and it is part of a set of cognitive primitives. Different combinations of the last three primitives create different kinds of causal relations, such as CLAIM-ARGUMENT (POSITIVE, PRAGMATIC and BASIC) or CONSEQUENCE-CAUSE (SEMANTIC, POSITIVE, NON-BASIC).

2.3 Causality in the biomedical domain

Causality has also been studied in the biomedical domain, in order to develop a basis for epidemiological research. Thus, it becomes easier to establish scientifically valid causal connections between potential disease agents and the multitude of diseases affecting humankind.

Bradford-Hill criteria are a set of nine minimal conditions which are necessary in order to provide adequate evidence for the existence of a causal relationship between an incidence and a consequence (Bradford-Hill, 1965). These were presented initially as a way of determining the causal link between a specific factor (e.g., cigarette smoking) and a disease (such as emphysema or lung cancer). The nine criteria are explained briefly in what follows.

1. Strength: the probability for the existence of a causal relationship between two events is directly proportional to the association between those two events, as measured by appropriate statistical tests;

2. Consistency: the association between the studied events must occur consistently when they are replicated in different settings and when using different methods.

3. Specificity: the probability of a causal relationship is increased if a single supposed cause is proven to produce a single specific effect. The lack of specificity does not, however, deny causality;

4. Temporality: in the timeline of events, the cause must necessarily always precede the effect, otherwise the causality relationship cannot be considered;

5. Biological gradient: if a dose-response relationship is present, it is strong evidence for a causal relationship; an increase or decrease in the exposure to the causal factor should be reflected in an increase or decrease, respectively, in the incidence of the effect;

6. Plausibility: the causal relationship must agree with currently accepted understanding of biomedical processes. Nevertheless, this might not be true in the case of new theories;

7. Coherence: the studied cause-effect association must not contradict, in a significant way, the current state of knowledge within its field and related fields;

8. Experiment: various experiments can be performed by taking prophylactic actions in the examined associations and observing whether the effect suffers alterations.

9. Analogy: in some cases, judging by the analogy with an already recognised causal relationship can prove beneficial.

However, none of these nine viewpoints is able to prove or deny indisputably the cause-effect hypothesis and, also, none can be required as a *sine qua non*. They give, in fact, with more or less strength, support for the causal relationship.

Nevertheless, the data in this work comes from scientific research articles, which have been analysed and reviewed by experts in their respective fields. Therefore, we consider that the information contained in the articles presents scientifically valid causal connections, the only remaining thing is distinguishing them from other relationships.

Amongst the large number of corpora that have been developed for biomedical text mining purposes, several include the annotation of statements regarding some type of causal associations, such as BioInfer (Pyysalo et al., 2007), GENIA (Kim et al., 2008) and GREC (Thompson et al., 2009). However, these corpora do not include an exhaustive coverage of causal statements. Furthermore, the granularity of the annotation of such statements is limited in several respects, which are described below. Since such corpus resources underlie most currently existing methods for the automatic analysis of biomedical text, there is an opportunity to advance the state of the art in domain-specific information extraction (IE) and text mining (TM) through the improvement of annotation schemata, resources and methods in the area of causal relation extraction.

2.3.1 Difference to the general domain

Mulkar-Mehta et al. (2011) state that only 11% of the biomedical causal connectives are found in football news and 12% of the causal connectives found in football news are found in biomedical publications. For instance, causal markers such as *inhibit*, *activate* or *induce* are specific to the biomedical domain, whereas causatives such as *edging* and *lifting* are found just in the football news domain. Common causal markers are limited in number, and are usually restricted to certain parts-of-speech. Most of these are conjunctions and prepositions, such as *after*, *because*, *for*, *when*, which are polysemous in nature. The only two common verbs found to denote causality are *lead* and *produce*.

2.3.2 Causality in biocuration efforts

General, non-specific physical causation is of obvious interest in biocuration efforts such as the assignment of Gene Ontology (GO) (Ashburner et al., 2000) terms to genes to characterise gene functions (Camon et al., 2004), in part because detailed molecular-level interactions are rarely known when a phenomenon is first observed. For example, an effect due to P_1 positively regulating the expression of P_2 through activation of a transcription factor of P_2 by catalysing its phosphorylation may be first observed, reported and curated simply as P_1 having a positive effect on the activity of P_2. Yet, general terms of causality such as "cause" rarely appear in biomedical domain ontologies or other formalisations of the ways in which entities, processes and events are associated with each other. Instead, such formalisations frequently apply terms such as "regulation", "stimulation" and "inhibition". Whilst such terms also carry specific senses in biology, their definitions in domain ontologies and use in biocuration efforts show that, typically, their scope effectively encompasses any general causal association.

The definitions of the Gene Ontology are good examples, due to the wide support of the ontology within the biocuration community, the large number of existing annotations and the adoption of the ontology definitions in prominent domain text annotation efforts. These definitions, included in Table 2.1, are broader than they may initially appear: they explicitly include indirect physical effects ("control of gene expression") without limitation on the length of the low-level causal chain and, through enumeration ("frequency, rate or extent"), effectively exhaust the ways in which a process can be affected by another. Specific cases can further illustrate the breadth of these definitions: GO terms such as REGULATION OF MULTICELLULAR ORGANISM GROWTH are used in curation efforts to capture such findings as the HDAC3 gene regulates the growth of humans – an indirect causal association across multiple levels of biological

GO ID	GO term	GO definition
GO:0050789	REGULATION OF A BIOLOGICAL PROCESS	Any process that *modulates* the frequency, rate extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule, or interaction with a protein or substrate molecule.
GO:0048518	POSITIVE REGULATION OF A BIOLOGICAL PROCESS	Any process that *activates* or *increases* the frequency extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule.
GO:0048519	NEGATIVE REGULATION OF A BIOLOGICAL PROCESS	Any process that *stops*, *prevents* or *reduces* the extent of a biological process. Biological processes are regulated by many means; examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule.

Table 2.1: Gene Ontology definitions of regulation, positive regulation and negative regulation of biological processes.

organisation that involves very complicated and only partially understood molecular pathways.

The GO definition of REGULATION OF BIOLOGICAL PROCESS is thus broadly equivalent to the explicitly comprehensive definition "any process that has any effect on another biological process". Furthermore, in a neutral biological context, the following pairs of statements are roughly synonymous according to the GO definitions:

"A affects B" $\rightarrow$ "A regulates B"

"A has a positive effect on B" $\rightarrow$ "A positively regulates B"

"A has a negative effect on B" $\rightarrow$ "A negatively regulates B"

and the following hold :

"A causes B" $\approx$ "A positively regulates B"

"A prevents B" $\approx$ "A negatively regulates B"

One should also consider the exact GO synonyms of positive regulation (e.g., UP REGULATION, UP-REGULATION, UPREGULATION OF BIOLOGICAL PROCESS and POSITIVE REGULATION OF PHYSIOLOGICAL PROCESS) and negative regulation (e.g., DOWN REGULATION, DOWN-REGULATION, DOWNREGULATION OF BIOLOGICAL PROCESS and NEGATIVE REGULATION OF PHYSIOLOGICAL PROCESS). Thus, whilst the observation that "causation" is rarely considered in general terms in domain curation, text annotation or IE, most of its scope covered in the many efforts that involve the general concept of regulation is physical causation.

2.3.3 Causality in pathway models

Pathway model curation is a specific biocuration task of particular interest to systems biology (Ghosh et al., 2011a). Pathway curation efforts seek to characterise complex biological systems involving large numbers of entities and their reactions in detail using formal, machine-readable representations. The Systems Biology Markup Language (SBML)[1] standard (Hucka et al., 2003) for pathway representation has been applied to a large number of curation efforts.

In particular, the SBML version used by the CellDesigner software[2] (Funahashi et al., 2008) has been adopted by major efforts, such as PANTHER[3] (Mi and Thomas, 2009). As such, the SBML/CellDesigner reaction semantics are of significant interest to domain IE efforts seeking to support automatic pathway curation.

SBML reactions are represented as typed associations of three sets of entities: reactants, products and modifiers. The base reaction types are normally specific biomolecular event/process types, such as binding or phosphorylation, and, thus, are out

[1] http://sbml.org
[2] http://celldesigner.org/
[3] http://www.pantherdb.org/

SBML/CellDesigner	GENIA
Catalysis	Positive regulation
Physical stimulation	Positive regulation
Modulation	Regulation
Trigger	Positive regulation
Inhibition	Negative regulation

Table 2.2: Comparison between SBML/CellDesigner reaction modifications and GENIA event types.

of scope for the study of general causality. However, SBML also allows the ways in which entities modify reactions to be characterised using specific types, summarised in Table 2.2, together with related GENIA event types (following Ohta et al. (2011a)). Some of the modification types (e.g., MODULATION and INHIBITION) are generic and used in practice to annotate general physical causal associations whose detailed molecular mechanisms may not be known.

2.3.4 Causality in biomedical corpora

A number of biomedical domain text annotation efforts include statements of general physical causality in their scope. The GENIA event corpus, the most widely adopted manually annotated domain resource for structured information extraction, adopts GO types and annotates statements of general causation using the types REGULATION, POSITIVE REGULATION and NEGATIVE REGULATION (Kim et al., 2008). Examples from the GENIA-derived annotation of the BioNLP shared task 2011 GE task corpus are shown in Figures 2.1 and 2.2. The GENIA event corpus annotation guidelines have been adapted also to a number of other tasks, such as in the annotation of the BioNLP shared task EPI and ID corpora (Ohta et al., 2011b; Pyysalo et al., 2011). An example from the ID corpus annotation is given in Figure 2.3.

Whilst other domain corpora with similar annotation targets have adopted different ontologies and annotation types, general causality is captured also in the annotation of

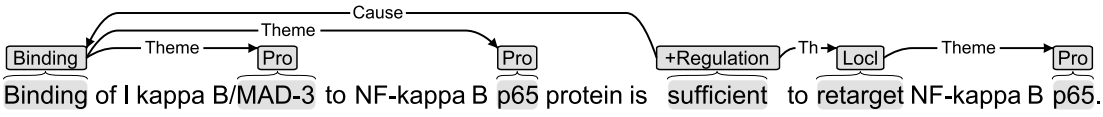


Figure 2.1: Example annotation from BioNLP shared task GE with annotation for general statement of causality ("is sufficient to").

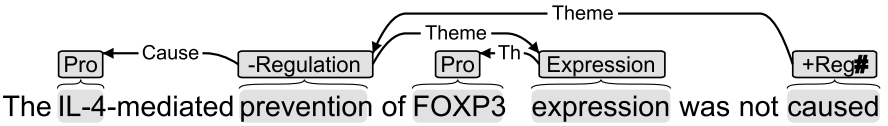


Figure 2.2: Example annotation from BioNLP shared task GE with annotation for general statements of causality ("prevention" and "caused").

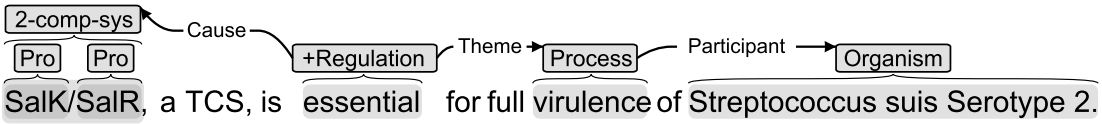


Figure 2.3: Example annotation from BioNLP shared task Infectious Diseases corpus with annotation for general statements of causality ("is essential for").

corpora such as BioInfer (Pyysalo et al., 2007) and GREC (Thompson et al., 2009). BioInfer applies an independently developed ontology that incorporates types capturing both the general positive-negative-unspecified distinction involved in GO and GENIA annotation, as well as more detailed subtypes capturing, e.g., the distinction between initiating a process and having a general positive effect on one (Figure 2.4). In contrast, the GREC corpus opts for an approach where only a small set of specific associations are assigned detailed types, with the majority being generically typed as GENE REGULATION EVENT (GRE). Nevertheless, the scope of this generic type extends to cover also general physical causal associations (Figure 2.5).

Thus, general physical causality is broadly included in the scope of many domain resources annotated with structured representations for information extraction. However, the scopes of these annotations do exclude a variety of statements potentially involving causal associations. Restrictions include limitation to specific forms of expression such as only verbal and nominalised forms, annotation of explicit statements only and exclusion of statements that only suggest possible causal connections ("A

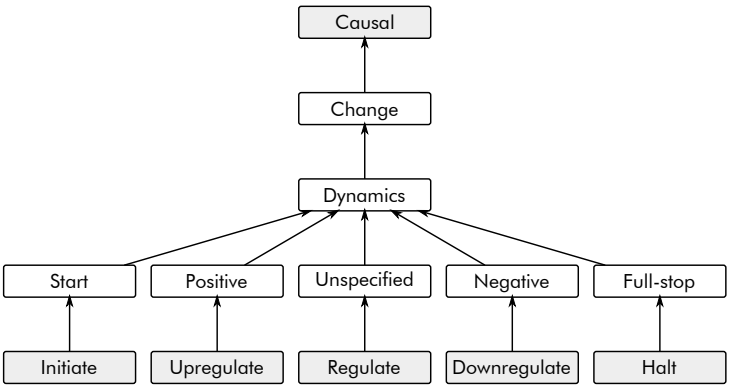


Figure 2.4: Fragment of the BioInfer ontology of causal associations involving change in process dynamics. Arrows correspond to IS-A relationships.

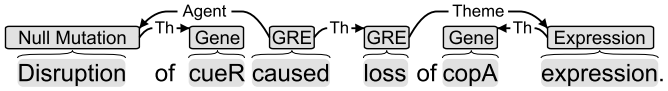


Figure 2.5: Example annotation from GREC corpus with annotation for general statement of causality ("caused").

happened after B"). Such limitations imply gaps between the full set of statements of interest and those annotated in domain resources and leave open a number of opportunities for further improvement of resources and tools for the analysis of causality in biomedical text.

Several other more discourse-oriented resources have also been created. The work most similar to ours is the BioDRB corpus (Prasad et al., 2011), which is a collection of 24 open-access full-text biomedical articles selected from GENIA, containing annotations of 16 types of discourse relations, one of which is causality. It was created by adapting the framework of the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), which annotates the argument structure, semantics and attribution of discourse relations and their arguments. The number of purely causal relations annotated in this corpus is 542. There are another 23 relations which are a mixture between causality and one of either background, temporal, conjunction or reinforcement relations. For machine learning purposes, this dataset is considered relatively small, as it might not

capture sufficient contextual diversity to perform well on unseen data. Thus, a detailed comparison and combining of this resource with the one described in this thesis represent an interesting opportunity.

2.4 Approaches to automatically recognising causality

Although prior studies in discourse causality extraction are relatively sparse, a large amount of work has been dedicated to generic discourse parsing and discourse relation identification in the general domain, where researchers have developed end-to-end discourse parsers. Most work is based on the PDTB (Prasad et al., 2008), a corpus of lexically-grounded annotations of discourse relations. Whilst there are many successful attempts in the direction of automatically classifying triggers, argument identification has been explored to a more limited extent. Furthermore, until now, comparatively little work has been carried out on causal discourse relations in the biomedical domain, although causal associations between biological entities, events and processes are central to most claims of interest (Kleinberg and Hripcsak, 2011).

There are several main aspects that need to be considered when attempting to recognise discourse causality. First, there is the problem of data. For machine learners to perform well, a significant amount of manually annotated text is necessary. The same applies for rule-based systems, as researchers need to develop the rules based on observations made on annotations. The second issue is, given any two spans of text, the recognition of the existence or inexistence of causal relations. This decision requires large amounts of background knowledge. The third aspect regards the identification of causal triggers. On the one hand, causal triggers are highly ambiguous, with some being used for multiple discourse relations or no discourse relation at all. On the other hand, they are highly variable, the same meaning being conveyed in numerous ways. Consequent to detecting triggers, another issue is the recognition of the

two arguments, namely *Cause* and *Effect*. Deciding which two spans of text around the trigger are connected causally is a very difficult problem, as the search space is very large. Furthermore, a significant amount of background knowledge is needed in order to establish which argument plays which of the two roles. In order not to go around these two steps of trigger and then argument detection, another approach is to decide directly on the existence of a causal relation between two (usually adjacent) spans of text. If not adjacent, there exists the problem of a very high number of combinations. Otherwise, a large number of relations can be missed if the two sentences are not adjacent. These aspects are discussed in detail in the following four subsections.

2.4.1 Acquiring data

Like in all NLP tasks, one of the most difficult problems to address is the acquisition of appropriate data. Whilst in some cases the creation of such data is relatively easy, requiring no expertise in a domain, extensive knowledge or thinking about interpretation, in discourse analysis the situation is different.

The most important problem when creating corpora for discourse analysis is the subjectivity of the annotators when interpreting the text they are annotating. A large part of this problem can be resolved with a number of training sessions, in which annotators discuss the annotation guidelines and agree on the interpretation of borderline cases.

In the case of biomedical discourse analysis, the problem is complicated by the fact that the information contained can be understood fully only by domain experts. Thus, it is important for the annotation to be performed by humans with extensive biomedical education and deep understanding of the subject. However, finding domain experts who are interested in performing tedious annotation tasks is very difficult. This leads to both an increased period of time needed for the annotation, and higher financial cost.

For example, the 24 full text articles in the previously described BioDRB corpus have been annotated over a period of three whole years (Prasad et al., 2011).

To overcome these problems, researchers have analysed the possibility of bootstrapping, in an unsupervised manner, a corpus of discourse relations. For instance, Marcu and Echihabi (2002) approach the task of disambiguating discourse relations in the general domain by employing automatic and unsupervised data acquisition. They focus on four relation types: CONTRAST, CAUSE-EXPLANATION-EVIDENCE, CONDITION and ELABORATION. As can be noticed, these relations are defined at a much coarser level of granularity than in most discourse theories, such as those previously described in Section 2.2. In fact, they decide to focus on these four categories as they are based on a common set of intuitions shared between the various theories. Another reason is that by having a coarse definition for discourse relations, both the complexity and ambiguity of the employed theory are reduced and, thus, would allow for computers to better generalise and ultimately recognise them in free text.

To create the corpus of bootstrapped discourse relations, they extract from a large collection of text (1 billion words, 41 million sentences) all adjacent sentence pairs which contain a specific cue phrase at the beginning of the second sentence. A similar process is applied to sentences which contain the cue phrases in the middle: the sentence is split into two spans, on each side of the cue phrase. The relations between the two spans of text (inter- or intra-sententially) are marked according to the cue phrase that was found. Examples of cues that are used are *but* and *although* for CONTRAST, *because* and *thus* for CAUSE, *if* and *then* for CONDITION, and *for example* and *which* for ELABORATION. However, this method has two significant problems. First, the size of the set of cue phrases is very limited. Second, the set of cue phrases is specifically chosen in order to minimise the number of resulting false positives. As it has been shown in previous research (Schiffrin, 1988; Marcu, 2000), some occurrences of *but*

Reference	Base alg.	Corpus	F-score
Marcu and Echihabi (2002)	NB	RSTDT	57%
Girju and Moldovan (2002)	Heuristics	TREC9	65.50%
Chang and Choi (2004)	NB	*TREC5	83.10%
Chang and Choi (2006)	EM	*cTREC5	77.37%
Blanco et al. (2008)	C4.5	*TREC5	90.45%
Pitler and Nenkova (2009)	NB	PDTB	94.15%
Subba and Di Eugenio (2009)	FOL+ILP	Instructional	62.78%
Subba and Di Eugenio (2009)	FOL+ILP	*Instructional	19.05%
Hernault et al. (2010)	ME	PDTB	10.80%
Hernault et al. (2010)	ME	*PDTB	2.60%
Do et al. (2011)	CEA	*PDTB	41.70%

Table 2.3: Approaches to relation classification.

and *because* do not have discourse functions. This percentage of cues without discourse function ranges between 15% and 20% of occurrences (Carlson et al., 2001). However, other discourse triggers can be more ambiguous than this.

2.4.2 Classification of discourse relations

Classifying discourse relations is an important step towards a more correct understanding of the meaning conveyed in text, and, subsequently, towards the improvement of several NLP tasks, such as question answering or automatic summarisation. A special focus has been given to causality and causal relations. Table 2.3 presents the most relevant work that has been performed in the task of classifying discourse relations. As most work focusses on discourse relations in general, we use an asterisk (*) to mark the causal subset of relations if results for these are available. Although there is research dating to the late 1980s, most initial work uses hand-made causality patterns, which function on restricted constructions and domains (Joskowsicz et al., 1989; Low et al., 2001).

Marcu and Echihabi (2002), whose dataset was described in the previous section, used inter-sentence word pair probability to discriminate between the four relation

types. Using Naïve Bayes (NB), an accuracy of 57% is obtained for extracting causality between sentences.

Girju and Moldovan (2002) investigate the recognition of causal relations between noun phrases (NPs) with the purpose of improving the performance of question answering. They filter causal triggers and create a ranking of five noun classes based on WordNet. Upon examination, a precision of 65.50% is computed, and later improved to 73.91% by using a decision tree classifier (Girju, 2003).

Chang and Choi (2004) move from the extraction of causal relation that consider lexical and syntactical patterns to a more advanced, co-occurrence-based method. In their approach, if two event pairs share some lexical pairs and one of the events is proven to be a causal relation, the causal probability of the other event increases. Events are represented as (S,V,O) ternary expressions, and a pre-defined list of cue-phrases, taken from Girju and Moldovan (2002), is used to filter causal events. A NB classifier then learns to distinguish between causal and non-causal event pairs, whilst an Expectation Maximisation (EM) classifier applies the NB model to a large raw corpus to re-estimate the parameters. The evaluation performed on a subset of the TREC5 corpus, selected to have the word *cancer* in each sentence, results in a maximum F-score of 83.10%.

Chang and Choi (2006) improve on this work by acquiring, in an unsupervised manner, new cue phrases for causality based on similar classifiers. The noise that was added to the model by the automatic cue phrase acquistion is reflected in a lower F-score when evaluated on the same dataset. However, the proposed method is attractive for cases where dictionaries and word-sense mappings are not available.

Blanco et al. (2008) propose a method for the detection of causal relations between a verb phrase (VP) and a subordinate clause. After manually classifying 1270 sentences from the TREC5 corpus into causal or non-causal, they extract the syntactic patterns from the 170 causation-encoding sentences. The machine learner classifies

one thousand sentences from SemCor 2.1 into causal or non-causal, based on an array of features. This includes the cue phrases themselves, their modifiers (e.g., adverbs like *long* or prepositions like *if*), the semantic class and tense of the potentially cause and effect verbs. The C4.5 decision trees classify the instances with an F-score value of 90.45%.

Looking at the four top-level classes of relations in the PDTB (Expansion, Comparison, Contingency – containing causality, and Temporal), Pitler and Nenkova (2009) attempt to automatically distinguish between them. Their NB classifier reaches a top F-score of 94.15%, which is the same as the human IAA observed in the creation of the corpus.

Presenting an innovative idea, Subba and Di Eugenio (2009) tackle the problem with a first order logic (FOL) learning approach combined with inductive logic programming (ILP). They motivate their decision linguistically: such models can leverage the rich compositional semantic data of the EDUs from VerbNet along with the structural relational properties of the text spans. The experiments are performed on a corpus of 176 instructional text documents, which has been manually split into 5744 EDUs. ILP is shown to surpass other machine learning (ML) classifiers, such as NB, Decision Tree, and RIPPER. The F-score of ILP reaches 62.78%, which is 6% higher than the second best, Decision Tree. Also provided by Subba and Di Eugenio (2009) are classification results for individual relation types. Causality is recognised with only 19.05% F-score, which is much less than the macro-average of 49.51%. In fact, causal relations are the second worst amongst all relation types as regards their correct recognition.

In an attempt to overcome the problem of data sparseness, Hernault et al. (2010) experiment with semi-supervised learning by creating a feature co-occurrence matrix from unlabelled data. Their proposed method computes the co-occurrence between each pair of features using unlabelled data by calculating the χ^2 of those two features

co-occurring in all feature vectors. Then this information is used to extend the feature vectors during both training and testing in case some features are not observed during training, thus reducing the sparseness in test feature vectors. No new features are introduced in this work, instead the method exploits features that have been previously shown to be discriminative for the task. Evaluated on the PDTB corpus, the system obtains results reaching a macro-average of 10.80% F-score and 2.60% F-score for causality only. However, the results are too low to be used as-is in discourse parsers.

Do et al. (2011) investigate causal relations that occur between events. They develop the concept of Cause-Effect Association to measure the causal associatedness of two events, based on combinations of pointwise mutual information (PMI), inverse document frequency (idf), distances and co-occurrences between event predicates, one predicate and the other's arguments, and between the arguments of the two predicates. The system using this new measure reaches an F-score of 41.70% in recognising causal relations between events.

2.4.3 Detecting causal triggers

Table 2.4 summarises the most relevant work that has been performed in the task of analysing discourse and discourse causality triggers. As can be noticed, there is no work which specifically addresses causal relations or which provides separate results for them. Furthermore, the amount of work addressing triggers in the biomedical domain is limited to only two recent studies.

Pitler and Nenkova (2009) are amongst the first researchers who have tackled the problem of identifying discourse connectives, but without determining the discourse relation, as a disambiguation task. Using almost only syntactic features related to the trigger, they achieve an F-score of around 91% when using automatic parses (result provided by Lin et al. (2012) for comparability reasons). The power of the employed

Reference	Base alg.	Corpus	F-score
Pitler and Nenkova (2009)	NB	PDTB	91.00%
Wellner (2009)	Reranking	PDTB	95.47%
Lin et al. (2012)	ME	PDTB	93.62%
Ibn Faiz and Mercer (2013)	ME	PDTB	95.81%
Ramesh et al. (2012)	CRF	BioDRB	75.70%
Ibn Faiz and Mercer (2013)	ME	BioDRB	82.36%

Table 2.4: Approaches to trigger detection.

features is impressive considering their simplicity, the syntax by itself reaching 88.19% F-score. In fact, their best result (94.19%, which is equivalent to the previously mentioned 91%) is obtained when using pair-wise interactions between features, such as trigger - syntax.

Basing their work on the one previously mentioned, Lin et al. (2012) have introduced new features and manage to slightly improve the overall performance. They included features related to the immediate context of the discourse trigger, such as the previous and next words, their part-of-speech (PoS) and syntactic interaction with the trigger itself. Also, they added as a feature the entire path from the connective to the root of the parse tree. Thus, the final F-score is 93.62%, with most error cases belonging to highly ambiguous triggers, such as *and*.

Another two approaches consider the syntactic constituency and dependency structure of the context of the trigger Wellner (2009). Features include the path from the trigger to the syntactic root, syntactic context features and conjunctive features in the case of the syntactic approach, whilst the dependency approach relies on features such as immediately neighbouring words and their part-of-speech, parents and siblings of the connective and clause detection.

Another small increase in F-score, with just under 1% over Pitler and Nenkova (2009) (considering their previously mentioned 94.19% F-score) and even less over Wellner (2009) is reached by slightly combining the surface level and syntactic feature

sets of these respective works on the PDTB corpus (Ibn Faiz and Mercer, 2013).

Regardless of the impressive amount and quality of work performed in the general domain, there exists a significant difference to the biomedical domain. As mentioned earlier, Mulkar-Mehta et al. (2011) have analysed the phrases that act as causal triggers in four different domains, i.e. financial news articles, football blog stories, football news articles, and biomedical scientific publications. Their investigation showed that there are only five common causal triggers amongst all four domains (*after*, *because*, *by*, *to* and *when*). However, these five triggers occur with different frequencies in the four domains and, furthermore, have different causal precision values. With regard to the biomedical domain, they discovered that only 11% of the biomedical causal triggers are found in football news, whilst 12% of football news causal triggers are found in biomedical articles.

Using the BioDRB corpus as data, Ramesh et al. (2012) have explored the identification of discourse connectives. Similar to work in the general domain, they do not distinguish between the types of discourse relations. Using mostly a set of orthographic features, they obtain the best F-score of 75.7% using Conditional Random Fields (CRFs), with Support Vector Machines (SVMs) reaching only 65.7%. These results were obtained by using only syntactic features, as semantic features were shown to lower the performance. Also, they prove that there exist differences in discourse triggers between the biomedical and general domains by training a model on the BioDRB and evaluating it against PDTB and vice-versa. Such cross-domain models reach a maximum F-score of 59.20%, which demonstrates the need of domain-specific models for biomedicine.

The same conclusions have been reached by Ibn Faiz and Mercer (2013), who manage to improve these results by around 6%. They notice that the automatic named entity recognition performed by ABNER (Settles, 2005) lowers the overall performance due to its orthographic features, which are already included in the feature set.

Reference	Base alg.	Corpus	F_1-A1	F_1-A2	F_1-Rel
Wellner and Pustejovsky (2007)	ME	PDTB	69.80%	90.80%	64.60%
Elwell and Baldridge (2008)	ME	PDTB	80.00%	90.20%	73.60%
Prasad et al. (2010)	Manual	PDTB	86.30%	-	-
Ghosh et al. (2011b)	CRF	PDTB	57.30%	79.10%	-
Ghosh et al. (2012)	CRF	PDTB	58.40%	79.30%	-
Lin et al. (2012)	ME	PDTB	47.68%	70.27%	40.37%
Xu et al. (2012)	ME	PDTB	50.48%	70.17%	48.66%
Stepanov and Riccardi (2013)	CRF	PDTB	57.26%	82.35%	-

Table 2.5: Approaches to argument detection. A1 represents the first argument, A2 is the second argument, and Rel is the relation as a whole.

2.4.4 Detecting causal arguments

Table 2.5 provides the list of the most relevant work that has been undertaken in the task of recognising discourse arguments. As can be seen from the table, all efforts are directed at the discourse in PDTB, whilst the biomedical domain is not investigated at all.

The research so far can be classified into two main categories on the basis of the model they employ. In the first category, researchers consider a single model discourse parsing method, where the two arguments are identified in a cascade of two sequential models (one for each argument). There is no distinction made between the location of the arguments, i.e., whether or not they are in the same sentence or not. Rather, a single model decides on both the position and span of an argument, but not for both arguments at the same time.

Ghosh et al. (2011b), for instance, design the argument detection as a cascade of decisions based on CRFs, trained on lexical, syntactic and semantic features. The system first identifies the second argument, with features including the surface expression of tokens, the syntactic category path from the root of the parse tree to the token, PoS tag, lemma, inflection, information about the main verb of the sentence, and whether the previous sentence starts with a connective. Furthermore, they add the relation that

the trigger has been assigned in the PDTB. Then the system proceeds to identifying the first argument, but adding to the feature set gold-standard second-argument labels for the tokens. The performance reaches 57.30% F-score for the first argument, and 79.10% for the second argument. Feature-wise, the trigger relation sense is deemed to be the most relevant, for both arguments, whilst the information regarding the previous sentence plays an important role for the first argument. Additionally, using lemmas combined with inflection information increases the performance more than when compared to their individual contribution.

In subsequent work, Ghosh et al. (2012) develop two constraint-based methods with the purpose of increasing the recall of their parser. There are five hard constraints, which refer to overgeneration (each argument must be located in only one sentence), undergeneration (each trigger must have exactly one argument of each type), inter-sentential second argument (the second argument must be in the same sentence as the trigger), post-sentential first argument (the first argument must be located in the same sentence as the trigger or in a prior sentence), and overlapping of arguments and triggers (arguments and triggers are disjunct textual spans). These constraints are assigned weights, with all being set to 1, except undergeneration, which is set to 2. Thus, the system tries to maximise the difference between the score assigned by the CRF classifier and the weight of violated constraints. The addition of these constraints increases the F-score in the case of the first argument by 1.1%, whilst for the second argument by 0.2%.

In the second category, a multiple model discourse parsing method splits the process into two steps. First, a decision is made on the position of the arguments (same sentence of different sentences), and then identifying the actual span of text composing the arguments. This method has been preferred in the literature, with much work following this process. Nevertheless, for the second step of identifying the text spans there are several approaches. Some researchers investigate the detection of the head of

the arguments (Wellner and Pustejovsky, 2007; Elwell and Baldridge, 2008). Prasad et al. (2010) limit their detection to identifying only the sentence in which the first argument is located. Actual argument spans have been detected as well, either as part of complete discourse parsers (Lin et al., 2012; Xu et al., 2012), or separately (Stepanov and Riccardi, 2013).

Wellner and Pustejovsky (2007) provide the first attempt at identifying the arguments of discourse triggers in the PDTB. However, instead of identifying the full argument extents, they have undertaken the restricted task of identifying the arguments' heads. They use Maximum Entropy (ME) rankers, combined with a reranking step to jointly select the two arguments of each trigger. By representing the arguments in this manner, their method reaches an accuracy of 74.20% on gold parses and 64.60% on automatic parses. To obtain these results, they employ an array of lexical, syntactic, dependency, constituency and trigger-based features. They conclude that dependency features are more informative than constituency features, a fact which is due to the reduced sparseness of the more compact way of representing syntax.

The results of Wellner and Pustejovsky (2007) have been later improved by Elwell and Baldridge (2008). The improvements refer to creating separate models which are tuned to specific triggers and trigger types. For instance, one model is created for subordinating conjunctions, another for coordinating conjunctions, and one for discourse adverbials. They base their decision on the observation that different types of triggers have a different behaviour towards their first arguments. For instance, subordinating and coordinating conjunctions are connected via syntactic constituency to their arguments, whilst discourse adverbials are not, since their arguments can be located anywhere in the prior discourse. Furthermore, they add new features related to the morphological properties of triggers and their arguments, additional syntactic features and an expanded context with the previous and following triggers. These extensions improve the F-score on automatic parses by 9%, to 73.60%.

Taking a different approach at recognising the more challenging first argument, Prasad et al. (2010) aim to predict the sentences in which it is located. Their baseline is the simple rule specifying that the first argument is located in the immediately previous sentence to that of the trigger, which results in 83.30% F-score. The algorithm involves a combination of cascaded heuristics that first filter potential candidates, then rank them based on coreference and finally evaluate the remaining candidates. The manual application of this algorithm shows a 3% improvement over the baseline.

Lin et al. (2012) develop the first full, end-to-end discourse parser for the PDTB. As part of it, the argument detection is split into three steps: deciding on their position (same sentence (SS) or different sentence (DS)), their nodes in the parse trees, and their textual spans. Finding the position of the arguments makes use of simple lexical features, such as the trigger itself and its left and right neighbours, as well as their PoSs. Under these circumstances, the performance reaches 92.09% F-score. Following this, an ME classifier assigns a triplet of probabilities to each node in the sentence parse trees, corresponding to three possible labels it can have: first argument, second argument, or none. Nevertheless, the arguments located in different sentences are not dealt with appropriately, and only a simple rule that labels the previous sentence as the first argument is used. Basing the decisions on mostly syntactic features, this module reaches an F-score of 82.60%. Finally, a tree subtraction module, proposed by Dinesh et al. (2005), computes the actual text spans of the two arguments by separating the tokens in the two subtrees according to the type of trigger: subordinating or coordinating conjunctions. The application of the last module results in an exact-match F-score of 40.37%, whilst the partial match is doubled, at 80.96%.

A similar approach is undertaken by Xu et al. (2012), who expand the context window of the trigger to identify the position of the arguments and their spans. In contrast, they first decide whether a node is a valid argument, and then decide its role from the three possible labels, first or second argument or none. This leads to a

statistically significant increase in F-score for the first argument, at 50.48%, whilst the second argument is detected with a slightly lower accuracy: 70.17% F-score. Again, they do not treat the inter-sentential case properly, employing the same simple heuristic as in the previous case.

Stepanov and Riccardi (2013) conducted a series of experiments in which they compare all of the previous approaches. They select the best of the two main categories that we mentioned earlier in order to increase the performance to 57.26% F-score for the first argument and 82.35% F-score for the second argument. Unfortunately, they do not report a relation F-score, where both arguments need to be identified to count as a correctly identified relation. Their method is an extension of that of Ghosh et al. (2011b), which used CRFs to extract the spans of arguments, by integrating the argument position detection and the immediately previous sentence heuristic for inter-sentential first arguments.

2.5 Summary

The purpose of this chapter is to highlight the main directions of research that exist in discourse analysis, whilst putting emphasis on the most relevant studies involved in the literature.

We first reviewed the literature regarding theories of discourse and discourse analysis. Together, these studies provide important insights into the structure of discourse and its coherence. Unfortunately, a common point between the various theories is that they lack an exact definition for causality.

Following this, we analyse causal relations in a biomedical context, where previous research has shown that causal triggers that are found in general language or other domains (e.g., sports, finance, news) are not applicable (Mulkar-Mehta et al., 2011).

Furthermore, we examined multiple curation and annotation efforts to create biomedical gold standard corpora. Although some of them include causal relations, they are limited in scope and usually involve only physical causation. The only exception is the BioDRB corpus, which contains, amongst others, a small number of causal discourse relations. Although causal relation bootstrapping has been attempted, the seed for the method needs to be manually provided, which already limits the range of discoverable relations. Additionally, a large number of false positives will be generated, which add noise to the created models.

As we have seen, most relation recognition techniques, causal or general, require some pre-existing lexical or syntactic patterns as a basis for creating models. However, according to Mulkar-Mehta et al. (2011), general language causal cue phrases are not applicable to the biomedical domain. Thus, data that is specific to biomedicine is necessary. Moreover, both in deciding the existence of causal relations between textual spans and extracting the arguments of causal triggers, previous research uses pair-wise similarity and WordNet as semantic relatedness resources. These are not completely suitable and transferable to the biomedical domain. One the one hand, there exists no biomedical version of WordNet, and research has shown its creation would be extremely challenging (Fellbaum et al., 2006) were its structure even suitable for the biomedical domain (Poprat et al., 2008). On the other hand, creating pair-wise word/-phrase similarities with regard to causation would prove to be not scalable. Within the fastly evolving biomedical language, new terms are coined daily, whilst others fall out of use, at higher rates than general language. Plus, the variability and ambiguity of terms is much greater in the biomedical domain. This highly volatile character of biomedical language would require the causal models to be continuously re-created.

The studies included in this review highlight the need for more biomedical gold standard data to capture a wider range of causal discourse relations in this specific domain.

Chapter 3

Methodology

The line of research reported in this thesis connects various concepts and techniques from different fields of study: it applies machine learning algorithms to the investigation of the nature of causality in the discourse of biomedical scientific language. As a result, it is necessary to briefly introduce the background concepts, tools and resources relevant to this research.

In this chapter, we outline the necessary information regarding the resources and tools used for this research. Since our studied hypotheses are related to the expression of discourse causality in biomedical scientific language, we need to use a corpus containing such annotations.

The chapter continues by describing the pipeline we have devised for the creation of a full discourse causality parser. It describes the high-level steps in recognising causal triggers, their arguments and disambiguating between the cause and the effect.

Finally, we outline the core concepts of machine learning and the machine learning algorithms employed in this research. We briefly describe machine learning and the types of machine learning that exist, and explain the main mathematical concepts of several algorithms pertaining to each type. We also introduce the employed machine learning frameworks.

3.1 Our approach

In order to understand the work described in subsequent chapters, we define several concepts used throughout this thesis. These refer to causal relations and their components, as well as the pipeline that we developed for automatic causal relation recognition.

3.1.1 Causality

Causality is a discourse relation which is very difficult to define, and most studies consider it as a fundamental, almost axiomatic relation. We will not attempt to provide a definition of causality. Instead, we rely on the existing definitions and theories and extract the commonalities between them.

Most causal relations are signalled by an explicit phrase, which tells humans that a causal link exists in discourse. These phrases, named from here on *triggers*, can take various surface expressions, ranging from single words to more than five words. They can belong to different parts-of-speech, and can form various syntactic structures.

Each trigger has an associated pair of text spans, its two arguments. According to their localisation relative to the trigger, arguments can be either SS arguments (example (3.1)) or DS arguments (example (3.2)).

(3.1) [Strains expressing the mutant PmrB proteins could express pbgP normally in response to the low Mg2+ signal], *indicating that* [mutations in residues of the periplasmic domain of PmrB do not impair the enzymatic activity of the cytoplasmic domain of the PmrB protein].

(3.2) [Indeed, the pKa of one of the glutamic acid residues of the regulatory protein

TraM is approximately7.7 in the folded protein].

Therefore, it is plausible that [protonation/deprotonation of one or more of the glutamic acids in the periplasmic domain of PmrB could occur at pH approximately 5.8].

Independent arguments can be located in another sentence anywhere in the text, therefore their recognition is more difficult, even for humans.

According to the syntactic relation to the trigger, an argument can be either a dependent argument (DA) or an independent argument (IA), as shown in example (3.3). Because the English language is a right-branching language, the dependent argument is usually located immediately after the trigger.

(3.3) [The low pH signal may also act synergistically with the low Mg2+ signal in vivo]$_{IA}$ *because* [Mg2+ deprivation alone is not sufficient to provide all the LPS modifications seen in Salmonella when present inside macrophages]$_{DA}$.

The argument that is located in a different sentence to the trigger is always the independent argument. When in the same sentence, the parse tree of the sentence needs to be analysed in order to distinguish between the two.

Furthermore, each argument plays one of the two roles in a causal relation, Cause or Effect. Example (3.4)

(3.4) [The levels of phosphorylated PmrA are determined by the balance of the autokinase + phosphotransferase activity of PmrB and PmrB's phosphatase activity towards phospho-PmrA]$_{Cause}$.

Require: a text T
Ensure: discourse causal relations in T
1: Identify all causal triggers in T
2: **for all** trigger t **do**
3: Label t as SS or DS
4: **if** t is SS **then** {arguments in same sentence}
5: Split sentence in clauses
6: Label the immediate right clause of t as DA
7: Label the rest of the sentence as IA
8: **else** {arguments in different sentences}
9: Label sentence of t as DepArg
10: Identify IndArg around the sentence of t
11: **end if**
12: Identify relation direction
13: **end for**

Figure 3.1: Pseudocode for identifying causal relations in the BioCause.

Thus, [PmrD may be necessary to ensure that the amount of phosphorylated PmrA is such to promote transcription of its regulated genes]$_{Effect}$.

3.1.2 Pipeline

The pseudocode for the causality recognition pipeline is shown in Figure 3.1. Similar to the annotation mechanism used by the experts who produced the BioCause corpus, we have split the recognition of causality into three major steps. In the first step, the annotators were given just the raw text T, which was then analysed to find causal triggers. Second, when a causal trigger was found, the annotators decided on whether its two arguments are in the same sentence or different sentence. In the former case, the clause syntactically depending on the trigger becomes the DA, whilst the rest of the sentence represents the IA. In the latter case, the sentence containing the trigger becomes the DA, whilst the IA is identified as one of the sentences around the trigger. Finally, in the third step, after both arguments are located, the annotator classifies the direction of the relation, that is which argument plays which of the semantic roles of

cause and effect.

3.2 Learning methods and tools

The choice of learning method and algorithm can significantly influence the performance of a classification task. In what follows, we briefly describe the main concepts behind the algorithms employed in this work.

3.2.1 Supervised machine learning

The goal of *supervised machine learning* is the ability to automatically infer a function from analysing labelled training data that can assign to a data point one label from a predefined set of labels. In an optimal scenario, this inferred function will be able to correctly determine the class label for unseen data points, via generalisation.

More specifically, let $\mathcal{D}$ be a set of n training data points of the form

$$\mathcal{D} = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, +1\}\},\ i = \overline{1,n} \tag{3.1}$$

where x_i is a p-dimensional feature vector for the i-th training instance, and y_i is either +1 or -1, indicating the class to which the point x_i belongs. A supervised learner must then find the function $g : X \to Y$ in a space of possible functions G that minimises the loss function $L : Y \times Y \to \mathbb{R}^{\geq 0}$. For a point $(x_i, y_i) \in \mathcal{D}$, the loss of predicting the class label $\hat{y}$ is $L(y_i, \hat{y})$.

Several supervised machine learning algorithms have been selected in this research. Their theory is briefly discussed in the following sections.

Decision Trees

Decision trees are one type of model used in machine learning, representing the decisions made by classifiers in its nodes. Each node inside the tree corresponds to one of the features in the dataset, and is used to make a binary classification. The tree is learned by splitting the initial dataset into smaller subsets based on an attribute evaluator. This process of splitting into smaller subsets is repeated recursively until either all instances in a subset all have the same label, or until the splitting does not add any more value to the predictions. Thus, the trees are induced in a top-down manner, making them a greedy algorithm.

There are many decision tree algorithms that have been created. In this work, we use C4.5 (Quinlan, 1993), one of the most popular algorithms. C4.5 functions on a simple recursive procedure, where the dataset is split for the feature that has the highest normalised information gain amongst all features.

In this work, we use the C4.5 algorithm as implemented in the Weka framework (Hall et al., 2009; Witten and Frank, 2005), where it is named J48.

Random Forest

Random Forests (Breiman, 2001) are an ensemble learning method that operate by constructing multiple decision trees during training and decide the label for each instance through a voting scheme amongst all the trees.

The multiple decision trees are built using a much smaller number of features than in the original dataset. However, each tree has a different subset of features, which are sampled randomly. Furthermore, each tree is trained and tested on a different subset of instances from the initial dataset. In the end, all trees vote for each instance and the majority decides on the final label. In this work, we use the implementation in Weka.

Support Vector Machines

Support vector machines (Cortes and Vapnik, 1995) perform binary classification by attempting to construct a p-dimensional hyperplane that optimally separates the data into two categories. This is achieved by transforming the data into a higher dimensional space by employing a kernel function, followed by solving the optimisation problem given in Equation (3.2).

$$\min_{w,\xi,b} \left\{ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{3.2}$$

subject to

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0,$$

$$i = \overline{1,n}$$

where x_i is the point to be classified, y_i is the label of point x_i, w is the feature weight vector, ξ_i is the error in the i-th instance, b determines the offset of the hyperplane from the origin along the normal vector w, and C is a constant representing the overall importance of errors.

In this work, we employ the LibSVM implementation (Chang and Lin, 2011), as, unlike the SMO implementation in Weka, it is more efficient, contains various SVM formulations and allows cross-validation.

Naïve Bayes

Naïve Bayes is one of the simplest probabilistic classification algorithms. It uses the Bayes probability model for predicting the class probabilities of inputs.

The candidate phrase t_i is classified into c_0, representing a non-causal phrase, or c_1, meaning a causal trigger. The class c^* of the candidate phrase t_i is computed as shown in Equation 3.3.

$$c^* = \arg\max_{c_j} P(c_j|t_i) = \arg\max_{c_j} \frac{P(c_j)P(t_i|c_j)}{P(t_i)} \tag{3.3}$$

In this work, we use the implementation in Weka.

3.2.2 Graphical models

Conditional Random Fields (Lafferty et al., 2001) are a type of discriminative undirected probabilistic graphical model. They are frequently used in pattern recognition and NLP, since they can predict a label for a single instance by taking into account the context of that instance, unlike ordinary classifiers.

The input to the CRF algorithm is a sequence of tokens of text. This can be provided by a pipeline of pre-processing methods taking raw text as input. Such a pipeline is described in the following sections. The CRF algorithm then finds the most probable label sequence y given an observation sequence x, as shown in Equation 3.4.

$$y = \arg\max_{y} P_\lambda(y|x) \tag{3.4}$$

where x consists of the sequence of tokens from the input text. The probability $P_\lambda(y|x)$ is calculated as shown in Equation 3.5.

$$P_\lambda(y|x) = \frac{1}{Z_x} \cdot \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, x, i)\right) \tag{3.5}$$

Each feature function $f_j(y_{i-1}, y_i, x, i)$ is assigned an individual learned weight λ_j and multiplied by it. All m weighted feature functions are summated for each item in the sequence, exponentiated and divided by the Z_x normalisation factor for all state

sequences.

An advantage of Conditional Random Fields is that they were designed to overcome the label bias problem, which had already been recognised in the context of neural network-based Markov models in the early 1990s. This occurs frequently in other graphical models, such as Maximum Entropy Markov models (MEMMs), where states with low-entropy transition distributions effectively ignore their observations.

In our research, we leveraged an existing implementation of CRF. More specifically, we employ CRFSuite[1], which provides fast training and tagging, simple data formats and state-of-the-art training methods.

3.2.3 Semi-supervised machine learning

Semi-supervised learning refers to the use of both labelled and unlabelled data for training. It contrasts with supervised learning, where the entire training data set is labelled. Other names are *learning from labelled and unlabelled data* or *learning from partially labelled/classified data*. Semi-supervised learning can be classified as either transductive or inductive.

One the one hand, a learner is said to be transductive if it only works on the labelled and unlabelled training data, and cannot handle unseen data. The early graph-based methods are often transductive. On the other hand, inductive learners can naturally handle unseen data. It is important to notice that, under this naming convention, transductive support vector machines (TSVMs) are in fact inductive learners, because the resulting classifiers are defined over the whole space and are therefore capable of handling unseen data. The name TSVM originates from the initial intention to having them work only on the observed data, although people now use them for induction as well.

[1] http://www.chokkan.org/software/crfsuite/

Whilst labels are hard to obtain, unlabelled data are abundant, and therefore semi-supervised learning is an appropriate method of reducing human labour and improving accuracy. Although the domain experts spend significantly less time in annotating and creating labelled data, a reasonable amount of effort needs to be invested in the designing of good models, features, kernels and similarity functions for semi-supervised learning. Such effort is more critical than for supervised learning, since the lack of labelled training data will affect the final performance.

In a semi-supervised learning setting, we modelled the problem as a self-training task. The main reason for including this method is the limited amount of existing gold standard data. Self-training has been previously used in NLP applications, such as word sense disambiguation (Yarowsky, 1995), identification of subjective nouns (Riloff and Wiebe, 2003) and emotions in dialogues (Maeireizo et al., 2004). Nevertheless, to the best of our knowledge, it has not been applied in discourse (causal) relation recognition.

Require: labelled data Λ, unlabelled data Υ, confidence threshold τ
Ensure: labelled data Λ
while $|\Upsilon| > 0$ **do**
  train model μ on Λ
  classify Υ using μ
  for all $x \in \Upsilon$ **do**
    if $\mu_{conf}(x) > \tau$ **then** {confidence of classification greater than threshold}
      $\Lambda \leftarrow \Lambda \cup (x, \mu(x))$
      $\Upsilon \leftarrow \Upsilon - x$
    end if
  end for
end while

Figure 3.2: Pseudocode for identifying causal relations using SSL.

The entire learning process is included in Figure 3.2, whilst a visual representation is depicted in Figure 3.3. We have started the learning process with a small amount of labelled data, Λ, for classifier training. This results in the creation of a classification model, μ. Then, the unlabelled data, Υ, is classified using μ. From these newly obtained

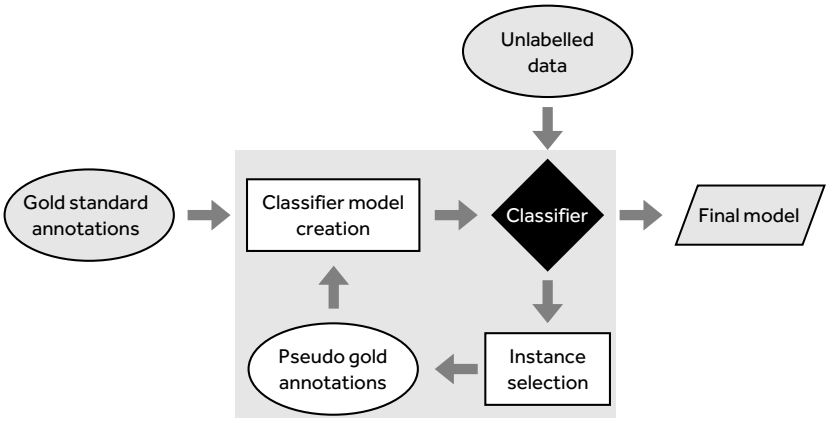


Figure 3.3: Self training approach

classifications, only those instances that have a classification confidence higher than a pre-set threshold τ are considered gold and are added to the labelled data as classified by μ. (Note that this might be different from the gold-standard label in the original corpus.) The rest are kept as unlabelled. If there are no instances that are classified with a confidence greater than τ, the model would come to a blocked state. Thus, we apply some simple heuristics to select several instances to be added to the labelled data. The process is repeated until all instances are classified.

3.3 Statistical and probabilistic methods for evaluation

This section describes briefly the main statistical concepts that are used throughout this work. We are interested in measuring how much human experts agree with each other in their decisions, as well as how well the models perform against gold standard data. Moreover, we look at statistical significance tests, as we want to know whether increases or decreases in performance occur by chance alone or not.

3.3.1 Inter-annotator agreement evaluation

Evaluating the agreement of multiple human annotators is an essential task to be performed after any annotation effort. This establishes both the quality of the resulting work and a possible upper-bound for the performance of automatic systems.

Cohen's kappa (κ) is a statistical measure of agreement between two annotators who each classify N items into C mutually exclusive categories (Carletta, 1996). Since κ takes into account the agreement occurring by chance, it is generally thought to be a more robust measure than simple percent agreement calculation. Kappa statistic is calculated as defined in Equation 3.6.

$$\kappa = \frac{P(a) - P(e)}{1 - P(e)} \tag{3.6}$$

where $P(a)$ is the observed agreement rate among annotators, and $P(e)$ is the estimated probability of the annotators agreeing by chance, using the observed data to calculate the probabilities of each observer randomly saying each category. If the annotators are in complete agreement, then $\kappa = 1$. If there is no agreement among the annotators, other than what would be expected by chance (as defined by $P(e)$), then $\kappa = 0$.

Both Landis and Koch (1977) and Fleiss (1981) suggest that κ values of 0 indicate no agreement, 0-0.20 slight agreement, 0.21-0.40 fair agreement, 0.41-0.60 moderate agreement, 0.61-0.80 substantial agreement, and 0.81-1 almost perfect agreement.

3.3.2 Performance evaluation

For performance evaluation purposes, we have used the standard metrics, namely precision, recall and F-measure. Precision measures the ratio of correct answers amongst those returned, whilst recall measures the ratio of correct answers amongst those that should have been returned. These two measures can be computed according to the

number of true positives (TPs), false positives (FPs) and FNs, which are counted by comparing the output of an automatic method (A) to a reference set of answers, usually gold standard data (R).

- TP represents the number of instances present in both the automatic answer and the reference data, i.e. $|A \cap R|$.
- FP represents the number of instances that are present in the answer, but not in the reference data, i.e. $|A - R|$.
- FN represents the number of instances that are not present in the answer, but are in the reference data, i.e. $|R - A|$.

Thus, precision is computed according to equation 3.7, whilst recall to equation 3.8.

$$P = \frac{TP}{TP + FP} \tag{3.7}$$

$$R = \frac{TP}{TP + FN} \tag{3.8}$$

Based on the above formulas for precision and recall, F-measure is defined as in equation 3.9.

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 \cdot P + R} \tag{3.9}$$

The constant β represents the weight balance between precision and recall. We have used $\beta = 1$ (and thus use the notation F_1), since we want to balance precision and recall. A $\beta = 2$ would give more weight to precision, whereas $\beta = 0.5$ gives more weight to recall.

By extension, F-measure can be computed directly from TP, FP and FN, as shown in equation 3.10.

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.10}$$

Due to its formulation, F-measure assigns equal weights to all instances in a classification task. However, in NLP, it is very common for the classification tasks to be performed on un-balanced datasets. This is due to the nature of language itself, where words or phenomena usually occur with a Zipfian distribution. The result of this is a bias of F-score towards the majority class. Computed in this setting, F-score is named *micro-average F-score*. To overcome this issue, it is possible to compute the *macro-average F-score*, which averages the precision and recall per class, and then computes the F-score on these two averages, as in equation 3.13.

$$P_{av} = \frac{P_i}{N} \tag{3.11}$$

where P_i is the precision for each class i, and N is the number of classes.

$$R_{av} = \frac{R_i}{N} \tag{3.12}$$

where R_i is the recall for each class i, and N is the number of classes.

$$maF_\beta = (1 + \beta^2) \cdot \frac{P_{av} \cdot R_{av}}{\beta^2 \cdot P_{av} + R_{av}} \tag{3.13}$$

3.3.3 Statistical significance

Statistical significance represents the probability that an effect occurs not due to just chance alone. Statistical hypothesis testing is used to determine the statistical significance of a result.

Student's t-test is a statistical hypothesis test in which the test statistic follows a Student's t distribution if the null hypothesis is supported. This test can be used to

decide whether two sets are or are not significantly different from each other, as shown in equation 3.14.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1X_2} \cdot \sqrt{\frac{2}{n}}} \tag{3.14}$$

where

$$s_{X_1X_2} = \sqrt{\frac{1}{2}(s_{X_1}^2 + s_{X_2}^2)}$$

The result of this t-test leads to a decision on the rejection or acceptance of the null hypothesis based on a pre-defined low probability threshold, called p-value, often coupled to a significance or alpha (α) level of 5%. If the p-value is found to be less than 5%, then the result would be considered statistically significant and the null hypothesis is rejected.

The analysis of variance (ANOVA) is a generalised form of the t-test, as it can be applied to more than two groups at the same time. This is better for testing three or more groups of observations since performing multiple t-tests results in a higher chance of committing type I errors[2].

3.4 Text pre-processing and NLP techniques

This section describes the text pre-processing techniques that are used in detecting discourse causality.

[2]A type I error is the incorrect rejection of a true null hypothesis, which is equivalent to a false positive.

3.4.1 Text pre-processing

The first step of pre-processing consists in sentence segmentation, which deals with splitting the raw text in the document into separate sentences, thus allowing sentence-by-sentence processing in the subsequent steps.

Since all subsequent processing is based on this task, it is necessary to obtain an accuracy as high as possible. Thus, six different sentence splitters were tested. These are listed in Table 3.1, together with specifications regarding whether they are statistical or rule-based, and whether they are designed for the biomedical domain or not.

Tool	**ML/Rule**	**BioMed**
Genia SS	ML	Y
LingPipe	ML	Y
OpenNLP	ML	N
RASP	R	N
NaCTeM	R	Y
UIMA	R	Y

Table 3.1: Sentence splitters.

The first three sentence splitters use machine learning models, whilst the last three contain rules to determine sentence boundaries. In contrast to the OpenNLP and RASP systems, which were designed for general language, the other four were built specifically for the biomedical domain, either with models obtained from the MEDLINE or GENIA corpora, or with rules encoding biomedical specificities.

GENIA Sentence Splitter[3] (Sætre et al., 2007) is a sentence splitter optimised for biomedical texts. The classification model is based on a supervised learning method using maximum entropy modelling, which is trained on the GENIA corpus. This sentence splitter outputs many false positive boundaries for abbreviated words, either generic (*e.g.*, *i.e.*, *Fig.*, *Dr.*, etc.) or biomedicine specific (*i.p. injection*, *hr.*). Moreover, it gives false negatives in the case of lowercase letter words that start the sentence

[3]http://www.nactem.ac.uk/y-matsu/geniass/

(*mTOR*, *pS6*, *eIF-4E*, *cDNA*, etc.).

LingPipe[4] is a toolkit for processing text using machine learners. The sentence splitter incorporated into LingPipe is trained with a MEDLINE sentence model. The small number of errors occur on abbreviations, such as *e.g.*, *i.e.*, *Fig.*, *Dr.*. Furthermore, LingPipe has errors due to missing closing parentheses and brackets in the source documents. However, these can be considered as not actual splitter errors, but human typing mistakes.

OpenNLP[5] is an open-domain machine learning based toolkit for the processing of natural language text. Errors occur very frequently in the sentence splitter included. Some of the errors are domain independent (false positives at abbreviations such as *Fig.*, *Dr.*, and false negatives in sentences ending in numbers), whilst other are specific to biomedicine (sentences starting with lowercase words *cDNA* or containing abbreviated words *granzyme B.*).

RASP[6] Briscoe et al. (2006) is another domain-independent, robust parsing system for English. Although it was not designed for a specific domain, the sentence splitter performs with a high accuracy in the biomedical domain. Since it is a rule-based system, the obvious problematic cases are abbreviations. It is very difficult to decide whether a sentence boundary follows an abbreviation or not. Erroneous cases include mainly units of measure (*hr.*, *ml.*, *mg.*, *rpm.*) and abbreviated biomedical entities (*granzyme B.*, *cyclin A.*, *antifarm A.*, *DNAse I.*).

The sentence splitter developed at NaCTeM[7] employs heuristic rules for identifying boundaries of sentences and paragraphs. However, it fails to recognise sentences which start with lowercase words or words beginning with Greek characters (β-*casein*), as well as several common abbreviations, such as *Fig.* or *Drs.*

[4]http://alias-i.com/lingpipe
[5]http://incubator.apache.org/opennlp/
[6]http://ilexir.co.uk/applications/rasp/download/
[7]http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector

UIMA[8] contains a rule-based component which is dedicated to sentence splitting. Like NaCTeM's splitter, it misses sentence boundaries if the sentence starts with lowercase or Greek characters and numbers, along with general abbreviations, such as *St.*, *vs.*, *Co.* and *Drs.*

Table 3.2 presents the performance of the six sentence splitters against a gold corpus of 7829 sentences. The performance is expressed as the rate of TP, standing for correctly identified sentence boundaries, FP, i.e. erroneous sentence boundaries, FN, corresponding to missed sentence boundaries, and F-score. For our final sentence splitting, we have considered the intersection of the six splitters as correct, and all differences in splitting have been manually treated.

Tool	**TP**	**FP**	**FN**	**F_1**
Genia SS	97.46%	4.44%	2.54%	96.54%
LingPipe	98.62%	0.66%	1.38%	98.97%
OpenNLP	94.43%	3.46%	5.57%	95.44%
NaCTeM	95.78%	4.59%	4.22%	95.60%
RASP	98.68%	0.77%	1.32%	98.95%
UIMA	95.98%	2.97%	4.02%	96.48%

Table 3.2: Performance of sentence splitters.

Following sentence splitting, the process of tokenisation determines token boundaries, such as words, numbers and punctuation in text. In biomedical text, this step is much more complicated than in general language text due to biomedical jargon. Example (3.5) shows a tokenised sentence. As can be noticed, the array of characters *2,3,4,9-Tetrahydro-1H-β-carboline* is a single token, although it contains different types of characters, including Latin letters, Greek letters, numbers, commas and hyphens, whilst *1,2,3,4-Tetrahydro-9H-pyrido[3,4-b]indole* contains brackets too. Although in general language the change in the type of character usually means a token boundary, in biomedical text this assumption is false. Additionally, the length of a

[8]http://uima.apache.org/index.html

token is much larger in biomedical text, therefore a model trained to consider length in general language will perform worse on biomedical text.

(3.5) [2,3,4,9-Tetrahydro-1H-β-carboline] [acid], [also] [known] [as] [1,2,3,4-Tetrahydro-9H-pyrido[3,4-b]indole], [is] [a] [natural] [organic] [derivative] [of] [β-carboline.].

In this research, we use the GENIA tokeniser (Tsuruoka et al., 2005) trained on MEDLINE.

3.4.2 Shallow NLP pre-processing

Shallow NLP techniques help in the analysis of morphological features of text, not providing syntactic or semantic information of any kind. These pre-processing steps are performed by the Enju parser (Miyao and Tsujii, 2008), which implicitly employs the GENIA tagger models trained on MEDLINE.

Part-of-speech tagging deals with assigning grammatical tags, such as *noun* and *verb*, to each token, as in example (3.6). This is mostly helpful in generalising in the cases where synonyms are used, since the grammatical category does not usually change.

(3.6) Acid[NN] activation[NN] of[IN] the[DT] two-component[JJ] regulatory[JJ] system[NN] of[IN] Salmonella[NNP] enterica[NNP]

Another important step is lemmatisation, which is the process of transforming the words into their dictionary base forms. This generalises the texts, since inflected verbs

and nouns are all normalised to the same value, their lemma. For instance, both *activates* and *activated* are changed into *activate*.

3.4.3 Deep NLP pre-processing

Deep NLP techniques are used to extract and analyse syntactic and semantic information from text. Superficial techniques, like those previously mentioned, cannot capture complex details needed in order to be able to discover causal relations and interpret discourse.

Extracting dependency relations from text returns syntactic relationships between pairs of words in a sentence. These are useful in cases where the syntax of a sentence is changed due to rephrasing, as the dependency relations should remain the same.

Syntactic constituency extraction allows the comparison of n-grams at a syntactic level and at multiple levels in the parse tree. Unlike simpler n-gram comparison that considers only exact words, this method generalises very well to capture sequences of syntactic categories. Both these relations are extracted with the help of the Enju parser.

However, syntactic analysis cannot capture information about the meaning of words in their context, and semantic analysis is necessary. Named entity recognition is the task of identifying and extracting named entities from text. Named entities are less likely to be replaced by others in the same context, whilst the syntax and functional words can change to reflect the same meaning. However, synonyms of named entities can be used and a mapping to unique identifiers is necessary for disambiguation purposes. Therefore, by analysing a large number of types of named entities, it is possible to produce significant relations between spans of text, including causality.

There exist a large number of automatic named entity recognisers, most of which are trained on specific classes of entities. For instance, OSCAR (Corbett and Copestake, 2008; Jessop et al., 2011) works mostly in the Chemistry domain, recognising

chemicals, reactions etc., whilst MetaMap[9] maps concepts to the Unified Medical Language System (UMLS) vocabulary. A wide array of named entity information can be obtained by applying several such systems, e.g., MetaMap, OSCAR, NeMine (Sasaki et al., 2008) and Europe PMC[10], so all these will be used in our methodology.

The named entities that have been recognised in the previous step can be further leveraged in order to extract events between them. For instance, one might be interested in the activation of particular genes under specific conditions, or the mechanism of dysregulation of apoptosis in cancer. This type of knowledge can be produced by EventMine (Miwa et al., 2012b), a machine learning-based pipeline system, that deals with extracting biomedical events from documents that are already annotated with various named entity information, such as genes and proteins. Given appropriate training data, EventMine can be trained to extract many different types and structures of events. The core system consists of four detection modules, which operate on the output of syntactic parsers.

3.5 Summary

This chapter described the general framework for our proposed discourse causal relation recognition in the biomedical domain. Additionally, we have reviewed and briefly explained some concepts that will be used throughout this work. Some core concepts of machine learning have been presented, showing the how the process of learning occurs. Moreover, we listed and defined the evaluation metrics that are used in this analysis. The chapter concluded with a description of both shallow and deep NLP pre-processing techniques that are incorporated in the causality detection framework.

[9] http://metamap.nlm.nih.gov/
[10] http://europepmc.org/

Chapter 4

BioCause

This chapter provides an overview of the process through which the BioCause corpus has been created. It starts with a description of the data source selection for the annotation effort and the experimental justification of selecting a single biomedical subdomain. We undertake the first analysis of the effectiveness of semantics alone in distinguishing biomedical subdomains and show that classifiers trained on named entity types perform very well in identifying the subdomain of an article. The corpus used for these experiments and analysis is not related to the Biocause corpus.

We then provide a description of the employed annotation scheme for the creation of BioCause and the training of the annotators. This is followed by detailed discussions on the characteristics of the corpus, causal triggers and causal arguments, together with an in-depth evaluation of inter-annotator agreement. Finally, a brief comparison of the annotation results between BioCause and the BioDRB is included.

4.1 Data source for BioCause

There are three main issues that need to be considered in order to select appropriate data for manual annotation and further using this annotation for the training of

automatic causality recognition systems. These relate to dissimilarities between biomedical sublanguages, the interaction of discourse annotations with other semantic mark-up, and the differences between abstracts and full-body texts. These issues are all discussed in what follows.

First, past work has shown that there are significant differences between various biomedical sublanguages at the levels of syntax and shallow discourse structure (Lippincott et al., 2011). Therefore, we extend this line of research and show in Section 4.2 that this hypothesis holds true even in the case of deeper semantics, such as named entity types. The documents in the corpus used for this experiment are not related to those in the BioCause corpus, but merely form the basis for the experiments supporting our decision. This is due to the fact that discourse relations, including causality, inherently belonging to semantics, are closely related to the named entities and events present in text. For instance, whilst in a disease subdomain causality would exist between pathologic agents, diseases, symptoms and drugs, in a pharmacological domain these relations would connect various chemical molecules, chemical reactions, or side effects, to name a few. We can conclude that linguistic observations at the lexical, syntactic and semantic levels made on one sublanguage may not necessarily be valid on another. Thus, we believe that attempting to train a machine-learning causality detection system on a mixture of subdomains, especially when the amount of manually annotated data is limited, would be detrimental to the learning process. Although we recognise that this choice is associated with high domain specificity, it is preferable to obtain a higher performance in a specific subdomain than a lower performance in a more general domain or a mixture of subdomains. Nevertheless, considering these differences, switching to a different subdomain should be simply a matter of re-training machine learners and re-creating the causality model. One can extend existing causality models by adding features that have not been encountered before. These would most probably be semantic features, such as a new typology for named entities and events, since these

are specific to subdomains.

Second, discourse causality, as a semantic relation, is inherently dependent on the named entities and events that are present in text. Hence, the errors in automatically recognising these essential bio-annotations can propagate in the processing pipeline and reflect negatively in the overall performance of recognising causality relations. Although the recognition of certain named entity types, such as proteins and genes, can currently reach performances of over 90% F-score, other named entity (NE) types and most event types are still at an unreliable level for inclusion in pipelines of discourse parsing. For instance, events can now be correctly identified with 50-55% F-score, depending on the complexity of each type of event (Nédellec et al., 2013a). Therefore, in order to isolate the task of recognising causality from that of recognising entities and events, gold standard named entity and event annotations are required. The effect of automatic named entity recognition (NER) and event extraction can be subsequently studied by replacing the gold standard annotations with automatic ones.

Finally, previous research has shown that although the information density is highest in abstracts, information coverage is much greater in full texts than in abstracts. Thus, these may be a better source of biologically relevant data (Schuemie et al., 2004; Shah et al., 2003). Therefore, it is important to develop a resource comprising full text articles in which to annotate discourse causality and extend previous work by analysing the distribution of causal relations between abstracts and text bodies.

For all of these three reasons, the causality annotation for BioCause is added on the top of existing event annotations from the BioNLP Shared Task (ST) on Infectious Diseases (ID) (Pyysalo et al., 2011). Whilst in other document sets, such as in those used for subdomain analysis (Mihăilă et al., 2012), entity and event annotations are automatically created by NER and event extraction systems such as NERsuite[1] or

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

EventMine (Miwa et al., 2010), the BioNLP ST ID task has annotations created manually by biomedical experts with experience in annotation efforts. Furthermore, the BioNLP ST ID corpus has a large size (19 documents, ̃100K words) and is comprised of full-text journal articles pertaining to a specific topic – infectious diseases.

4.2 Subdomain analysis

Whilst a multitude of tools and resources has been introduced in domain-specific NLP efforts for the recognition of entity mentions in text, a high proportion of these was trained and evaluated on popular corpora such as BioInfer (Pyysalo et al., 2007), GENETAG (Tanabe et al., 2005), GENIA (Kim et al., 2008), and PennBioIE (Kulick et al., 2004), as well as shared task corpora from BioCreative I, II, III (Arighi et al., 2011) and BioNLP 2009, 2011 and 2013 (Kim et al., 2011; Nédellec et al., 2013b). Most of these corpora consist of documents from the molecular biology subdomain. However, previous studies have established that different biomedical sublanguages exhibit linguistic variations.

The work of Harris (1968) introduced a formalisation of the notion of *sublanguage*, which he defined as a subset of general language. According to his theory, it is possible to process specialised languages, since they have a structure that can be expressed in a computable form. Several subsequent works on the study of biomedical languages have substantiated his theory, including the work of Sager et al. (1987) on pharmacological literature and lipid metabolism, and that of Friedman et al. (2002) analysing the properties of clinical and biomolecular sublanguages.

Taking a different angle, Stetson et al. (2002) uncovered the differences between "signout" notes and other medical notes (e.g., ambulatory clinic notes and discharge summaries) in terms of three aspects: discourse length, abbreviation use and abbreviation ambiguity. Based on their findings, "signout" notes are shorter and use a higher

number of less ambiguous abbreviations.

Verspoor et al. (2009) measured the lexical and structural variation in biomedical Open Access journals and subscription-based journals, concluding that there are no significant differences between them. Therefore, a model trained on one of these sources can be used successfully on the other, but only as long as the subject is maintained. Furthermore, they compare a mouse genomics corpus with two reference corpora, one composed of newswire texts and another of general biomedical articles. In this case, unsurprisingly, significant differences are found across many linguistic dimensions. Relevant to our study is the comparison between the more specific mouse genome corpus and the more general biomedical one: whilst similar from some points of view, such as negation and passivisation, they differ in sentence length and semantic features, such as the presence of various named entities.

These experiments, in contrast, investigate the differences and similarities between any two of twenty biomedical sublanguages at the level of named entities. Examining the distributions of different named entity types across several categories, our work is subtly similar to that of Cohen et al. (2010), who looked at the distributional variations of semantic classes in their effort to characterise the differences between abstracts and full texts. Four semantic classes, namely, *Gene*, *Mutation*, *Drug* and *Disease*, were taken into account in their study. Except for *Gene*, significant differences in terms of densities per thousand words have been observed between abstracts and full texts.

Also relevant is the work of Lippincott et al. (2011) in which a clustering-based quantitative analysis of the linguistic variations across 38 different biomedical sublanguages was presented. They investigate four dimensions relevant to the performance of NLP systems, i.e. vocabulary, syntax, semantics and discourse structure. With regard to semantic features, the authors induced a topic model using Latent Dirichlet Analysis (LDA) for each word, and then extended the model to documents and subdomains according to observed distributions. Their conclusion is that an unsupervised

machine learning system is able to create robust clusters of subdomains, thus proving their hypothesis that the commonly used molecular biology subdomain is not representative of the domain as a whole. In contrast, we examine the differences and similarities between biomedical sublanguages at the level of named entities, using supervised machine learning algorithms and on a different number of subdomains.

It follows that tools which were developed and evaluated on corpora derived from one subdomain might not always perform as well on corpora from another subdomain. Understanding these linguistic variations is essential to domain adaptation of natural language processing tools, e.g., cross-domain instance weighting, ensemble learning and semi-supervised learning (Jiang, 2008).

We initially created a corpus of documents from various biomedical subdomains, from which we then extracted named entity information automatically. The NEs were later transformed into input for machine learning algorithms, as discussed below.

4.2.1 Document Collection

A corpus was created by first searching the National Library of Medicine (NLM) Catalog[2] for journals which are in English and available via PubMed Central (PMC), and then narrowing down the results to those whose Broad Subject Term attributes contain only one biomedical subdomain name. Since we are interested in full-text articles, we retained only those journals which are available within the PubMed Open Access subset[3]. After obtaining the total number of documents across different journals in each subdomain, we retained only those subdomains with at least 400 documents.

Using the PMC identifiers of all articles under the 20 remaining subdomains, we

[2]http://www.ncbi.nlm.nih.gov/nlmcatalog
[3]http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist

Subdomain	Shortname	No. of words
Allergy and Immunology	Allergy	0.9M
Biology	Biology	3.3M
Cell Biology	CellBio	3.2M
Communicable Diseases	Communi	1.4M
Critical Care	Critica	1.6M
Environmental Health	Environ	1.9M
Genetics	Genetic	3.0M
Health Services Research	HealthS	1.7M
Medical Informatics	Medical	2.6M
Medicine	Medicin	2.1M
Microbiology	Microbi	2.6M
Neoplasms	Neoplas	2.2M
Neurology	Neurolo	2.3M
Pharmacology	Pharmac	1.8M
Physiology	Physiol	3.5M
Public Health	PublicH	1.7M
Pulmonary Medicine	Pulmona	1.9M
Rheumatology	Rheumat	1.9M
Tropical Medicine	Tropica	1.7M
Virology	Virolog	2.3M

Table 4.1: The 20 subdomains in the corpus, their shortnames and number of words in the corpus subset.

retrieved documents from Europe PMC[4]. For each subdomain, we randomly selected 400 documents, which contain automatically annotated named entities. Since the retrieved documents are in XML format, several unusable fragments were removed before converting them to plain text. Examples of such fragments are article metadata (authors, affiliations, publishing history), tables, figures, and references. Table 4.1 shows the 20 subdomains and the approximate size of the corresponding corpus subset (in number of words) after the pre-processing step.

[4]http://europepmc.org/

4.2.2 Tagging of Named Entities

We formed a silver standard corpus by harmonising the annotations of multiple resources and named entity recognisers. This method was chosen due to the fact that there are no gold standard annotations available for such a large number of full-text articles.

To create the named-entity-tagged corpus, we used a simple method that augments the named entities present in the Europe PMC articles with the output of two NER tools, i.e. NeMine and OSCAR. In Europe PMC, only six named entity types are annotated; with the use of NeMine and OSCAR, however, we obtained a total of 19 different classes of entities, summarised in Table 4.2.

Named entities in the Europe PMC database were identified using NeMine (Sasaki et al., 2008), a dictionary-based statistical named entity recognition system. This system was later extended and used by Nobata et al. (2009) to include more types, such as phenomena, processes, organs and symptoms. We used this most recent version of the software as our second source of more diverse entity types.

The Open-Source Chemistry Analysis Routines (OSCAR) software (Corbett and Copestake, 2008; Jessop et al., 2011; Kolluru et al., 2011) is a toolkit for the recognition of named entities and data in chemistry publications. Currently in its fourth version, it uses three types of chemical entity recognisers, namely regular expressions, patterns and Maximum Entropy Markov models.

Nevertheless, due to the combination of several NER systems, some NE types are more general and comprise other more specific types, therefore leading to double annotation. For instance, the *Gene—Protein* type is more general than both *Gene* and *Protein*, so only *Gene* or *Protein* will be kept in case they overlap with *Gene—Protein*. The same applies to the *Chemical molecule* type, which is a hypernym of *Gene*, *Protein*, *Drug* and *Metabolite*. In the case of multiple annotations over the same span of

Type	Europe PMC	NeMine	OSCAR
Gene	✓	✓	
Protein	✓	✓	
Gene—Protein	✓		
Disease	✓	✓	
Drug	✓	✓	
Metabolite	✓	✓	
Bacteria		✓	
Diagnostic process		✓	
General phenomenon		✓	
Indicator		✓	
Natural phenomenon		✓	
Organ		✓	
Pathologic function		✓	
Symptom		✓	
Therapeutic process		✓	
Chemical molecule			✓
Chemical adjective			✓
Enzyme			✓
Reaction			✓

Table 4.2: Named entity types and their source.

text, we removed the more general *Chemical molecule* type, so that each entity is labelled only with the more specific category assigned. Although this type of multiple annotations was frequent, we did not encounter any case of contradicting annotations over the same span of text.

This corpus is available upon request from the author.

4.2.3 Experimental Setup

Based on the corpus previously described, we created a data set for supervised machine learning algorithms. Every document in the corpus was transformed into a vector consisting of 19 features. Each of these features corresponds to an entity type in Table 4.2, having a numeric value ranging from 0 to 1. This value, θ, represents the ratio of the specific entity type to the total number of named entities recognised in that document,

as shown in Equation 4.1.

$$\theta = \frac{n_{type}}{N} \tag{4.1}$$

where n_{type} represents the number of named entities of a certain type in a document and N represents the total number of named entities in that document. Each vector was labelled with the name of the subdomain to which the respective document belongs.

From the twenty subdomains in the corpus, we formed all possible combinations of two (thus resulting in a total of 190 pairs), for each of which we built a binary classifier. Weka (Witten and Frank, 2005; Hall et al., 2009) was employed as the machine learning framework, due to its large variety of classification algorithms. We experimented with a large number of classifiers, including J48, JRip, Logistic, RandomTree, RandomForest, SMO and combinations of these with AdaBoost. Evaluation was performed using the 10-fold cross-validation technique. RandomForest obtained the best F-score in 86 out of the 190 subdomain pairs, whilst the best result in 98 cases was obtained by AdaBoost in combination with other algorithms (JRip, RandomTree, Logistic). The remaining pairs were best classified by JRip (4 pairs) and Logistic (2 pairs). We therefore decided to present only the results using RandomForest.

4.2.4 Feature Evaluation

To confirm the value of the selected features in classifying documents into subdomains, we performed the chi-squared (χ^2) test of independence between each named entity and each pair of subdomains. Chi-squared is defined in Equation 4.2, whilst the expected value of the observation is computed according to Equation 4.3.

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{4.2}$$

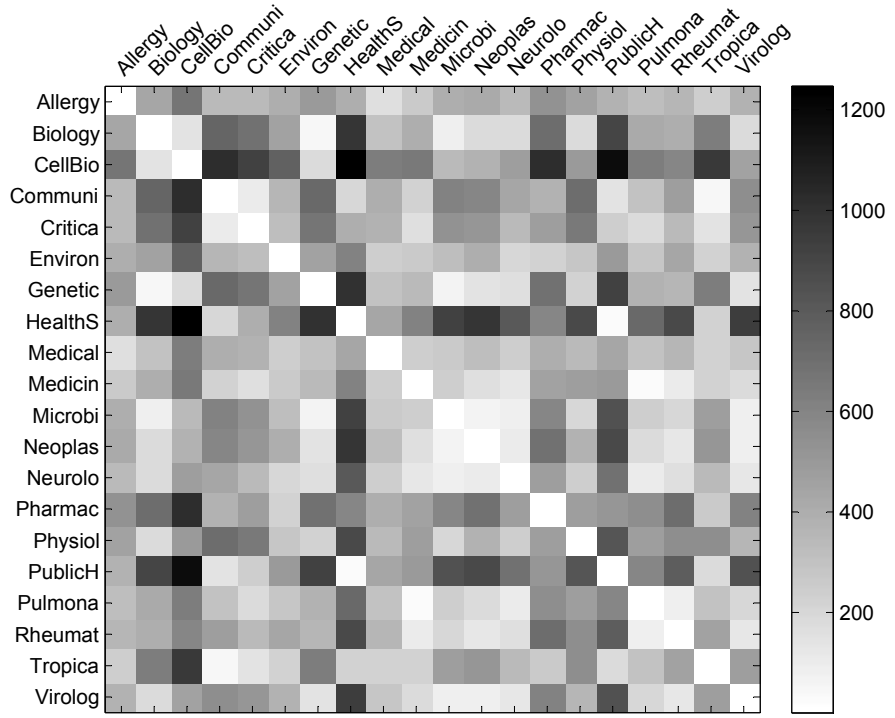


Figure 4.1: A heatmap showing the Frobenius norm based on the chi-squared vector for each pair of subdomains.

$$E_{i,j} = \frac{\sum_{k=1}^{c} O_{i,k} \sum_{k=1}^{r} O_{k,j}}{N} \tag{4.3}$$

where r and c are the number of rows and columns, respectively, in the contingency table.

The values are obtained by applying the ChiSquare Attribute Evaluator that is implemented in Weka. Each result contains a vector of 19 chi-squared scores, one for each feature. To visualise this graphically, we computed the Frobenius norm of the vector of chi-squared values for each subdomain pair. The Frobenius norm is defined as the square root of the sum of the absolute squares of its elements, as seen in Equation 4.4 (Golub and van Van Loan, 1996).

$$\|A\|_F = \sqrt{AA^*} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{4.4}$$

where A^* denotes the conjugate transpose of A, and m and n are the size of the matrix A.

The resulting heatmap is included as Figure 4.1. The higher the value of the Frobenius norm, the better is the combination of features for distinguishing between the two subdomains in the pair.

To gain an insight into which features contribute most or least to the overall task, the sum of the chi-squared statistic for each feature was taken over all pairs of subdomains. We present the mean values obtained from this exercise in Table 4.3.

Type	**Mean**
Disease	195.06
Gene—Protein	145.94
Protein	140.83
Metabolite	112.17
Reaction	108.43
Chemical molecule	87.84
Drug	82.57
Gene	78.03
Indicator	63.10
Therapeutic	56.09
Organ	35.78
Enzyme	30.77
Diagnostic process	24.30
Chemical adjective	19.07
Symptom	16.46
Bacteria	10.57
Natural phenomenon	7.07
Pathologic function	5.79
General phenomenon	0.34

Table 4.3: Mean values of the χ^2 statistic for each feature over all pairs of subdomains.

4.2.5 Classifier Results

From the 20 subdomains, a binary classifier was built for each possible subdomain pair, as discussed in the previous section. The heatmap in Figure 4.2 shows the performance

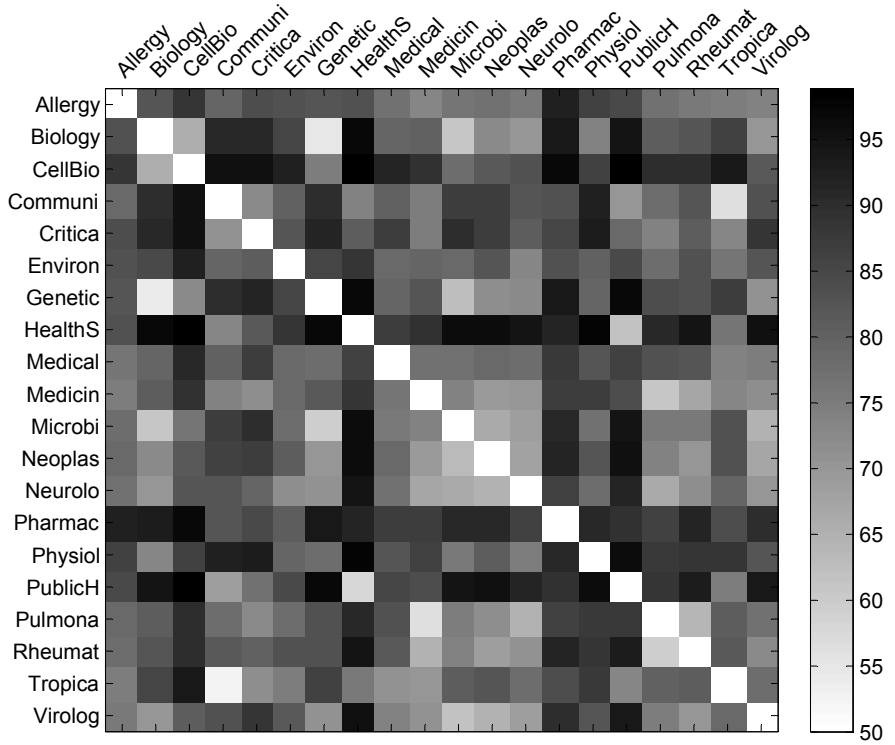


Figure 4.2: A heatmap showing the performance (in F-score) of each classifier built for each pair of subdomains.

of each of the 190 pairs in terms of F-score. This heatmap is non-symmetric, in the sense that the F-score of subdomains A and B is different from that of B and A. All F-scores presented in this heatmap are computed with respect to the subdomain on the Y-axis (left) and against the subdomains on the X-axis (top).

A cell with a dark shade of grey corresponds to a pair of subdomains which are discernible from each other by a classifier trained on named entity type frequencies. *Cell Biology* and *Pharmacology*, for example, are found to have very distinct named entity type frequencies, as evidenced by the very good performance (97.15% F-score) of the classifier for them.

In contrast, a lighter tint of grey means that the corresponding pair consists of subdomains which are very similar in their named entity type frequencies. Such is true in the case of *Communicable Diseases* and *Tropical Diseases*, for instance, in which the classifier obtained an F-score of 56.63%.

Subdomain	Similar subdomains
Biology	Cell Biology, Genetics, Microbiology
Communicable Diseases	Tropical Diseases
Medicine	Pulmonary Medicine
Health Services Research	Public Health
Genetics	Microbiology
Pulmonary Medicine	Rheumatology
Microbiology	Virology

Table 4.4: Similar subdomains; the subdomains listed in the second column can be considered as highly similar to the corresponding subdomain in the first column based on their named entity type frequencies.

4.2.6 Analysis

From these results, we are able to enumerate the subdomains which can be considered as different or similar to a subdomain of interest in terms of frequencies of their named entity types. In obtaining the most similar subdomains, we looked at the pairs whose F-score is at the lower end of the scale. There are no pairs for which the F-scores are between 50 to 55%, and only two pairs fall within the 55-60%-range. We hence used as threshold an F-score of 65% (i.e., subdomains in pairs for which the F-score of the classifier is 65% and below were considered similar). In contrast, we looked at the other end of the scale (i.e., pairs for which the F-score of the classifier is 95% and above) to obtain a listing of the most dissimilar subdomains.

Findings in Table 4.4 suggest that when building NLP tools (e.g., named entity recognisers) for documents under the subdomain in the first column, one might trivially adapt those developed for the corresponding subdomains in the second column. A named entity recogniser for the *Microbiology* subdomain, for example, might be trivially applied to *Virology* documents. However, it might also be the case that there are no named entity recognisers built yet that are specialised for these subdomains.

In contrast, those built for the subdomains in the second column of Table 4.5 might need further training or adaptation in applying them to the corresponding subdomain

Subdomain	Dissimilar subdomains
Biology	Public Health, Health Services Research
Cell Biology	Critical Care, Communicable Diseases, Pharmacology, Public Health, Health Services Research
Genetics	Public Health, Health Services Research
Health Services Research	Microbiology, Neoplasms, Physiology, Rheumatology, Virology
Neoplasms	Public Health
Physiology	Public Health

Table 4.5: Dissimilar subdomains; the subdomains listed in the second column can be considered as different from the corresponding subdomain in the first column based on their named entity type frequencies.

in the first column, as these tools might have been trained on documents where the named entity types which occur frequently in the subdomain of interest are sparse. For instance, there is no certainty that NER tools developed for the *Pharmacology* domain will work well on *Neoplasms* documents.

We computed the mean along each row and column of the heatmap, and determined that both the row and column corresponding to *Medicine* produced the minimum, while *Pharmacology* has the maximum. This finding suggests that *Medicine* is the biomedical subdomain which is most "alike" every other subdomain, irrespective of the direction F-score is computed in, whilst *Pharmacology* is the least one. In developing a named entity recogniser for *Pharmacology*, one has to consider its differences with other biomedical subdomains in terms of named entity type distributions.

4.2.7 Summary

We formed a silver standard corpus from 20 biomedical subdomains and built a binary classifier for each possible subdomain pair. From the results, we have observed that most subdomains are highly discernible from each other by a classifier, in terms of

named entity type frequencies. This proves the fact that semantics too plays an important role in characterising a subdomain and its corresponding sublanguage, exhibiting slight variations to which machine learners are sensitive. However, there are also several cases when a classifier is unable to distinguish between subdomains, implying that they have highly similar named entity type distributions. This usually happens when the two subdomains are in a IS-A relation, such as *Cell Biology* and *Biology*. Since discourse relations, and, implicitly, causality, depend on semantics, such differences and similarities in named entity type frequencies should be considered when developing automated tools for one subdomain and adapting them for use on another.

4.3 Causality representation

Conceptually, the annotation involves two basic annotation primitives, spans and relations. Spans represent continuous portions of text with an assigned type, whilst relations are directed, typed, binary associations between two spans. Spans mark both the specific statements in text that play the roles of *Cause* and *Effect* in statements of causality, as well as expressions that explicitly state the existence of a causal relation.

The annotation involves two span types: ARGUMENT and TRIGGER. The former is used to mark statements that are part of a causal relationship, whilst the latter is used to mark phrases that express causal triggers. For instance, in example (4.1), the text spans "*A occurred*" and "*B happened*" would be marked as ARGUMENT, whilst the text span "*Thus*" as TRIGGER.

(4.1) A occurred.

Thus, B happened.

In contrast, relations identify connections between the various spans of text. The relation types identify the roles that the spans of text play in the association. The annotation involves two relation types: CAUSE and EFFECT. EFFECT always marks the statement that is stated as the result, whilst CAUSE marks the statement that leads to that result. The difference between these two concepts is detailed below, in Section 4.4. In example (4.1), the relation from the trigger *Thus* to the argument *A occurred* would be a CAUSE relation, whilst the relation from the trigger to the argument *B happened* would be an EFFECT relation.

4.4 Causality annotation

The sense type "Cause" is used when the two arguments of the relation are related causally and are not in a conditional relation. As previously mentioned, this definition is rather vague, so annotators must also use other methods in order to recognise causality. Thus, considering previous research (Bethard et al., 2008; Grivaz, 2010), they were asked to check for temporal asymmetry and counterfactuality, try rewording and other linguistic tests, such as the insertion of explicit causal triggers and checking whether the rephrasing is equivalent to the original.

Cause - Effect pairs are annotated as centred on a TRIGGER span, whilst the associated spans are of type ARGUMENT, with CAUSE and EFFECT representing the direction (Figure 4.3). The span identifying the causal trigger (TRIGGER) may be empty, but a non-empty span is marked in all cases where an explicit connective occurs. In cases where there is no explicit connective expressed, the TRIGGER span is placed in between the two ARGUMENT spans with an empty (zero-width) span, as shown in Figure 4.4.

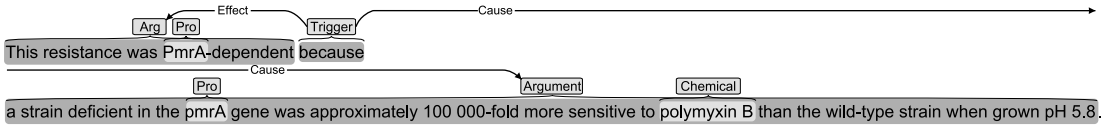


Figure 4.3: Example of Cause–Effect annotation with an explicit trigger.

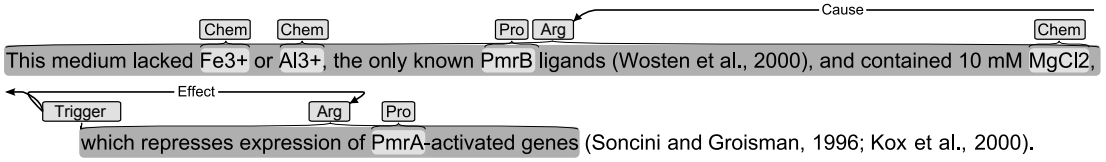


Figure 4.4: Example of Cause–Effect annotation with an implicit trigger.

4.5 Annotation software and format

The original event annotation of the BioNLP ID Shared Task corpus was performed using BRAT (Stenetorp et al., 2012). This is a web-based annotation tool aimed at enhancing annotator productivity by simplifying and automating parts of the annotation process. Customising the settings of BRAT is reasonably straightforward, allowing users to change the information to be annotated and the way it is displayed. Furthermore, BRAT is freely available under the open-source MIT licence from its homepage[5]. As such, we decided to continue to use this tool for our task of annotating causality relations in text.

The stand-off annotation files are kept separate from the original text files and are connected to them by character offsets. Each span annotation (TRIGGER and ARGUMENTS) has a unique identifier and encodes the start and end offsets of the text span, the type of the span and the actual text span annotated, all separated by tabs. Each causal relation has a unique identifier and stores the identifiers of the trigger and the two arguments, together with their relation subtype. An example of a complete relation annotation is illustrated in Figure 4.5.

[5]http://brat.nlplab.org

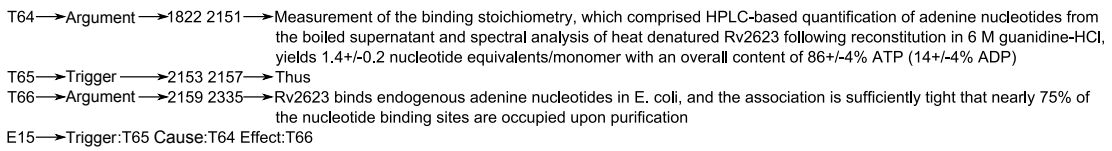


Figure 4.5: Example of an annotation file as created by BRAT.

This simple, yet highly efficient format allows for easy processing and full transformation into other formats (e.g., XML), thus increasing the portability between various annotation systems. Furthermore, since this schema is not very specific, it can be reused and easily applied to other datasets, not necessarily belonging to the biomedical domain. Moreover, being represented in an offset stand-off format, the schema can allow the existence of other annotations over the same source text without creating annotation conflicts, such as overlapping in XML. In this case, the text is already annotated with named entity and event information. Other types of annotation are allowed and can be successfully integrated (e.g., part-of-speech and dependency).

4.6 Annotators and training

Although it has been shown that linguists are able to identify certain aspects in biomedical texts reliably, such as negation and speculation (Vincze et al., 2008), they could be overwhelmed in trying to understand the semantics. Identifying which events affect which events, especially when a causal trigger is not explicitly stated, is an extremely difficult task, as it requires vast, domain-specific background knowledge and an almost complete understanding of the topic. Therefore, due to the specificity of the biomedical domain, it is necessary for the annotators to be experts in this field of research. Furthermore, the annotators must have near-native competency in English. For the purpose of this task, two human experts have been employed to create the annotations in the corpus.

Besides the biomedical expertise, the two selected annotators also have extensive

experience in annotating text from the biomedical domain for text mining purposes. They have previously participated in other annotation efforts focussing on creating gold standard corpora of named entities, events and meta-knowledge. The annotators undertook a period of training prior to commencing the annotation task proper. During this time, they were given a small set of documents to practise on. As a result, they became accustomed to both the annotation tool and the guidelines.

Both annotators were given the same subset of articles to annotate, independently of each other. This allowed the detection of annotation errors and disagreements between annotators. They produced annotations in small sets of documents, which were then analysed and in response to which the annotators obtained feedback detailing their errors. Also, the annotators offered feedback regarding the annotation tool and guidelines, in order to increase the speed of the process. This led to noticing potential problems with the guidelines, which were addressed accordingly. The final guidelines were produced after the training period finished and these were used for the actual annotation.

4.7 General analysis of BioCause

The corpus contains a total of 850 causal relation annotations spread over 19 open-access biomedical journal articles regarding infectious diseases.

Table 4.6 summarises the general statistics of the corpus. Counting the unique explicit trigger types was performed using two settings. On the one hand, we considered the surface expression of the trigger, thus distinguishing between all morphological variants and modifications by adverbs, prepositions or conjunctions. For instance, the triggers *thus* and *and thus* were treated as separate types, as well as *suggest* and *suggests*. However, the case of the triggers was ignored. As can be seen from the table, there are 381 unique explicit triggers in the corpus. This means that, on average, each

Feature	Value
No. of articles	19
No. of causal associations	850
No. of implicit associations	50
No. of unique explicit triggers	381
No. of unique lemmatised explicit triggers	347
Tokens per trigger	3.09
Tokens per CAUSE arg.	21.31
Tokens per EFFECT arg.	16.87

Table 4.6: General statistics for the BioCause corpus.

trigger is used only 2.10 times.

On the other hand, all tokens forming triggers were lemmatised prior to counting. This means that both *suggest* and *suggests* are counted for the same trigger type. There are 347 unique lemmatised triggers in the corpus, corresponding to an average usage of 2.31 times per trigger. Both count settings show the diversity of causality-triggering phrases that are used in the biomedical domain.

Furthermore, the causal argument of the relation is, on average, almost 1.27 times longer than the other argument, the effect. This is due to the specificity of the biomedical domain and also the nature of research articles, where usually a causal argument that leads to an effect is complex and is composed of several, concatenated causes. This is exemplified below, in Section 4.9.

We also looked at the distribution of causality relations in the distinct discourse zones that are common in research articles. Figure 4.6 depicts the percentage of causal relations over six discourse zones, as given in Equation 4.5.

$$f_a(i) = \frac{n_i}{\sum_{j \in D} n_j} \tag{4.5}$$

where n_i is the number of causal relations in zone i, and D is the set of discourse zones. The discourse zones are *Title and abstract*, *Introduction*, *Background*, *Results*,

Discussion, *Results and discussion* and *Conclusion*. The zone *Results and discussion* is included because this is how some of the articles have been segmented in their original form.

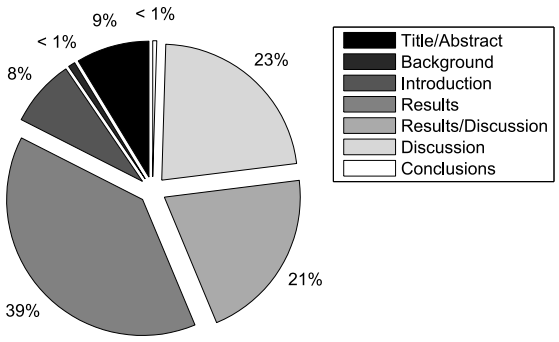


Figure 4.6: Actual distribution of causal associations in the corpus amongst seven different discourse zones.

As expected, most causal relations (over 80%) occur in the *Results*, *Discussion and Results* and *Discussion* section of articles, whereas the *Background* and *Conclusion* section contain a very small number of relations, just over 1%. However, because the discourse zones are very different in size, we also computed the frequency of causal relations relative to the number of tokens present in that respective discourse zone, as given in Equation 4.6.

$$f_r(i) = \frac{n_i}{||d_i||} \tag{4.6}$$

where n_i is the number of causal relations in zone i, and $||d_i||$ is the size of discourse zone i in words.

This distribution is depicted in Figure 4.7. The results change quite dramatically and tend to be more balanced when computed in this manner. The *Title and abstract* section becomes the zone with the highest causal relation density (over 23%), whilst in *Background* and *Conclusion* there are 17%. The *Results*, *Discussion* and *Results and discussion* sections contain 50% of the total number of causal relations.

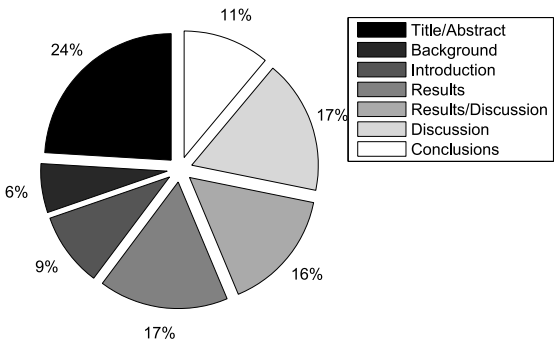


Figure 4.7: Distribution of causal associations in the corpus amongst seven different discourse zones relative to the number of tokens in each zone.

4.8 Analysis of causal triggers

Table 4.7 lists the 22 most frequent causality triggers in the corpus, together with their count in the corpus as a whole. These are counted in a surface expression setting. In total, the causality relations that are centred on these 22 triggers (only 5.77% of all trigger types) constitute more than 30% of the cases of causality in the entire corpus.

Similarly, Table 4.8 contains the 22 most frequent triggers that occur at least five times, counted in a lemmatised setting. The lemmas are automatically generated by the Enju parser. These 22 triggers occur 332 times, accounting for almost 41.5% of the total number of causality cases. The data in both these tables suggest that the majority of relevant causality relations are centred on a relatively small set of phrases and words. Indeed, in the entire corpus, only 22 distinct phrases or words have been used to annotate five or more causal relations, whilst the remaining explicit triggers have a very low frequency of less than five occurrences. As with many other natural language phenomena, this distribution is Zipfian. Almost all of the entries in Tables 4.7 and 4.8 correspond to phrases or words which usually denote a causal relation or inference between two spans of text.

Figure 4.8 shows the usage of annotated triggers as having causal and non-causal meaning in black and grey, respectively. Each trigger type has been allocated an ID

Feature	Count (relative frequency)
suggesting that	51 (6.04%)
thus	42 (4.98%)
indicating that	34 (4.03%)
therefore	17 (2.01%)
these results suggest that	14 (1.66%)
suggests that	12 (1.42%)
due to	10 (1.18%)
suggesting	10 (1.18%)
indicating	9 (1.06%)
the results indicate that	9 (1.06%)
these results indicate that	9 (1.06%)
suggest that	8 (0.94%)
because	7 (0.83%)
caused	6 (0.71%)
required for	6 (0.71%)
resulting in	6 (0.71%)
which suggests that	6 (0.71%)
and thus	5 (0.58%)
indicates that	5 (0.58%)
suggests	5 (0.58%)
these data indicate that	5 (0.58%)
these observations suggest that	5 (0.58%)

Table 4.7: Count and relative frequency for the most frequently occurring triggers using surface-expression forms.

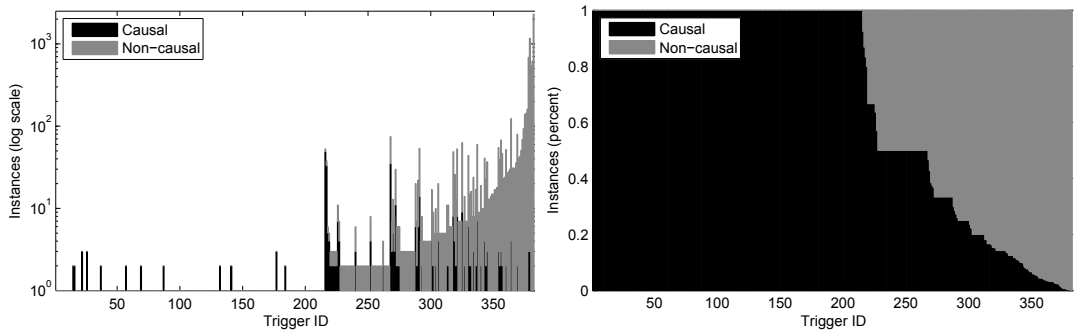
and two charts have been produced. The trigger IDs have remained unchanged for the purpose of producing the two charts. Figure 4.8a depicts the actual number of causal/non-causal instances for each trigger, whilst Figure 4.8b is based on the ratio of causal:non-causal instances for each trigger. A logarithmic scale is used in Figure 4.8a for visibility purposes, as there are many small values and very few large ones.

By analysing both figures simultaneously, it can be noticed that there exists a large number of triggers which seldom occur, but which are exclusively causal in meaning. More than 200 triggers, to the left of both charts, occur less than 20 times each, but they are 100% causal. This high variability in expressing causality represents one significant problem in automatically detecting causal triggers.

Feature	Count (relative frequency)
suggest that	75 (9.36%)
indicate that	45 (5.62%)
thus	42 (5.24%)
suggest	20 (2.50%)
therefore	17 (2.12%)
these result suggest that	15 (1.87%)
indicate	12 (1.50%)
cause	10 (1.25%)
due to	10 (1.25%)
result in	9 (1.12%)
the result indicate that	9 (1.12%)
these result indicate that	9 (1.12%)
because	7 (0.87%)
demonstrate that	7 (0.87%)
which suggest that	7 (0.87%)
lead to	6 (0.75%)
require for	6 (0.75%)
these observation suggest that	6 (0.75%)
and thus	5 (0.62%)
confirm that	5 (0.62%)
our finding indicate that	5 (0.62%)
reveal that	5 (0.62%)

Table 4.8: Count and relative frequency for the most frequently occurring triggers using lemmatised forms.

On the right side of the graphs, the total number of occurrences of causal triggers in the corpus increases very quickly, but the percentage of causal meaning decreases drastically. Example (4.2) shows one of the 78 instances of the word *when* acting as a non-causal trigger. In the case of *when*, there are only two instances which denote causality. There are over 50 triggers (rightmost) that occur less than 20% causally from their at least 20 occurrences in the corpus. In fact, there are 64 trigger types which occur only once as a causal instance, whilst the average number of non-causal instances for these types is 14.25. This shows the high ambiguity of causal triggers, the other important issue of automatically identifying them.



(a) Number of causal/non-causal instances for each annotated trigger

(b) Percentage of causal/non-causal usage for each annotated trigger

Figure 4.8: Usage of annotated triggers as having causal and non-causal meaning.

(4.2) Colonization analysis also revealed the incapability of the DeltasalKR mutant to colonize any susceptible tissue of piglets *when* administered alone.

Furthermore, the explicit triggers can be classified into two categories, according to their means of lexicalisation. Firstly, there are triggers which are expressed using subordinating conjunctions or adverbials. These are shown in examples (4.3) and (4.4), respectively. There are 37 distinct triggers which belong to this class.

(4.3) This acid pH-promoted increase appears to be specific to a subset of PhoP-activated genes that includes pmrD *because* expression of the PhoP-regulated slyA gene and the PhoP-independent corA gene was not affected by the pH of the medium.

(4.4) Mlc is a global regulator of carbohydrate metabolism and controls several genes involved in sugar utilisation.

Therefore Mlc also affects the virulence of Salmonella.

The second type is composed of triggers belonging to open-class part-of-speech categories, mainly verbs or nominalised verbs, which are usually modified by conjunctions, prepositions or subordinators. Most of these are of the form subject-predicate, lexicalised as pronoun/noun + verb + adverbial/conjunction/subordinator, where the pronoun/noun is an anaphorical referent to the argument that first appears in the text and the verb shows the relation to the following argument. An instance of this case is shown in example (4.5), where the verb *suggested* denotes the causal relationship and the subject *This* refers anaphorically to the first sentence. Other patterns also exist, although with a lower frequency, such as prepositional phrases and verb phrases.

(4.5) There was residual pbgP expression in the pmrB mutant induced with mild acid pH, which was in contrast to the absence of pbgP transcription in the pmrA mutant.
This suggested that PmrA could become phosphorylated from another phosphodonor(s) when PmrB is not present.

In fact, there are 165 distinct syntactic patterns that cover the entire set of causal triggers in BioCause. Only nine patterns have a count of over ten instances, but they make up for half of all triggers. These are listed in Table 4.9. As can be noticed, most of the triggers contain one verb, which usually comes with a noun subject and complementiser. Adverbials and conjunctions also make it in the top nine.

The rest of 156 patterns are more complex versions of the ones listed in this table. Variations include multiple nouns, auxiliary verbs, determiners, adjectives or prepositions.

Pattern	Count (relative frequency)
V-C	133 (20.27%)
V	63 (9.60%)
ADV	59 (8.99%)
D-N-V-C	42 (6.40%)
N-V-C	28 (4.27%)
V-P	22 (3.35%)
SC	14 (2.13%)
N-V	11 (1.68%)
D-N-V	10 (1.52%)

Table 4.9: Count and relative frequency for the most frequently occurring PoS patterns for triggers.

Parent	Count (relative frequency)
S	331 (41.48%)
VP	228 (28.57%)
ADV	72 (9.02%)
V	68 (8.52%)
PP	21 (2.63%)
SCP	16 (2.00%)
SC	14 (1.75%)

Table 4.10: Count and relative frequency for the most frequently occurring parent constituents for triggers.

Triggers are covered by constituents that can belong to various syntactic categories. Table 4.10 lists the seven parents that have more than ten occurrences in the corpus. In total, there are 17 different types of parents. As can be observed, most triggers (41%) have a Sentence constituent as their parent, whilst another 28% belong to a verb phrase. These seven types represent almost 94% of all trigger parents.

We also report, in Figure 4.9, the distribution of the length of triggers annotated in the corpus, in terms of tokens. As can be seen in the figure, more than 50% of the total number of triggers consist of one or two words, whilst around 25% consist of three or four words. The length of the trigger appears to be inversely proportional to its frequency – the longer the trigger, the more uncommon it is. Again, the distribution

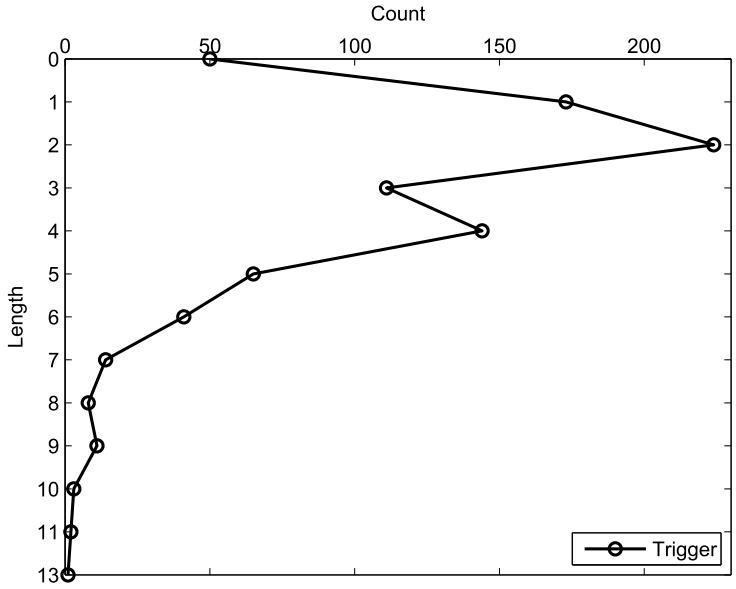


Figure 4.9: Distribution of triggers according to their length in tokens.

has a Zipfian shape.

4.9 Analysis of causal arguments

The arguments of a causal relation can be classified into two categories, depending on the type of relation to the trigger. Syntactically, one argument is DA on the causal trigger, whilst the other is IA of the causal trigger. Furthermore, the IA can be found in the SS as the trigger, or in a DS. Semantically, one argument plays the role of Cause, whilst the other plays the role of Effect. In this section, we analyse causal arguments from both these perspectives in this section, and also observe the connections between the two types.

Figure 4.10 shows the distribution of the lengths of both the Cause (black) and the Effect arguments (grey) in the corpus, in terms of tokens.

As previously mentioned, it can be noticed that the Cause argument is usually longer than the Effect argument. This is due to the style used in biomedical research articles, in which multiple causal elements are concatenated or explained in order to

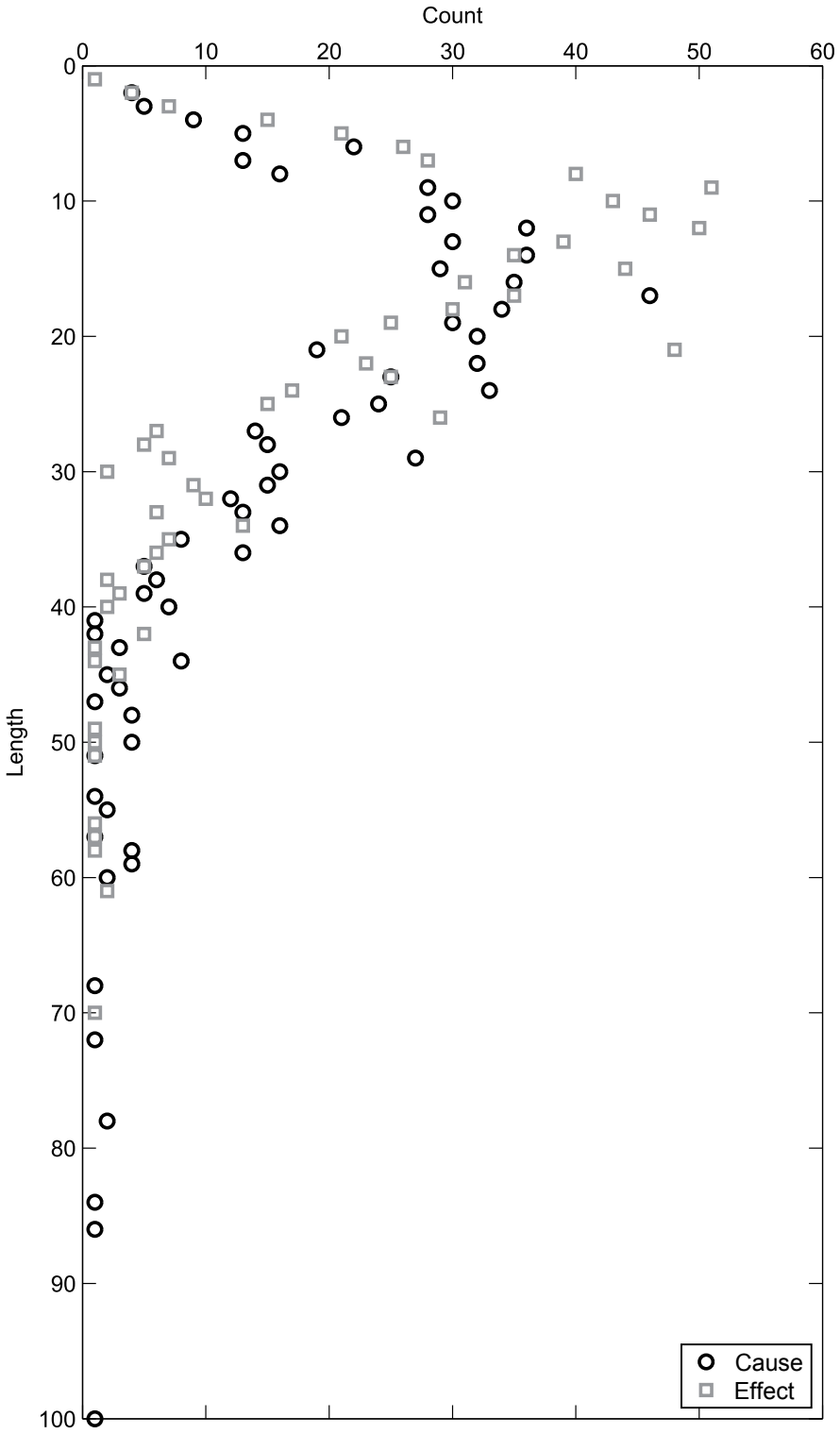


Figure 4.10: Distribution of Cause and Effect arguments according to their length in tokens. Data points are plotted only where there are instances of arguments of that length.

Order	Count	Relative frequency
C-_-E	30	3.52%
E-_-C	20	2.35%
C-T-E	687	80.82%
E-T-C	91	10.69%
E-C-T	2	0.23%
T-C-E	9	1.06%
T-E-C	11	1.29%

Table 4.11: Distribution of the order of arguments (Cause – C, Effect – E) relative to the trigger (T). Implicit triggers are marked using an underscore character ('_').

infer an effect. Take, for instance, the sentences in example (4.6), where two causal elements (namely "the activation of the hilA transcription" and "that of HilC/D-dependent invFD expression") are connected by a coordinating conjunction ("and"). Another frequent case is the inclusion of explanations or supplementary information, without which the inference could not be possible. This explains why this information is also included in the argument annotation spans.

(4.6) Since HilD activates the transcription of hilA (14), which in turn can activate HilA-dependent invFA expression (10), and directly activates HilC/D-dependent invFD expression, *these results establish that* the mlc mutation exerts a negative effect on SPI1 gene expression, mainly by increasing the level of hilE expression.

The order of the arguments does not vary significantly, with more than 80% occurring in the form of Cause-Trigger-Effect. Table 4.11 shows the complete distribution of the order of the two arguments relative to the trigger. As can be seen, there are only 24 cases where the trigger appears before or after both arguments. In the case of implicit triggers, we considered them as being placed in between the two arguments.

It is also noticeable that the Effect argument is usually the dependent argument.

		Cause	
		SS	**DS**
Effect	**SS**	447 (55.88%)	320 (40%)
	DS	33 (4.12%)	0 (0%)

Table 4.12: Distribution of the position of Cause and Effect arguments in the same sentence (SS) or different sentences (DS) relative to the trigger.

This is specific to the scientific domain, where first a cause (or list of causes) is given, which is then followed by its effect(s).

The relative position of the two arguments is roughly balanced, as can be observed from Table 4.12. Just over half of causal relations are intra-sentential (55.88%), whilst 44.12% of these are inter-sentential. This proves the difficulty of the task: since there is no syntactic dependency between the trigger and its extra-sentential argument, the identification can be performed based only on lexical and semantic features.

Furthermore, the Effect argument is located in the same sentence with the causal trigger in more than 95% of instances. In contrast, the Cause argument is slightly more balanced: 60% of instances are located in the same sentence as the trigger, whilst 40% are in a different sentence.

When located in a different sentence than the causal trigger, the distance to the independent argument has the distribution given in Figure 4.11. As can be noticed, almost all DS IAs are located in previous sentences. Moreover, almost 61% of all DS IAs are located in the immediately previous sentence to that of the trigger. The frequency rapidly decreases to almost 20% and 11% in the case of the second and third previous sentences, respectively. A very low frequency exists up to the tenth previous sentence. With regard to the following sentences, only less than 2% of arguments are found there. Furthermore, they are usually found very close to the trigger sentence, mostly in the immediately following sentence.

There are no restrictions on how far the two arguments can be from each other in text. In other words, they may or may not be adjacent. Therefore, we have looked at

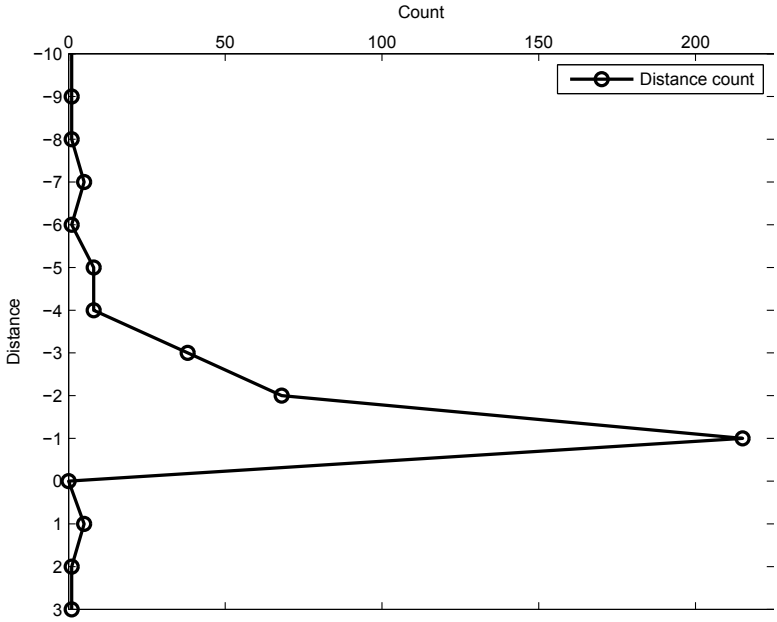


Figure 4.11: Distribution of the number of sentences from the trigger sentence to that of the DS IndArg inclusive.

the distance between the two arguments. We show in Figure 4.12 the frequency of the various distances measured by the number of tokens. The average distance between the two arguments is of 13.5 tokens. It should be noted that this distance also includes the trigger if this is placed in between the two arguments.

There are more than one hundred cases where the distance is two or three tokens (116 and 177, respectively). For the distance of four to six tokens, there are between 50 and 100 instances. It can be observed that the graph has a flat, yet long tail. There are almost 200 cases where the distance is greater than or equal to 10 tokens.

In terms of sentences, the distance is closely related to the distribution of distances from the trigger to the DS independent argument in Figure 4.11. As Figure 4.13 shows, around 60% of DS IAs are located in an immediately neighbouring sentence. About 30% are found in the second or third sentence, whilst an extremely low proportion of DS IAs are found in up to the tenth sentence.

Most arguments are found in the same sentence. Otherwise, they are found in immediately neighbouring sentences, or at most in the second or third neighbouring

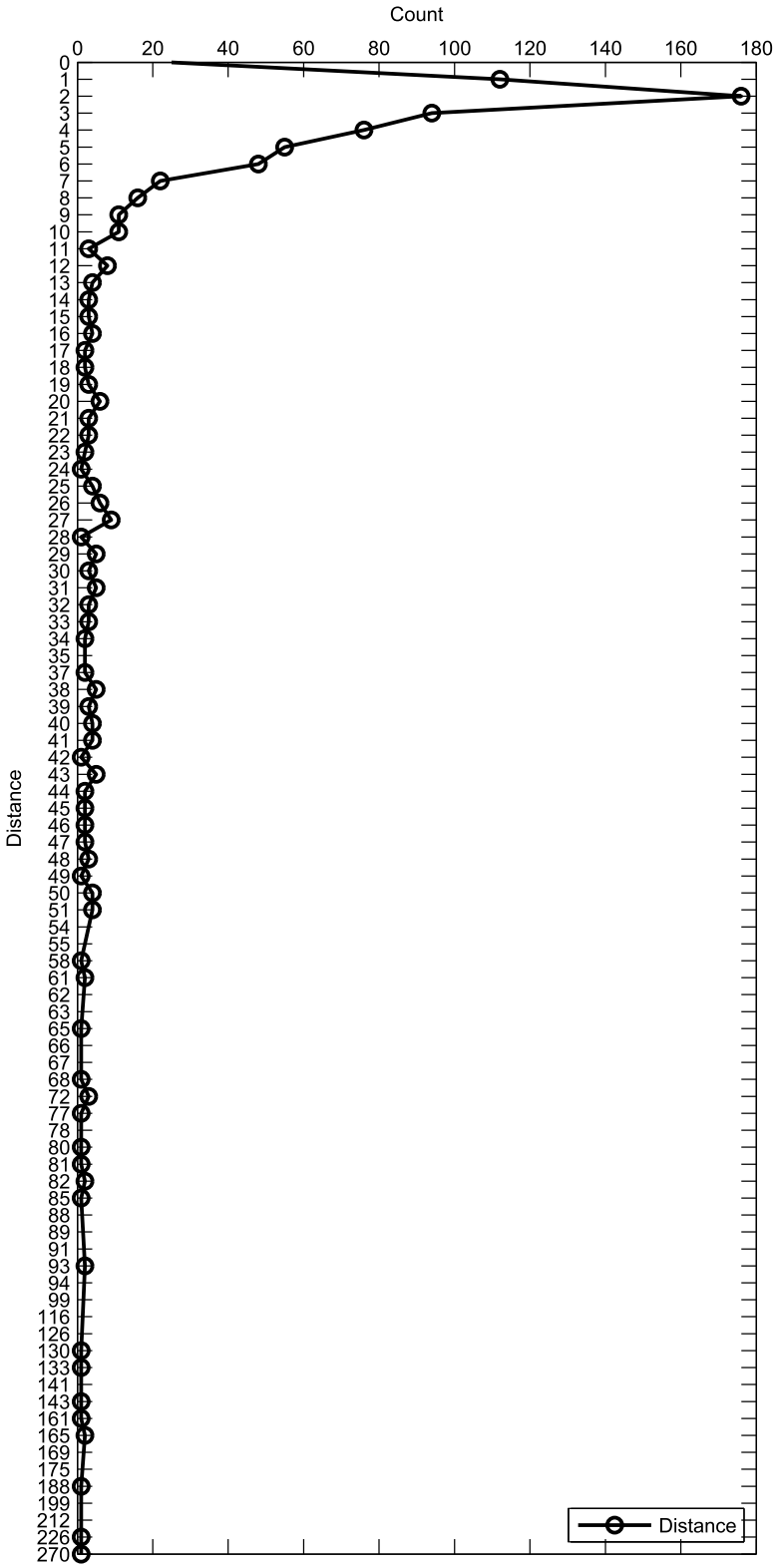


Figure 4.12: Distribution of the number of tokens between the arguments.

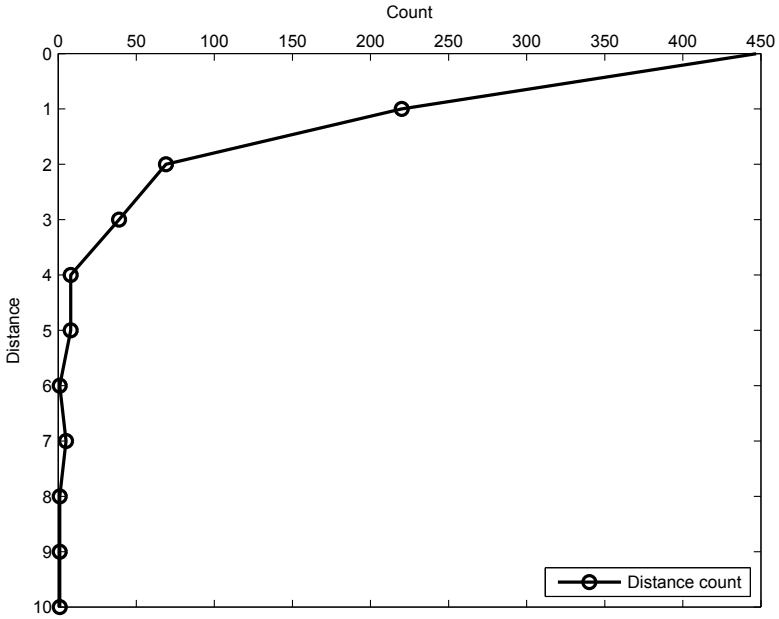


Figure 4.13: Distribution of the number of sentences between the arguments.

sentences. There exists a long tail of infrequent pairs of arguments located up to ten sentences away from each other.

4.10 Evaluating inter-annotator agreement

Due to the complexity of the annotation task and the variety of types of spans and relations, IAA cannot be computed using standard means. For instance, the Kappa statistic (Fleiss, 1981) cannot be used in our case, as this requires classifications to correspond to mutually exclusive and discrete categories. Instead, we have chosen to follow similar cases in selecting F-measure to calculate IAA (Thompson et al., 2009; Hripcsak and Rothschild, 2005).

F-measure is usually used to combine the precision and recall in order to compare the performance of an information retrieval or extraction system against a gold standard. In our case, precision and recall can be computed by considering one set of annotations as the gold standard. The resulting F-score will be the same, regardless of

which set is considered gold.

Because of the various angles of annotation, we have split the evaluation methodology into several subtasks of the annotation process. For each subtask, we calculated the inter-annotator agreement in terms of F-score. Initially, we computed the number of identical and overlapping triggers. For these triggers only, we then continued by counting the arguments, using both the exact match criterion and the relaxed match criterion introduced below. This is done separately for the CAUSE argument and for the EFFECT argument.

- Trigger identification – how many causal associations have the same trigger. Two separate values are computed here:
  - Exact match – trigger text spans match exactly.
  - Relaxed match – trigger text spans overlap with each other, but do not necessarily match exactly.
- Argument identification – for agreed triggers, how many have the same arguments. Four separate values are computed here, two for each argument:
  - Exact match – argument text spans match exactly.
  - Relaxed match – argument text spans overlap with each other, but do not necessarily match exactly.

In order to ensure the quality and consistency of the causality annotation throughout the corpus, three full articles (approximately 15% of the corpus) were annotated by both human experts. This allowed us to calculate the agreement levels between them. We first present some general agreement statistics on the corpus as a whole, followed by detailed numbers on each subtask. We also analyse the differences in annotation between the two experts.

Feature	First annotator	Second annotator
No. of causal associations	109	125
No. of implicit triggers	13 (11.93%)	18 (14.40%)
No. of explicit triggers	96 (88.07%)	107 (85.60%)
No. of tokens per trigger	2.80	2.87
No. of tokens per CAUSE arg.	19.55	17.60
No. of tokens per EFFECT arg.	13.94	13.84

Table 4.13: General inter-annotator agreement statistics for the corpus. The percentages represent the proportion of that specific type of causal relations in the total number of causal relations identified by that annotator.

4.10.1 General statistics

Table 4.13 contains a comparison between the two human expert annotators from various points of view. We included the number of causal associations, the number of implicit and explicit triggers, as well as the average length of the trigger and of the two arguments in tokens. These numbers are obtained from the annotations following the final guidelines.

As can be observed from the table, there is little difference between the two annotators in terms of the different comparison criteria. The second annotator has identified 16 more causal associations than the first annotator. Nevertheless, the percentage of explicit and implicit triggers remains rather stable over the two sets of annotations. This is also true with respect to the length in tokens of the triggers and the two arguments.

4.10.2 Subtask statistics

In order to compute the agreement level in F-score terms, we considered one annotator as the gold standard against which we compare the other annotator. We report in Table 4.14 the F-scores for the various subtasks. As can be observed, in all the doubly annotated documents, the two annotators agreed, with an exact match criterion, on 60 relations. This gives an F-score of 51.28%, which again proves the difficulty

Match type	Feature	F-score
	Exact relation	51.28%
	Relaxed relation	65.81%
	Exact trigger	64.10%
	Relaxed trigger	65.81%
ET	Exact CAUSE arg.	82.67%
	Relaxed CAUSE arg.	90.67%
	Exact EFFECT arg.	94.67%
	Relaxed EFFECT arg.	98.67%
RT	Exact CAUSE arg.	82.52%
	Relaxed CAUSE arg.	90.91%
	Exact EFFECT arg.	93.51%
	Relaxed EFFECT arg.	98.70%

Table 4.14: Inter-annotator agreement for relations, triggers and the two arguments in the case of exact-match triggers (ET) and relaxed-match triggers (RT).

and subjectivity of the task. In the case of relaxed matching, the F-score increases to 65.81%.

The two annotators agreed only on two thirds of the total number of triggers using an exact match criterion. The agreement increases by a small amount when relaxed matching is used. This demonstrates that identifying causal discourse relations is a relatively difficult task, even for experienced human judges.

The agreement on argument spans, nevertheless, is extremely high. This strongly suggests that once the annotators decide to mark a causal relation, finding the arguments is a rather straightforward task to accomplish. The F-score for identifying the CAUSE argument with an exact match rule is just over 80%, whilst the EFFECT argument is around 94%. This is due to the difficulty in recognising the exact cause in a causal relation. When the relaxed matching is used, the F-score increases significantly, to 90% for the CAUSE argument and 98% for the EFFECT argument.

These agreement values are in line with similar semantic annotation efforts for which F-score has been computed. For instance, in the BioNLP ST ID task, the partial-match inter-annotator agreement for event annotation is approximately 75%. However,

the arguments of these events have been already given as gold standard, therefore the task is significantly simpler than the one described in this article. Nevertheless, the best performing system participating in the shared task obtained an F-score of 56%.

After performing the double annotation and computing of the agreement scores, the disagreed cases were discussed between the annotators and the correct annotations were decided upon. Specifically, one of the two annotations was determined to be correct, an alteration was made or the annotation was removed completely. We also computed the agreement of each of the annotator with respect to the resulting gold standard corpus. In an exact-match setting, the F-score of each of the two annotators against the gold standard is 78.26% and 64.68%, respectively. Using a relaxed-match criterion, the F-scores increase to 86.17% and 87.73%, respectively.

4.10.3 Annotation discrepancies

We also looked at the differences between the two annotators. A number of these differences were simply annotation errors, where the selected spans contained extra characters from surrounding words or missed characters from the words on the boundaries. These have been corrected. The other differences relate to actual disagreements between the two annotators. Similarly to the subtasks on which we computed the agreement scores, the differences can be categorised in those relating to triggers or either of the two arguments.

Trigger discrepancies

In the doubly annotated section of the corpus, there are only two cases of overlapping, but not identical, triggers. One of them is given in example (4.7) below. One annotator considered the span “therein” to be the trigger, whilst the other annotator considered it

to be “therein appears to be”.

(4.7) Further bioinformatics analysis of the 89K island revealed a distinct two-component signal transduction system (TCSTS) encoded $_{Ann1}[_{Ann2}[\text{therein}]_{Ann2}$ appears to be$]_{Ann1}$ orthologous to the SalK/SalR system of S. salivarius, a salivaricin regulated TCSTS.

Otherwise, the triggers are either exactly agreed upon or completely distinct. The distinct triggers, i.e. those identified by one annotator and not by the other, are not realised linguistically in a different manner than those which were agreed upon. The annotators simply did not agree on considering those cases as suggesting causality.

Argument discrepancies

Cases where the two annotators choose overlapping arguments are more frequent than overlapping triggers, but are still insignificant compared to the number of agreed arguments. There are eight cases of overlapping CAUSE and four of overlapping EFFECT arguments. Examples for both CAUSE and EFFECT are included below, in example (4.8) and example (4.9), respectively.

(4.8) $_{Ann1}[$Results of real-time quantitative RT-PCR also confirmed that, $_{Ann2}[$in the complemented strain CDeltasalKR, only partial genes identified as down-regulated in the mutant rebounded to comparative transcript levels of the wild-type strain.$]_{Ann2}]_{Ann1}$ Those unrecovered genes were probably irrelevant to the bacterial virulence of SS2.

(4.9) The acid tolerance response of Salmonella results in $_{Ann1}[_{Ann2}[$the synthesis of over 50 acid shock proteins (Bearson et al., 1998) that are likely to function primarily when variations in internal pH occur$]_{Ann2}$, i.e. when Salmonella experiences severe acidic conditions (pH approximately 3).$]_{Ann1}$

In example (4.8), the CAUSE arguments chosen by the two annotators overlap. Whilst one annotator considered the entire first sentence as the CAUSE argument, the other expert did not include the first clause, related to the results. Thus, their argument was annotated as "in the complemented strain CDeltasalKR, only partial genes identified as down-regulated in the mutant rebounded to comparative transcript levels of the wild-type strain". After discussions, the two annotators agreed to exclude the clause related to the results, as this is not necessary for the correct interpretation of the stated facts.

In contrast, example (4.9) shows a case of overlapping EFFECT arguments. One annotator considered the effect to be "the synthesis of over 50 acid shock proteins (Bearson et al., 1998) that are likely to function primarily when variations in internal pH occur". The other annotator, however, also included the span of text that further explains and describes the context, "i.e. when Salmonella experiences severe acidic conditions (pH approximately 3)". The selected argument was the extended version annotated by the first annotator, mainly due to the fact that only the specification of the mentioned condition provides biologists with sufficient detail to correctly understand the biochemical processes that occur in the described situation.

Besides overlapping arguments, there are several cases of completely different arguments. More specifically, there are seven cases of disagreed CAUSE arguments and only one case of a disagreed EFFECT argument. As we mentioned above, identifying the CAUSE argument is a much more difficult task than that of identifying the EFFECT

argument. Since this subtask depends on the background knowledge, expertise and interpretation of each annotator, they might have different biomedical points of view on how events connect to each other causally.

In example (4.10), we provide one case in which the two annotators select different text spans for the CAUSE argument of a causal relation.

(4.10) $_{Ann1}$[In the animal model, attenuation of virulence has been noted for Salmonella strains that carry mutations in the pts, crr, cya or crp genes, which encode the general energy-coupling enzymes of the PTS, enzyme IIAGlc of the PTS, adenylate cyclase and cyclic AMP receptor protein, respectively.]$_{Ann1\ Ann2}$[Mlc is a global regulator of carbohydrate metabolism and controls several genes involved in sugar utilization.]$_{Ann2}$
Therefore, it seemed possible that Mlc also affects the virulence of Salmonella.

This is due to the fact that Mlc is closely related functionally to the mentioned list of genes (pts, crr, cya and crp). On the one hand, the first sentence provides a more detailed explanation of the cause without mentioning Mlc, together with the observation of the attenuation of virulence. On the other hand, the second sentence mentions Mlc and the genes in general, but it is not linked to the virulence of Salmonella. Thus, the final decision in this case has the first sentence as the cause, since it includes the virulence of Salmonella and the genes that produce it.

4.11 Comparison to the BioDRB

The major difference between BioCause and BioDRB is the fact the latter allows for discontinuous argument spans, whilst the former does not. This setting increases the

Feature	BioCause	BioDRB
No. of causal associations	850	565
No. of implicit triggers	50 (5.88%)	98 (17.34%)
No. of explicit triggers	800 (94.12%)	467 (82.65%)
No. of tokens per trigger	3.09	2.46
No. of tokens per CAUSE arg.	21.31	31.24
No. of tokens per EFFECT arg.	16.87	20.56
C-_-E	30 (3.52%)	78 (13.80%)
E-_-C	20 (2.35%)	20 (3.53%)
C-T-E	687 (80.82%)	192 (33.98%)
C-E-T	0 (0%)	81 (14.33%)
E-T-C	91 (10.69%)	135 (23.89%)
E-C-T	2 (0.23%)	10 (1.76%)
T-C-E	9 (1.06%)	49 (8.67%)
T-E-C	11 (1.29%)	0 (0%)

Table 4.15: Comparison between BioDRB and BioCause with respect to various measures.

difficulty in automatically determining the argument spans. As we have previously mentioned in Section 2.3.4, the BioDRB contains 542 purely causal relations, as well as 23 relations which are a mixture of causality and other discourse relations. Since the BioDRB and BioCause have somewhat similar sizes, we performed a comparison with respect to some of the previous characteristics. The results are included in Table 4.15. As can be seen, the BioDRB corpus contains a greater number of implicit relations than BioCause. Furthermore, whilst the explicit trigger length is shorter, causal relations in the BioDRB have generally longer cause and effect arguments. The major difference in the order of arguments consists in the lack of the C-E-T pattern in BioCause and the lack of the T-E-C pattern in the BioDRB.

With regard to the distributions of lengths and distances, these are roughly similar in shape when plotted against each other. Figure 4.14 contains the distributions of trigger lengths, Cause and Effect argument lengths and distance between arguments between the BioCause and BioDRB corpora. The distribution of the distance between

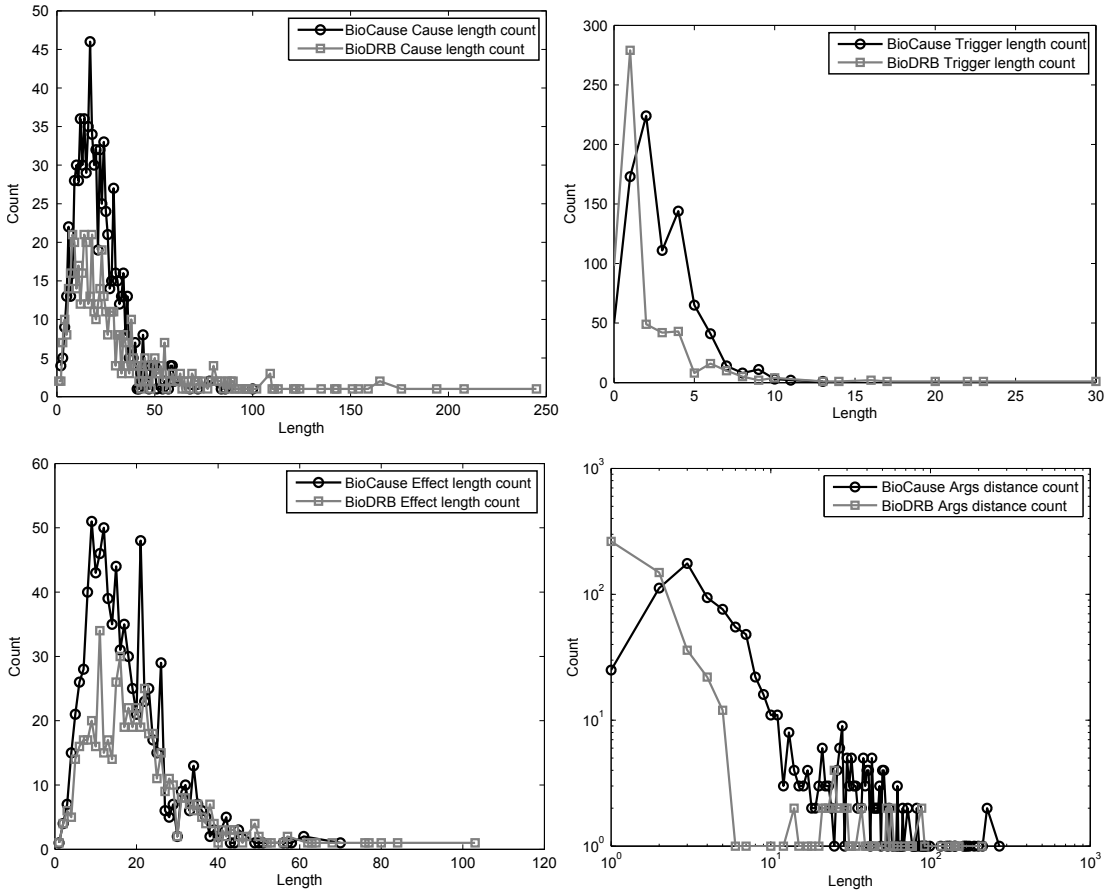


Figure 4.14: Comparison of the distributions of trigger lengths, Cause and Effect lengths and distance between arguments between the BioCause and BioDRB corpora. The distance between arguments is given using a logarithmic scale in order to provide a better view.

arguments is given using a logarithmic scale in order to provide a better view of the graph for small values. It can be noticed that the first three figures are consistent with the data in Table 4.15: in BioCause, the triggers are slightly longer, whilst the arguments are slightly shorter.

4.12 Summary

This chapter has focussed on describing the process of creating BioCause, the first biomedical text corpus specifically addressing the problem of discourse causality. We

have justified the selection of the data source that has undergone the annotation effort, discussing the three main reasons for our choice: the differences between biomedical sublanguages, the interaction of discourse relations with other semantic annotations, and the differences between abstracts and full-body texts. Further to the differences between biomedical sublanguages, we conducted the first study that analyses the performance of the ratio of named entity types as the only discriminant between biomedical subdomains. We thus prove that, by employing only this kind of features, classifiers can successfully distinguish between pairs of subdomains with F-scores reaching values of 97%.

We have detailed the underlying annotation scheme, the employed annotation format and software, and the selection and training of annotators. We have then analysed the annotated discourse causality relations within the corpus. Of interest are the statistics regarding the causal triggers and their arguments. For instance, we noticed both the high ambiguity and high variability of causal triggers, which, on average, occur 2.10 times per trigger type. With respect to causal arguments of triggers, we have shown that about half are located in the same sentence as the trigger, whilst the rest are inter-sentential relations. In this latter case they are found in neighbouring sentences in about 60% of instances, but there are numerous cases with longer distances.

Furthermore, we have looked at the inter-annotator agreement between the two domain experts that have produced the BioCause. We concluded that once the annotators agree on a causal trigger (approximatively 66% F-score), identifying the arguments becomes relatively easy, with high agreement rates (more than 90% F-score for Cause and more than 98% F-score for Effect).

Finally, we have compared the annotations in the newly created resource with those found in the BioDRB corpus.We have showed that these two corpora complement each other in terms of the order of arguments around the trigger. Otherwise, the corpora have similar statistics, i.e. trigger and argument lengths, distances between them etc.

Chapter 5

Causal trigger detection

We conducted the first study on the analysis and identification of discourse causal triggers in the biomedical domain. In this chapter, we provide a detailed account of our analysis and results.

We investigate the key aspects of a machine learning solution to the problem. These include the selection of feature engineering and the choice of learning method and algorithm.

We start by motivating our research and describing the features that are used in the training of machine learners. We employ six types of features, i.e. lexical, syntactic, dependency, command, semantic and positional, and provide production details and motivation for each feature. The features are then tested for relevancy to the task and only a subset is maintained for training learners based on a series of experiments using two attribute evaluators.

Several experiments have been designed in order to find the best learning paradigm, algorithm and settings. The first system relies on rules developed by analysing the data and looking at statistics regarding triggers. Such rules concern dictionaries, syntactic patterns and dependency relations. The performance of these rules is, as expected, low, reaching a maximum of 24% F-score. The second experiment is dedicated to

supervised machine learning: given gold standard labels for all tokens in BioCause, we evaluate the performance of multiple machine learning algorithms. A top F-score of 81.53% is obtained by CRFs. Finally, given the low amount of positive training data available, we employ a semi-supervised approach, i.e. self-training. The algorithm learns by itself starting from a subset of training data, and automatically corrects its mistakes when new data is available. The performance is similar to that obtained in the case of supervised learning, reaching 83% F-score.

The effect of all features used for machine learning is investigated. For every feature, we analyse whether its addition increases or decreases the overall performance of the classifier and the amplitude of this change.

We have evaluated our system on two open access corpora of biomedical discourse causality, BioCause and BioDRB. The performance remains consistent when run on each of the corpora or on their combination. Experimental results show that there is an acute need for more training data, as the learning curve increases with a polynomial trend.

5.1 Motivation

Causal triggers and, more generally, discourse triggers pose two main difficulties when trying to recognise them. First, causal triggers are highly ambiguous. The same tokens in a trigger can also have non-causal meaning on other contexts. One such case is the conjunction *and*, shown in example (5.1), for which the number of non-causal instances (2305) in BioCause is much greater than that of causal instances (1).

(5.1) SsrB binds within SPI-2 *and* activates SPI-2 genes for transcription.

This is the usual case with closed-class part-of-speech words, such as conjunctions and adverbials. Other examples of trigger types more commonly used as causal triggers and belonging to open-class parts-of-speech are *suggesting* (9 causal instances, 54 non-causal instances), *indicating* (8 causal instances, 41 non-causal instances) and *resulting in* (6 causal instances, 14 non-causal instances). For instance, example (5.2) contains two mentions of *indicating*, but neither of them implies discourse causality.

(5.2) Buffer treated control cells showed intense green staining with syto9 (*indicating* viability) and a lack of PI staining (*indicating* no dead/dying cells or DNA release).

This high ambiguity of causal triggers leads to a very high number of false positives and, subsequently, a final low precision.

The second issue in detecting causal triggers is the fact that they are highly variable. There are numerous ways of expressing the same causal trigger, due to the open-class properties of nouns and verbs. Take example (5.3), where the trigger *this result suggests that* indicates a causal relation.

(5.3) The hilE mRNA level measured by real-time PCR also revealed that hilE expression was increased in SR1304 by about 2-fold (Figure 3A).
This result suggests that Mlc can act as a negative regulator of hilE.

The same idea can be conveyed using synonyms of these words, such as *observation*, *experiment*, *indicate*, *show*, *prove*, etc. The high variability reflects in obtaining a low recall, since there will be many false negatives.

To overcome these two issues, we introduce new features which, to some extent, can resolve these problems. More sophisticated structural features are needed in order to better capture the syntactic properties of causal triggers. We use c-command relationships, which have been ignored until now in the task of identifying causal triggers. Moreover, we explore the parse tree both vertically and horizontally in order to provide more information to classifiers regarding the syntax of the sentence. However, although syntax plays a strong role in identifying discourse triggers, even for structural triggers it by no means "aligns" with the discourse structure (Dinesh et al., 2005). Therefore, we also include the semantic context of the triggers into the feature set. We add both general language and biomedical semantic knowledge from WordNet, named entity and event recognisers and UMLS.

5.2 Experimental setup

We experimented with various rule-based and machine learning algorithms and various settings for the task of identifying causal triggers. We have modelled the trigger recognition in three ways.

The first method is rule-based. Three different types of rules, based on lexical, dependency and syntactic features, are combined into five systems. These systems are evaluated on the whole of BioCause.

The second method approaches the problem as a supervised machine learning paradigm. On the one hand, we consider identifying triggers as a sequence labelling task. We have experimented with CRF, a probabilistic modelling framework commonly used for sequence labelling tasks. In this specific task, we employed the CRFSuite implementation[1]. On the other hand, we modelled trigger detection as a classification task, using NB, SVMs and Random Forests (RFs). More specifically, we

[1] http://www.chokkan.org/software/crfsuite

employed the implementation in Weka (Hall et al., 2009; Witten and Frank, 2005) for RFs and NB, and LibSVM (Chang and Lin, 2011) for SVMs. All evaluations are performed in a 10-fold cross-validation setting. Although we analyse the features in the following sections using automatic evaluators, we perform the experiments using all features that have been produced. This decision is based on the fact that we want to observe the impact of all features, without any type of initial selection.

Finally, semi-supervised learning is used to overcome the low amount of gold standard data. We evaluated this method with CRF, RF and SVM classifiers that learn from their mistakes and correct them in the self-training period. Five rounds of evaluations have been undertaken by splitting BioCause into five equally sized distinct subsets. At each round, the learning is performed on four subsets (80% of BioCause), whilst the models are tested on the remaining subset, ensuring that each subset is tested. Furthermore, we create semi-supervised models by using unlabelled data. BioCause is split into two equally sized subsets, one used for seed data, and one for final model evaluation. The self-learning is performed on unseen and unlabelled data comprising 24 full-text articles.

5.3 Feature engineering

Feature engineering and selection is a vital part of any machine learning system. As seen in Chapter 2, various types of features have previously been used for the task of detecting causal triggers, including lexical, syntactic, semantic and statistical (bag of words) features. However, most past work has concentrated around lexical and syntactic features, whilst the semantic aspects of causality (like named entities and events) have been ignored or deemed detrimental to the task in the few cases in which they were considered (Ramesh et al., 2012). In addition to these features, we introduce a new set of features derived from command relationships and position in sentence.

Thus, based on our analysis of causal triggers, we engineered six types of features for the development of this causality model, i.e., lexical, syntactic, dependency, command, semantic and position in sentence. A more detailed description is given in subsequent sections. For brevity, we code the types of features as follows:

- L: lexical
- X: syntactic
- D: dependency
- C: command
- S: semantic
- P: position

5.3.1 Lexical features

The lexical features are built from the actual tokens present in text, and are summarised in Table 5.1. Their utility has been noticed by several researchers (Wellner, 2009; Lin et al., 2012; Ibn Faiz and Mercer, 2013), who state that both the surface level token and its neighbours help towards a correct classification.

The tokenisation and lemmatisation steps are performed by employing the GENIA tagger (Tsuruoka et al., 2005) trained on MEDLINE. The first two features represent the token's surface expression and its lemma. The previously mentioned studies do not mention the use of lemmas in their feature lists, but Katrenko and Adriaans (2007) do in their slightly different task of extracting factual biomedical relations. The inclusion of lemmata is justified by the need of generalisation: some inflected lexemes may occur very rarely (if at all) in the limited amount of training data, and, in a real-world deployment, a learner may be perplexed when encountering them.

ID	Short description	Values
L1	token	8509
L2	*lemma*(token)	5795
L3	*neighbour*(token,[left,right],[1..5])	8509
L4	*lemma*(*L3*)	5795

Table 5.1: Lexical features used in identifying causal connectives.

In contrast, there exists a need for specialisation due to the polysemy and homonymy of words. Take, for instance, Example (5.4), where *and* can refer to both a causal relation (the first occurrence) and an enumeration (the second occurrence).

(5.4) SsrB binds within SPI-2 *and* activates SPI-2 genes for transcription *and* traslation.

It is noticeable how the context affects the meaning of a token and therefore it is necessary to include surrounding tokens in order to allow a learner to differentiate between *and* as a causal trigger or enumerating conjunction. Thus, we included the five tokens immediately to the left and the ones immediately to the right of the current token. In the case of causal triggers, this decision is based on two observations. First, in the case of tokens to the left, most triggers are found either at the beginning of the sentence (311 instances) or are preceded by a comma (238 instances). These two left contexts represent 69% of all triggers. Second, for the tokens to the right, almost 45% of triggers are followed by a determiner, such as *the*, *a* or *an* (281 instances), or a comma (71 instances).

5.3.2 Syntactic features

Syntax is the main provider of features in the literature. Almost all approaches use the PoS and syntactic category of the token and its neighbours (Pitler and Nenkova, 2009;

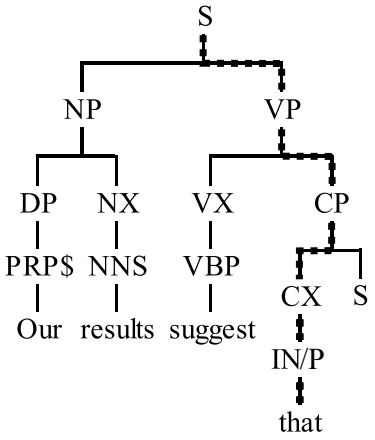


Figure 5.1: Partial parse tree of a sentence starting with a causal trigger.

ID	Short description	Values
X1	*partOfSpeech*(token)	47
X2	*syntCat*(token)	11
X3	*partOfSpeech*(*L3*)	47
X4	*syntCat*(*L3*)	11
X5	*syntCatPathFromRoot*(token)	51811
X6	*syntCatCollapsedPathFromRoot*(token)	21691
X7	*syntCatPositionPathFromRoot*(token)	55850
X8	*ancestor*(token,[1..3])	20
X9	*lowestCommonAncestor*(token,*neighbourOf*(token,left,1))	21
X10	*distanceBetween*(token, *X9*)	37

Table 5.2: Syntactic features used in identifying causal connectives.

Wellner, 2009; Ramesh et al., 2012; Ibn Faiz and Mercer, 2013). Pitler and Nenkova (2009) explore the parse tree horizontally, including the neighbours into the equation. In contrast, Wellner (2009) explores it vertically, deriving features from the path from the root of the parse tree to the token.

The syntax, dependency and predicate argument structure are produced by the Enju parser (Miyao and Tsujii, 2008). Figure 5.1 depicts a partial lexical parse tree of a sentence which starts with a causal trigger, namely *Our results suggest that*. From the lexical parse trees, several types of features have been generated, a list of which is included in Table 5.2.

The first two features represent the PoS and syntactic category of a token. For instance, the figure shows that the token *that* has the Penn Treebank-style part-of-speech *IN* (representing a preposition or subordinating conjunction), whilst its syntactic category is *P*. Moreover, the word *Our* is marked as a possessive pronoun (*PRP$*), *results* as a plural noun (*NNS*), and *suggest* as a non-3rd person singular present verb (*VBP*). Syntactic categories are generalised parts-of-speech, created by removing, e.g., inflection details. A complete list of PoS and syntactic category tags is included in Appendix C. These features are included due to the fact that either many triggers are lexicalised as an adverb or conjunction, or are part of a verb phrase.

For the same reason, the syntactical category path from the root of the lexical parse tree to the token is also included as X5. Because in parse trees there are many cases where constituents will repeat when moving vertically, we collapse X5 into a new feature (X6) by deleting consecutive repetitions of the same syntactic category. For instance, in a path such as *S/VP/VP/V*, the adjacent identical tags *VP/VP* are combined into *VP*, thus creating a collapsed path of *S/VP/V*.

Also based on X5, the path encodes in feature X7, for each parent constituent, the position of the token in its subtree, i.e., beginning (*B*), inside (*I*) or end (*E*); if the token is the only leaf node of the constituent, this is marked differently, using a *C*. Thus, the path of *that*, highlighted in the figure, is *I-S/I-VP/B-CP/C-CX*. Feature X7 has been used before by Ghosh et al. (2011b), whilst Wellner and Pustejovsky (2007) used X5, both in their task of extracting the arguments of discourse triggers in general.

Furthermore, the ancestors of each token to the third degree are instantiated as three different features. This has been found by Ibn Faiz and Mercer (2013) to better generalise the syntactic context of the token than X5, although they restrict it to only the first parent. In the case that such ancestors do not exist (i.e., the root of the lexical parse tree is less than three nodes away), a "none" value is given. For instance, the token *that* in Figure 5.1 has as its first three ancestors the constituents marked with

ID	Short description	Values
D1	*pas*(token)	3241
D2	*pas-role*(token)	2
D3	*pos*(D1)	28
D4	*distanceBetween*(token,D1)	11

Table 5.3: Dependency features used in identifying causal connectives.

CX, *CP* and *VP*.

Finally, the lowest common ancestor in the lexical parse tree between the current token and its left neighbour has been included. The lowest common ancestor of two nodes A and B in a dependency tree is a node L, and there exists no other node N such that L is an ancestor of N. In the previous tree example in Figure 5.1, the lowest common ancestor for *that* and *suggest* is *VP*.

These last two feature types have been produced on the observation that the lowest common ancestor for all tokens in a causal trigger is S or VP in over 70% of instances. Furthermore, the percentage of cases of triggers with V or ADV as lowest common ancestor is almost 9% in each case. Also, the average distance to the lowest common ancestor is 3.

5.3.3 Dependency features

These features are constructed based on the dependency relations found by Enju in the sentence. Table 5.3 includes all dependency features employed in this study.

First, for each token, we extracted the predicate-argument structure and included the arguments as surface expression forms. We also included the PoS of these arguments, as well as the distance from the token.

5.3.4 Command features

Command features are constructed from *command relations* found in the constituency parse tree of the sentence. The concept of command relation was initially introduced by Langacker (1966), who defined it as 'a node X commands a node Y if neither X nor Y dominates the other and the S (sentence) node most immediately dominating X also dominates Y'. A more general definition has been provided by Reinhart (1976), who defined a *constituent command* (*c-command*) by eliminating the restriction of having the node dominating both X and Y being a sentence. Barker and Pullum (1990) relaxed this definition even further, by removing the non-co-dominance condition between X and Y.

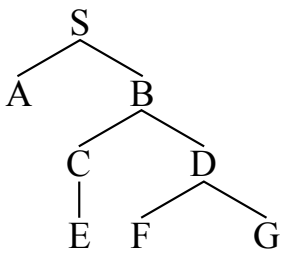


Figure 5.2: c-command syntax tree: A c-commands B, B c-commands A, C c-commands D, D c-commands C etc.

Based on command relations as defined by Barker and Pullum (1990) and exemplified in Figure 5.2, we developed several features, which, to the best of our knowledge, have not been previously used for identifying discourse causal triggers. These are included in Table 5.4.

Features C1-C3 indicate whether the current token c-commands a clause (SBAR), VP or NP constituent, respectively. Features C4-C6 are similar, with the exception that the dominant node must be an S (sentence). In the case of features C7-C9, the dominant node must be a VP.

All mentioned features rely on the observation that a trigger c-commands at least one of its arguments (more specifically, the dependent argument). In most cases, trigger

ID	Short description	Values
C1	*c-commands*(token, SBAR)	2
C2	*c-commands*(token, VP)	2
C3	*c-commands*(token, NP)	2
C4	*S-commands*(token, SBAR)	2
C5	*S-commands*(token, VP)	2
C6	*S-commands*(token, NP)	2
C7	*VP-commands*(token, SBAR)	2
C8	*VP-commands*(token, VP)	2
C9	*VP-commands*(token, NP)	2

Table 5.4: Command features used in identifying causal connectives.

tokens S-command or VP-command argument tokens, whose superparent is usually an SBAR, VP, or NP.

5.3.5 Semantic features

Although the role of semantic features has been previously explored, the results are contradictory. In one study in the biomedical domain, adding a semantic layer lowers the performance of recognising discourse triggers (Ramesh et al., 2012), whilst in the general domain rich compositional semantic information (i.e. VerbNet and CoreLex) manages to produce a statistically significant increase in F-score (Subba and Di Eugenio, 2009). Ramesh et al. (2012) use the BANNER gene tagger and LINNAEUS species tagger to obtain named entity information about genes and species, as well as MetaMap to map text elements to UMLS.

We have exploited several semantic knowledge sources to identify causal triggers more accurately, as a mapping to concepts, named entities and events acts as a back-off smoothing, thus increasing performance. This happens due to the fact that causal triggers do not encode biomedical knowledge, thus tokens recognised as named entities or events should not be recognised as causal triggers. A list of all semantic features in included in Table 5.5.

ID	Short description	Values
S1	*isNamedEntity*(token)	2
S2	*namedEntityType*(token)	9
S3	*isEvent*(token)	2
S4	*eventType*(token)	8
S5	*wordnetHypernym*(token)	1158
S6	*isUMLSEntity*(token)	2
S7	*UMLSEntityType*(token)	126

Table 5.5: Semantic features used in identifying causal connectives.

One semantic knowledge source is the BioCause corpus itself. All documents annotated for causality in BioCause had been previously manually annotated with biomedical named entity and event information. This was performed in the context of various shared tasks, such as the BioNLP 2011 Shared Task on Infectious Diseases (Pyysalo et al., 2011). We therefore leverage this existing information to add another semantic layer to the model. Moreover, another advantage of having a gold standard annotation is the fact that it is now possible to separate the task of automatic causal trigger recognition from automatic named entity recognition and event extraction. The named entity and event annotation in the BioCause corpus is used to extract information about whether a token is part of a named entity or event trigger. Furthermore, the type of the named entity or event is included as a separate feature. Whilst named entities have been employed before (Ramesh et al., 2012), to the best of our knowledge, event information has not.

The second semantic knowledge source is WordNet (Fellbaum, 1998). Using this resource, the hypernym of every token in the text has been included as a feature. This is needed for those tokens which are not specific to biomedicine. Only the first sense of every token has been considered, as no sense disambiguation technique has been employed.

Finally, tokens have been linked to the UMLS (Bodenreider, 2004) semantic types. Thus, we included a feature to say whether a token is part of a UMLS type (S6) and

ID	Short description	Values
P1	*indexInSent*(token)	123
P2	*percentageInSent*(token)	2798
P3	*positionInSent*(token)	3
P4	*length*(*sentence*(token))	94

Table 5.6: Position features used in identifying causal connectives.

another for its semantic type if S6 is true.

5.3.6 Position features

Position features have also been engineered and included in Table 5.6.

First, the location of the token in the sentence is important, as most of the triggers occur in the beginning or middle of the sentence. This feature takes integer values, representing the index in the sentence. However, due to the various sentence lengths in which causality occurs, this may result in data sparseness. Thus, we add a feature which shows the token's index in the sentence percentage-wise. That is, we divide the value of feature P1 by the length of the sentence. To be more discrete, we also add a feature which takes only three values: "Beginning", "Middle", and "End".

Furthermore, the sentence length has been included, as this is correlated with the position: the shorter the sentence, the smaller the chances that a token is part of a trigger in the middle of the sentence.

5.4 Feature analysis

The aim of our investigation was to:

- identify the optimum set of features for the task of identifying causal triggers
- compare the performance of different feature sets by evaluating their individual and combined impact on the overall performance

Many of the features that have been employed in this research are based on the tokens present in text and, thus, the set of values for each feature usually contains several hundred entries, e.g., lexical features and MetaMap features. As would be expected, some values occur more often than others: stop-words are the most frequent, whilst highly specialised concepts appear infrequently. This imbalanced nominal feature set with a high number of values can be confusing for machine learners. Therefore, all nominal features have been transformed into numerous binary features: one binary feature for each value of a nominal feature, only one of which can be true for an instance. However, this results in an extremely high dimensional and sparse feature space, which can be difficult to process and learn from.

To this end, we automatically analysed all binary features to decide which are relevant to our task. We have evaluated the entire feature space using two attribute evaluators, InfoGain and ChiSquare, which are implemented in Weka. Table 5.7 shows the top features from an optimal set, as assigned by InfoGain. ChiSquare offers a similar set of top features, with slight order changes.

The top ten most predictive features relate to surface expression forms and lemmas (three features), syntactic categories (two features), parents in the parse tree (two features), WordNet hypernyms (one feature), index in the sentence features (one feature) and distance to the lowest common ancestor (one feature). These are immediately followed by more diverse features: the common parent constituent with the previous token, MetaMap and named entity information, part of speech, c-command, S-command and VP-command features, and sentence length.

At the other end, there are many lexical features that have almost no predictive power. Be it lemmata or surface expressions, the token under study or its neighbours, biomedical terminology does not help as much as the features previously mentioned. However, semantic features based on biomedical terminology help classification, occurring in the top third of the table.

Feature	InfoGain score
L1_*that*	0.00888
L2_*that*	0.00887
X2_*C*	0.00816
L2_*suggest*	0.00755
X1_*C*	0.00751
D2_*CP*	0.00744
S5_*declare*	0.00727
D2_*NX*	0.00702
P4	0.00634
D4	0.00597

Table 5.7: Top ten predictive features in identifying causal connectives using InfoGain.

5.5 Experimental results

We ran a series of experiments in order to systematically evaluate the effect of the numerous learning algorithms. This section describes the results of our experiments.

5.5.1 Rule-based

Several rule-based baseline systems have been devised based on the observations and analysis of the corpus. The overall performance of all rule systems is included in Table 5.8.

The first baseline is a simple dictionary-based heuristic, named *Dict*. It consists of a lexicon which is populated with all annotated causal triggers and which is then used to tag all instances of its entries in the text as causal connectives. As expected with an approach of this type, the precision of this heuristic is very low, reaching only 8.36%. This leads to an F-score of 15.43%, considering that the recall is 100%. This is mainly due to often occurring words and/or phrases which are rarely used as causal triggers, such as *and*, *by* and *that*.

Based on the observation about the lowest common ancestor for all tokens in a causal trigger mentioned in Section 4.8, we built a baseline system that checks all

Classifier	P	R	F_1
Dict	8.36%	100%	15.43%
Depend	8.05%	76.69%	14.57%
Synt	14.61%	20.45%	17.04%
Dict+Depend	14.47%	74.5%	24.23%
Dict+Synt	21.88%	20.45%	21.13%

Table 5.8: Performance of rule-based classifiers in identifying causal connectives.

constituent nodes in the lexical parse tree for the sentence (S), verb (V), VP and adverb (ADV) tags and marks them as causal triggers. The name of this system is *Depend*. Not only does *Depend* obtain a slightly lower precision than *Dict*, but it also performs worse in terms of recall. The F-score is 14.57%, largely due to the high number of intermediate nodes in the lexical parse tree that have VP as their category.

The third baseline is a syntax-based approach, *Synt*. We extracted the part-of-speech patterns from all triggers, creating a set of 165 unique patterns. For instance, for the trigger *suggesting that*, the part-of-speech pattern is *V-C* (verb-complementiser). We experimented with all possible sets of patterns to search for. The best performing pattern was found to be *V-C*, which occurs in 20.45% of triggers. It gives a precision of 14.61% and a recall of 20.45%, thus resulting in an F-score of 17.04%.

We then combined *Dict* and *Depend*: we considered only constituents that have the necessary category (S, V, VP or ADV) and include a trigger from the dictionary. Although the recall decreases slightly, the precision increases to almost twice that of both *Dict* and *Depend*. This produces a much better F-score of 24.23%. Similarly, the combination of *Dict* and *Synt* results in a precision of 21.88%, a recall of 20.45%, and thus in an F-score of 21.13%.

5.5.2 Supervised learning

In a supervised learning approach, we experimented with several algorithms, which are listed in Table 5.9 together with their top performance.

Classifier	**P**	**R**	**F_1**
CRF	89.99%	74.53%	81.53%
Random Forest	78.45%	67.26%	72.42%
SVM	87.56%	61.60%	72.32%
Naïve bayes	56.95%	80.15%	66.58%

Table 5.9: Performance of various classifiers in identifying causal connectives.

Features	**P**	**R**	**F_1**
L	89.00%	67.09%	76.50%
XDC	92.30%	66.21%	77.10%
LX	86.41%	73.26%	79.29%
LS	89.54%	69.10%	78.00%
XDCS	83.95%	70.78%	76.80%
LXD	87.76%	73.29%	79.87%
LXDCS	89.29%	73.53%	80.65%
LXDCSP	89.99%	74.53%	81.53%

Table 5.10: Effect of feature types on CRF.

As can be observed, we obtained the best performances, in terms of not only F-score, but also precision, in the case of Conditional Random Fields. A slightly lower performance is obtained in the case of Random Forests and Support Vector Machines. Although the F-scores of these two algorithms are similar, SVM has a better precision, whilst RF results in a better recall. Naïve Bayes performs worst, with just over 66% F-score. Its recall, however, is the highest amongst all classifiers, reaching more than 80%. In what follows, we will discuss the performance of each algorithm individually, together with the best performing sets of features. We have included only the top performing feature combinations in order to reduce space.

In the case of CRFs, as can be noticed from Table 5.10, the best performance, in terms of F-score, is obtained when combining all six types of features. The best precision is, however, obtained by using the syntactic, dependency and command features, reaching over 92%, almost 3% higher than when all six feature types are used.

Adding command and semantic features to the feature set increases the precision

Features	P	R	F_1
L	77.12%	67.40%	71.93%
X	68.41%	62.57%	65.35%
S	84.34%	57.25%	67.89%
LX	77.12%	66.75%	71.56%
LS	78.45%	67.26%	72.42%
XS	72.33%	64.20%	68.08%
LXDCS	77.45%	66.23%	71.40%
LXDCSP	76.92%	67.36%	71.82%

Table 5.11: Effect of feature types on Random Forests.

in every case. Looking at the results of LXD and LXDCS, it can be noticed that the precision increases by over 1.5%, resulting in an almost 1% improvement of F-score. Adding positional features further improves the precision by 0.7%, but also increases the recall by 1%.

As can be seen from Table 5.11, the best performance of RFs is obtained when combining lexical and semantic features. Due to the fact that causal triggers do not have a semantic mapping to concepts in the named entity and UMLS annotations, the trees in the random forest can easily produce rules that distinguish triggers from non-triggers. We have experimented with different values for the parameters of the forest: the number of trees and the number of random features. These results are obtained for 21 trees and 10 random features.

As such, the use of semantic features alone produces a very good precision of 84.34%. Also, in all cases where semantic features are combined with other feature types, the precision increases by 0.5% in the case of lexical features and 3.5% in the case of syntactic features. However, the recall of semantic features alone is the lowest. The best recall is obtained when using only lexical features. Nevertheless, the best performance, 72.42%, is still more than 9% lower than that obtained in the case of CRF.

For SVMs, we have experimented with two kernels, namely polynomial (second

Features	P	R	F_1
L	81.23%	60.76%	69.51%
X	82.19%	56.74%	67.13%
S	85.69%	57.15%	68.56%
LXD	86.74%	54.28%	66.77%
LS	87.56%	61.60%	72.32%
XCS	84.22%	55.67%	67.03%
LXS	88.10%	54.35%	67.22%
LXDCSP	89.25%	57.33%	69.81%

Table 5.12: Effect of feature types on SVM.

Features	P	R	F_1
L	57.42%	70.67%	63.35%
X	51.25%	76.45%	61.36%
S	44.33%	67.88%	53.63%
LX	51.78%	81.34%	63.27%
LS	57.89%	71.00%	63.77%
XS	51.75%	77.20%	61.96%
LXS	52.00%	81.11%	63.37%
LXDCSP	54.00%	80.65%	64.69%

Table 5.13: Effect of feature types on Naïve Bayes.

degree) and radial basis function (RBF) kernels. For each of these two kernels, we have evaluated various combinations of parameter values for cost and weight. Both these kernels achieved similar results, indicating that the feature space is not linearly separable and that the problem is highly complex.

The effect of feature types on the performance of SVMs is shown in Table 5.12, where the used kernel is polynomial. As can be observed, the best performance is obtained when combining the lexical and semantic feature types (72.32% F-score). Nevertheless, the best precision is produced by the combination of all feature types, whilst the best recall is obtained by combining lexical and semantic features. Actually, the addition of command and semantic features again improves precision: by more than 2% in XCS, and by 2.5% in the case of LXDCSP. In this latter case, the recall also increases thanks to the positional features.

Finally, Table 5.13 shows the performance of the Naïve Bayes classifier. This algorithm has obtained the worst precision in all tests, which resulted in the lowest F-score. The recall, however, is very high, reaching more than 80% in some cases.

As we expected, the majority of errors arise from sequences of tokens which are only used infrequently as causal triggers. This applies to 107 trigger types, whose number of FPs is higher than the number of TPs. In fact, 64 trigger types occur only once as a causal instance, whilst the average number of FPs for these types is 14.25. Such errors are also found by Ibn Faiz and Mercer (2013), for whom 62% of the erroneous cases are due to highly ambiguous triggers such as *and*, *as*, *if* and *when*.

5.5.3 Semi-supervised learning

For the supervised classification part of semi-supervised learning (SSL), we have employed CRFs, RFs and SVMs, as they have performed best in the experiments described in the previous section. As for the heuristics used in case no instance is classified with a confidence greater than τ, we have used several rule-based routines. We consider for marking as labelled instances only those which have the confidence in the top 5% of all confidences. We then filter these instances and select only those which have several feature values that were deemed important by the experiments discussed in Section 5.6. These include the lemma of the token (L2), the predicate-argument structure links of the token and ancestor constituents (D1, D2), its c-command and VP-command values (C1-C3, C7-C9), and named entity information (S1, S5, S6).

The lemma has to be part of a lexicon of lemmas contained in causal triggers that is pre-compiled. At least one of the ancestor constituents must be either a VP, NP or S. The token must c-command or VP-command a VP or NP. Furthermore, the token must not bear any biomedical meaning. These rules are given equal weights, and each token must comply with at least two of the rules in order to be considered as labelled

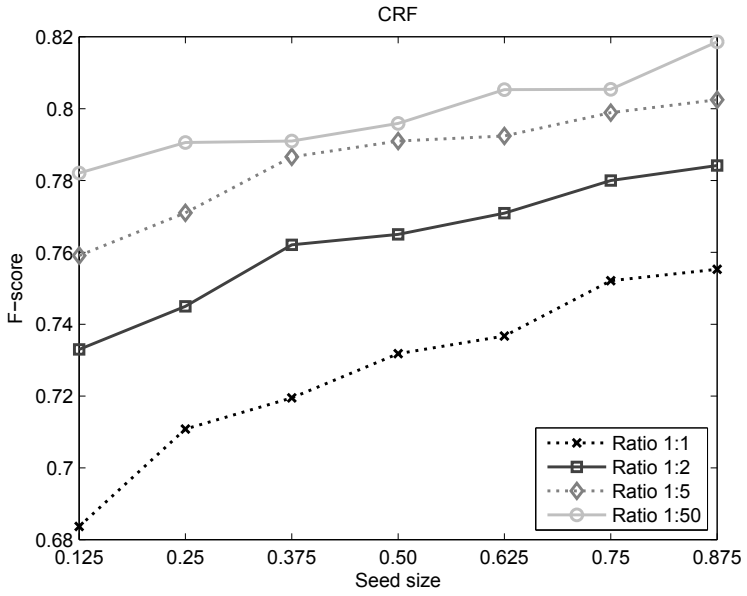


Figure 5.3: Self training results for causal trigger identification at $\tau = 0.6$ using CRF.

correctly.

We developed several models, each with a different size for the initial labelled set, Λ, as well as a different ratio of positive to negative instances in Λ. The data for the experiments comes from two different sources. First, we tested our approach with gold data from BioCause only. However, the size of BioCause is rather small for semi-supervised methods. Therefore, we created another evaluation where we use BioCause for the gold seed and test, whilst the learning is performed on unlabelled data. Both these experiments are detailed in the following subsections.

Experiments on BioCause

For training and evaluating our approach, we split the data in BioCause into five, equally sized folds. One fold (20% of BioCause) was used for testing the final model, whilst the other four (80% of BioCause) were used for the self-training part. Thus, the experiment has been repeated five times for each variation, ensuring that each fold

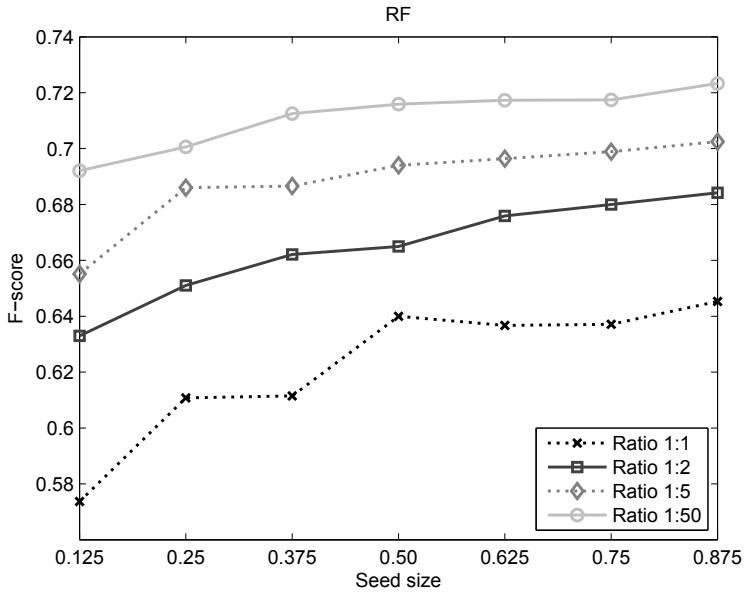


Figure 5.4: Self training results for causal trigger identification at $\tau = 0.6$ using RF.

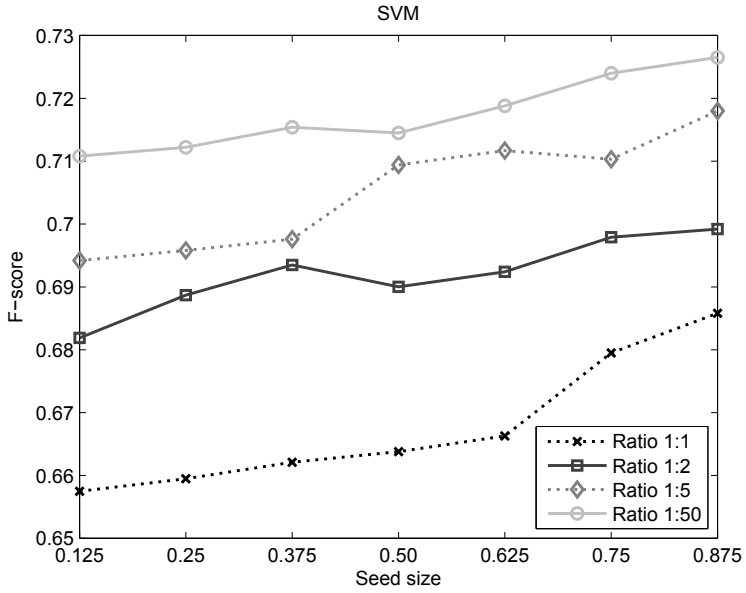


Figure 5.5: Self training results for causal trigger identification at $\tau = 0.6$ using SVM.

is becomes a test fold. The average performance for each variation is given in Figures 5.3, 5.4 and 5.5.

On the one hand, we trained models with different sizes for the seed labelled sets Λ. There are seven models, varying in the percentage of positive instances from 12.5%

to 87.5%, in steps of 12.5%, extracted from the self-training part of the corpus. On the other hand, we changed the ratio of positive to negative instances in each labelled set. The ratios are 1:1, 1:2, 1:5, and the actual ratio in BioCause, approximately 1:50.

As can be noticed, all models have a generally increasing trend, showing that the amount of gold standard training data is essential to this task. Furthermore, the learning curve does not turn into a plateau when a high percentage of data is available for training. This suggests that the performance could be improved if more data were available. The top results, when the seed size is 87.5%, are similar to those obtained by employing supervised algorithms (summarised in Table 5.9).

The difference in F-scores between the various ratios for small amounts of gold training data is large. However, this decreases progressively as the amount of training data is increased. More specifically, when the amount of data is only 12.5% of the total size of BioCause, the F-score ranges from 57% to 69% for RF, 66% to 71% for SVM, and 68% to 78% for CRF. At the other end, for the highest amount of data available, the F-score varies between 64% and 71% for RF, 69% to 73% for SVM, and 76% to 82% for CRF. This effect has also been noticed by Hernault et al. (2010), who employ a semi-supervised approach to exploit the co-occurrence of features in unlabelled data in infrequent discourse relation classification. The reason for this is that for a small number of training instances, the number of unseen features in the testing data is large. When more training data is added at each step, the number of unseen features diminishes and the learning curves of all the models tend to converge. This shows that there is no need to learn models by giving high amounts of negative examples. These examples will most probably be repetitive and will not affect the final performance.

The best F-scores are obtained when the ratio is the natural ratio. Actually, the closer the ratio is to the natural one, the better the performance. Training a model on an artificially created corpus, that does not reflect the natural balance, will affect

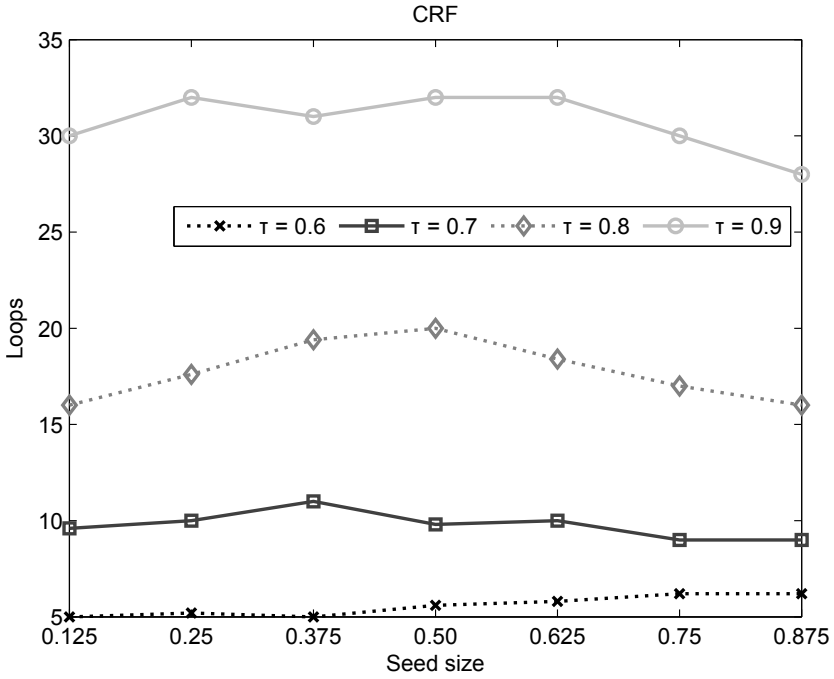


Figure 5.6: Number of self-training loops when varying τ for the natural ratio using CRF.

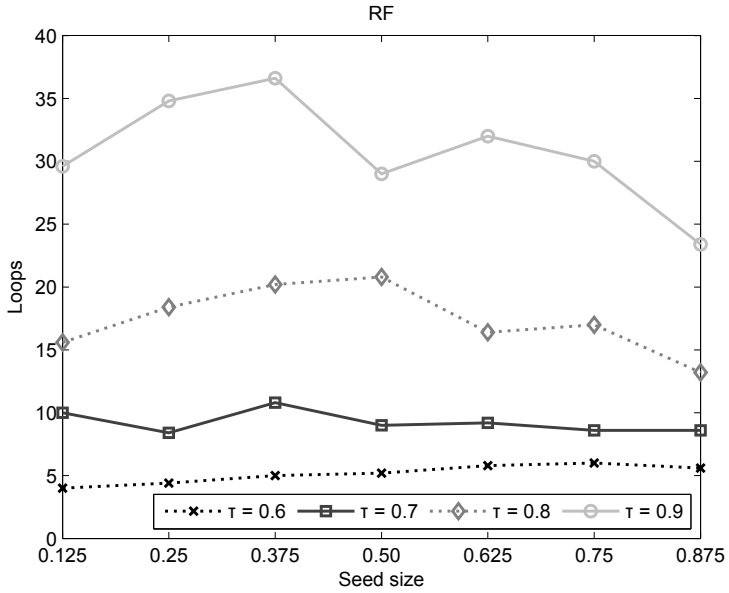


Figure 5.7: Number of self-training loops when varying τ for the natural ratio using RF.

its performance in a real-world situation. The model becomes less strict the more balanced the data is, and will thus produce more false positives. In the case of 1:1 ratio,

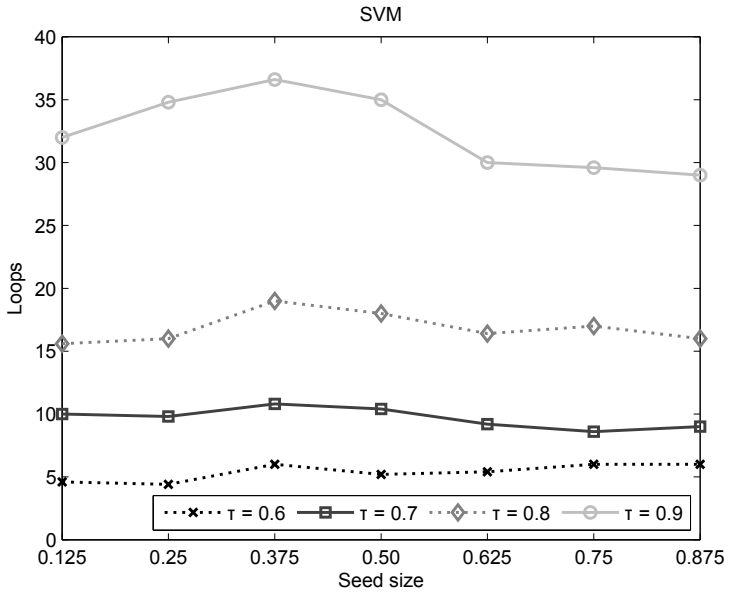


Figure 5.8: Number of self-training loops when varying τ for the natural ratio using SVM.

the recall of the model is very high, reaching values of more than 90%. The precision, however, is extremely low, varying between 10% and 20%. As the seed ratio is shifted towards the natural ratio, the precision and the recall become more balanced: precision increases and recall decreases, but with an overall increased F-score.

We have also run experiments with different values for the threshold τ. This parameter affects both how quickly the classifier learns the model and its quality. A small value for τ will result in a fast convergence with many false positives treated as positive labelled data, thus leading to a lower final score. Conversely, a higher value lengthens the time needed for convergence, but the model should be more accurate. However, this could also lead to using the heuristics more often, if no instance is classified with high confidence. This will introduce more errors into the model, which will see a drop in performance. Therefore, we tested values ranging from 60% to 90% in increments of 10% for the seven seed set sizes above and measured the number of loops each model needs for convergence.

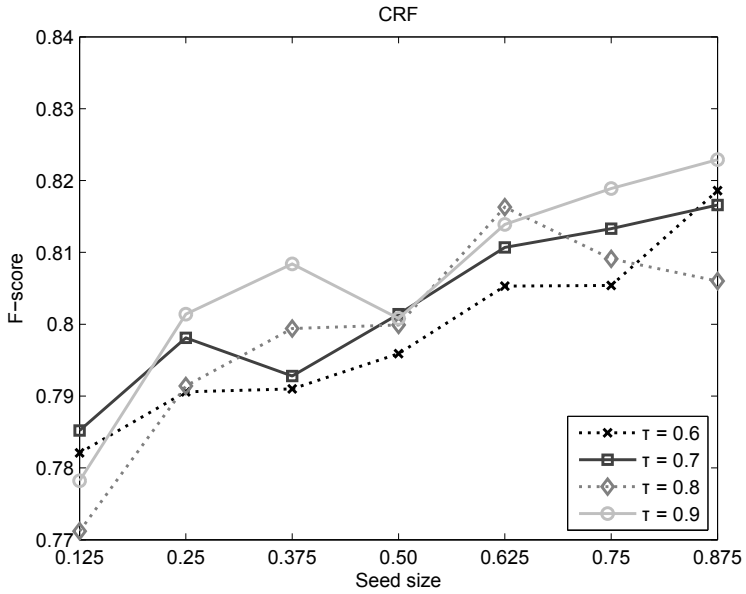


Figure 5.9: Self training results when varying τ for the natural ratio using CRF.

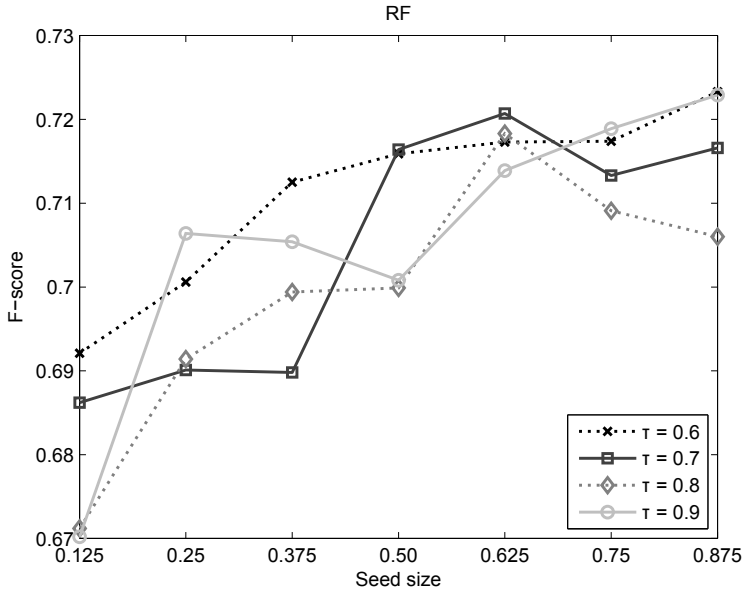


Figure 5.10: Self training results when varying τ for the natural ratio using RF.

As can be seen in Figures 5.6 - 5.11, the convergence time varies significantly for different values of τ, but the F-score does not. Increasing τ results in an increasing number of loops needed for convergence over all seed sizes. The difference in the number of loops for small seed sizes is much larger than in the case of large seed sizes.

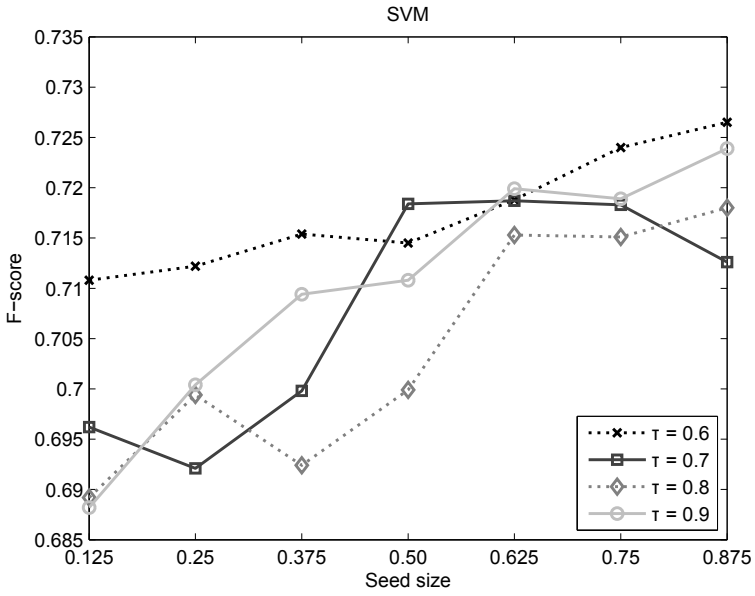


Figure 5.11: Self training results when varying τ for the natural ratio using SVM.

This is because of the large number of unseen features in a small seed, which results in low confidence classifications. Thus, very few instances are moved to the seed, whilst many instances will be kept as unlabelled for the next loop.

Furthermore, increasing τ does not significantly increase the F-score of the self-trained model. Although it is to be expected to have fewer confident classifications as τ increases, this does not happen. This can be explained by the low frequency and high variability of causal triggers. Classifications are made with similar levels of confidence, regardless of the amount of training data. However, the more training data is given, the more correct classifications are made.

Experiments on unlabelled data

In this case, we have mixed the gold-standard data with unseen and unlabelled data. The gold standard data is represented by BioCause. We have split BioCause into two equally sized sets. One set is used for the seed set, whilst the other is used for the final model evaluation. The experiment is then repeated with swapped sets.

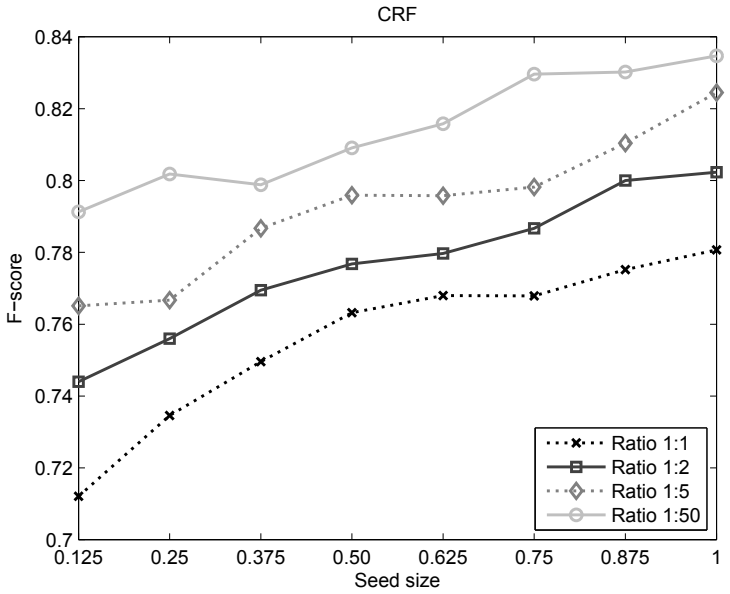


Figure 5.12: Self training results for causal trigger identification at $\tau = 0.6$ using CRF.

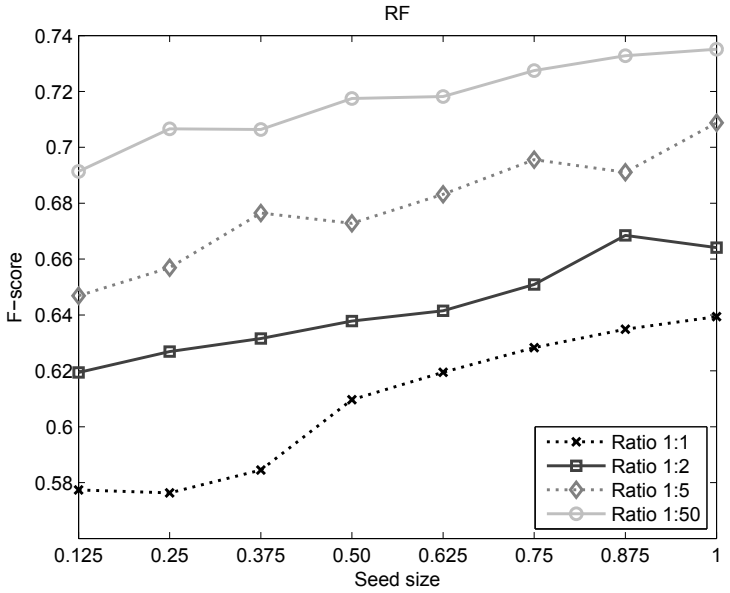


Figure 5.13: Self training results for causal trigger identification at $\tau = 0.6$ using RF.

The unlabelled data consists of 24 full-text open-access journal articles also on infectious diseases. Unlike BioCause, they do not contain any type of gold standard annotations. All features that are used in the experiments, i.e. lexical, syntactic, dependency, command, semantic, and position, are derived from fully automatic parses.

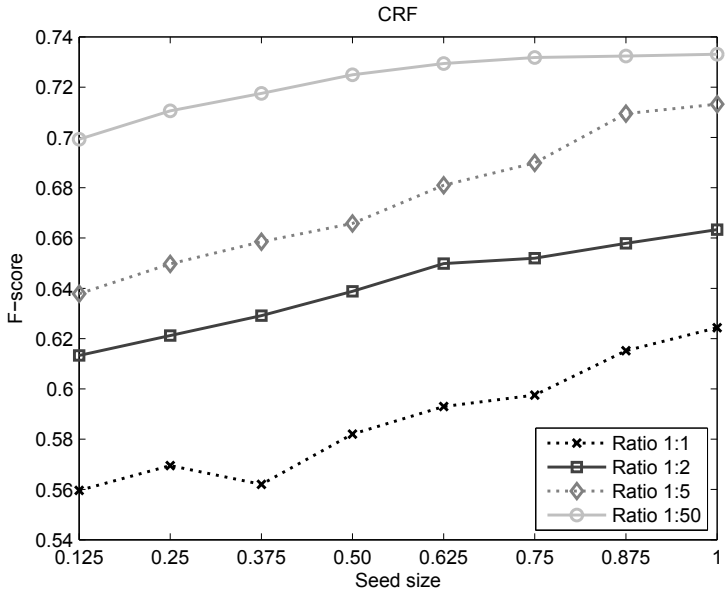


Figure 5.14: Self training results for causal trigger identification at $\tau = 0.6$ using SVM.

Our analysis follows the same structure as in the previous section. We investigate the effect of both parameters in the self-learning method, i.e. the confidence threshold τ and the size of the gold seed Λ. In contrast to the previous section, the seed size has another value, 100%, as the learning is now done on unlabelled data.

Figures 5.12 - 5.14 show the performance of the three classifiers at $\tau = 0.6$. As can be observed, the learning curve is similar to that depicted in the previous section. The performance improves slightly, to 83% F-score, in the case of CRF, whilst for RF and SVM it revolves around 74%.

The first observation is that the learning time for each loop increases considerably due to the larger amount of data that needs to be processed into a model. The number of learning loops increases significantly in the case where the seed size is very small. As can be noticed in Figures 5.15 - 5.17, the necessary number of learning loops is much larger than in the previous experiment, whilst the performance increases only slightly. At the other end, when a large amount of data is available as seed, the training time decreases considerably.

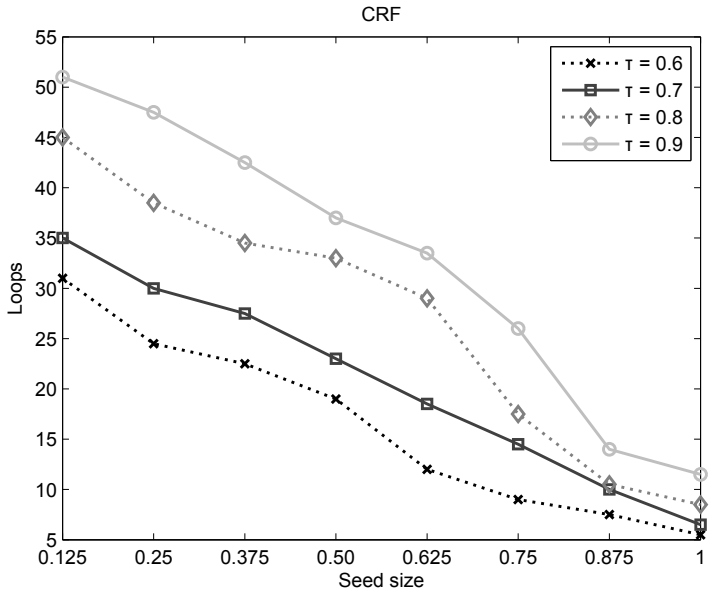


Figure 5.15: Number of self-training loops when varying τ for the natural ratio using CRF.

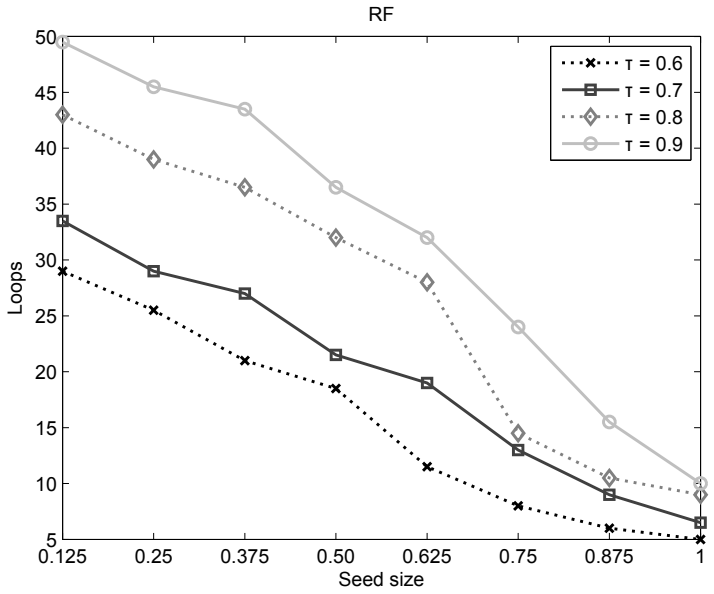


Figure 5.16: Number of self-training loops when varying τ for the natural ratio using RF.

Comparing these figures with those obtained for the experiment on BioCause, it can be seen that the curves have different slopes. Whilst the lines were previously stable throughout the increase of the seed size, in this case they drop. Thus, we investigated

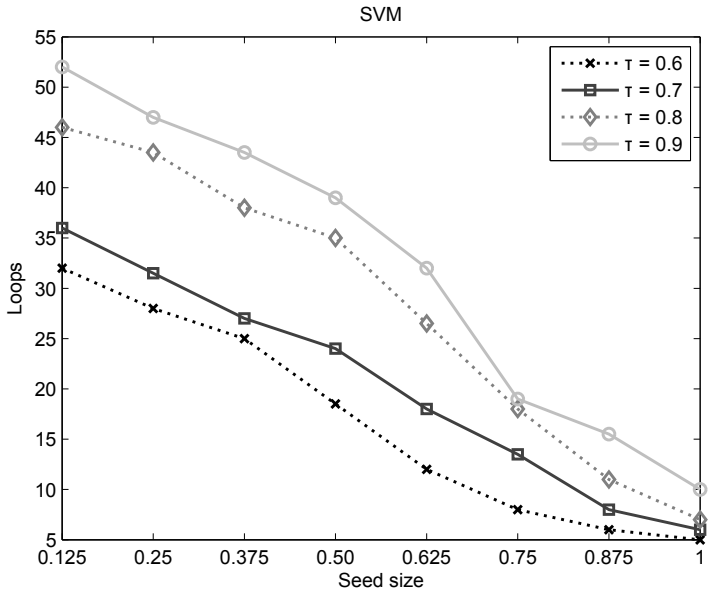


Figure 5.17: Number of self-training loops when varying τ for the natural ratio using SVM.

the process taking place inside each loop. In the first experiment, the learning takes place at a constant rate of unlabelled data being classified with a confidence greater than the threshold regardless of the seed size. In this case, when the seed size is small, only few instances are classified with a higher-than-τ confidence in each loop, thus resulting in a large number of loops. When the seed size increases, the classifier becomes more and more confident, and thus more and more instances are added to the labelled group.

The threshold τ again does not affect the resulting performance, similar to the experiment on BioCause. Varying τ from 0.6 to 0.9 confidence yields similar F-scores for all three classifiers, as can be noticed in Figures 5.18 - 5.20.

5.6 Effect of features

We have also studied the usefulness of the numerous features that we have engineered. Whilst in the previous section we showed the effect of some combinations of different

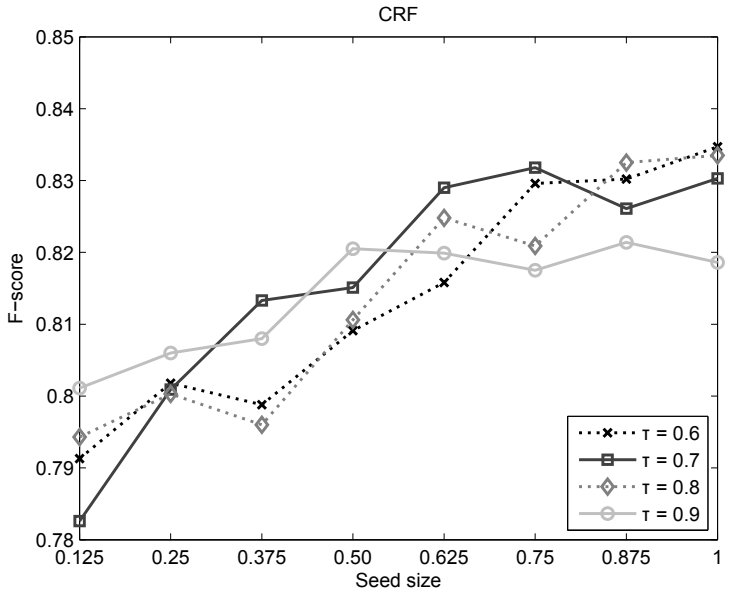


Figure 5.18: Self training results when varying τ for the natural ratio using CRF.

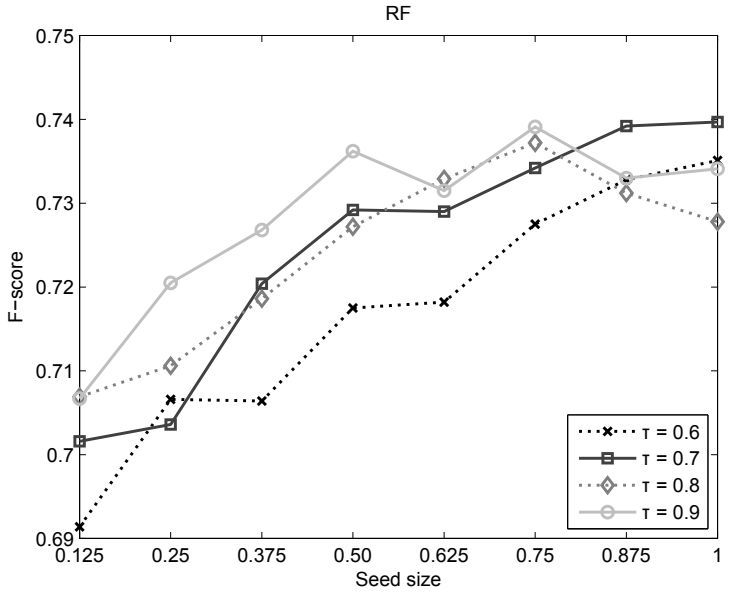


Figure 5.19: Self training results when varying τ for the natural ratio using RF.

feature types on CRFs, RFs, SVMs and NB, we will now look at how helpful are the features for the best performing algorithm, CRF. The following subsections discuss the behaviour of each feature type and its interaction with the other features types. The tables show the percentage of feature combinations where by adding that specific

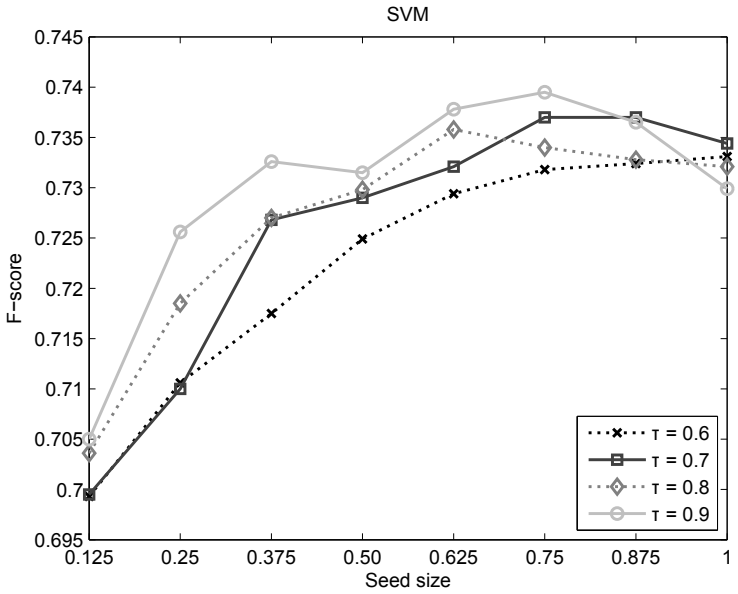


Figure 5.20: Self training results when varying τ for the natural ratio using SVM.

feature the performance is improved in terms of F-score. Not included in the table are the individual values for precision and recall; these will be commented in the text. Furthermore, both the increase and decrease in performance by adding a feature are statistically significant for an a = 0.05 with a p-value of almost 0, unless otherwise stated in the text, using Student's t-test.

5.6.1 Lexical features

Lexical features are by far the most helpful in determining whether a token is part of a trigger or not. Table 5.14 shows that the lemma of a token (feature L2) improves the classification in almost 94% of the feature combinations, whilst the lemmata of the neighbouring tokens (feature L4) improve it in more than 88% of the cases. Slightly less helpful, but still around 80%, are the actual tokens in the text, of both the token under classification and the context tokens. This is to be expected, as the use of lemmata is a simple way to obtain generalisation.

Furthermore, the average increase in F-score for lemmata is more than 3.2%, whilst

ID	Usefulness	Av. increase	Av. decrease
L1	78.37%	2.75%	0.73%
L2	93.73%	3.47%	0.54%
L3	82.68%	2.98%	0.68%
L4	88.42%	3.23%	0.59%

Table 5.14: Usefulness of lexical features in identifying causal connectives.

for tokens is just under 3%. The average decrease varies between 0.5% and 0.75%. Both precision and recall are much higher in the case of lemmata than in the case of tokens. The precision increase revolves around 4%, whilst the average increase in recall is about 2.5%.

5.6.2 Syntactic features

The usefulness of syntactic features is included in Table 5.15. As can be noticed, the percentage of combinations which are improved by syntactic features lies mostly between 60% and 70%, with the exception of X5, the syntactic category path from the token to the root with position information, which is just over 51%. The average increase is also rather low, varying between 0.75% and 1.75%.

The large difference between features X5/X6 and X7 is due to data sparsity. By encoding the position in the subtree besides the path from the root, feature X7 becomes too information rich and its values occur rarely. X5 and X6, in contrast, encode less information, thus they have a lower number of possible values which occur more frequently.

Precision-wise, all features improve classification in less than 50% of the cases. The PoS and syntactic category (X1 and X2) are actually quite detrimental to most cases, increasing performance in less than 30% of the combinations. The average increase is also much lower than the average decrease.

ID	Usefulness	Av. increase	Av. decrease
X1	67.74%	1.74%	0.77%
X2	64.00%	1.35%	0.79%
X3	65.26%	1.45%	0.74%
X4	61.32%	1.23%	0.58%
X5	58.65%	1.17%	0.74%
X6	59.23%	1.21%	0.75%
X7	51.29%	0.75%	0.60%
$X8_2$	65.98%	1.23%	0.58%
X9	52.33%	0.78%	0.61%
X10	53.71%	0.54%	0.62%

Table 5.15: Usefulness of syntactic features in identifying causal connectives.

Regarding the ancestors of the token in the parse tree, the usefulness differs according to the level of the ancestor. In the case of the second ancestor, the performance increases in 65.98% of the cases where this is added. However, the first and third ancestors only improve almost 36% of the cases each. This is mainly related to the structure of the tree and that of the causal triggers. Regardless of the level, all three ancestors increase mostly precision (75% in case of level two and 50% in case of levels one and three), as it selects as triggers only those nodes that have the appropriate ancestry.

Features X9 and X10, the lowest common parent and the distance to it, increase recall slightly more than precision, resulting in an overall improvement of around 53%.

However, recall benefits the most from syntactic features. All features result in recall improvement of around 1.50% in more than 70% of feature combinations. If not improved, the combination performance is decreased with less than 0.50% on average.

5.6.3 Dependency features

Table 5.16 includes the usefulness of all dependency features employed in this study. Feature D1, the surface expression of the arguments which are dependent on the current token, is rather helpful, increasing both precision and recall in around 71% of the cases.

ID	Usefulness	Av. increase	Av. decrease
D1	71.58%	0.43%	0.48%
D2	65.23%	0.14%	0.15%
D3	72.67%	0.44%	0.47%
D4	67.28%	0.39%	0.51%

Table 5.16: Usefulness of dependency features in identifying causal connectives.

ID	Usefulness	Av. increase	Av. decrease
C1	71.02%	1.01%	0.61%
C2	81.56%	1.15%	0.60%
C3	75.23%	1.03%	0.51%
C4	62.23%	0.85%	0.41%
C5	65.10%	0.74%	0.49%
C6	66.62%	0.97%	0.62%
C7	75.58%	0.95%	0.48%
C8	79.75%	1.28%	0.63%
C9	77.49%	1.08%	0.59%

Table 5.17: Usefulness of command features in identifying causal connectives.

Similar values are obtained for the PoS of these arguments. In contrast, the distance between the arguments and the token is not that helpful. It increases the performance in about a third of the cases, but the average decrease is much higher than the increase.

5.6.4 Command features

The nine command features provide significant information to the classifier, according to the data in Table 5.17. As can be observed, the most useful features are c-command and VP-command, where the commanded constituent has the syntactic category VP or NP. These help in more than 70% of the feature combinations, and have high average increase and low average decrease values.

The S-command features (C4-C6) also help in the classification task, but not as much as the rest. They improve only 62-66% of cases by about 0.85%. This can be explained by the fact that, although a high proportion of triggers S-commands SBARs, VPs or NPs, there are also many non-triggers which S-command the same syntactic

ID	Usefulness	Av. increase	Av. decrease
S1	65.77%	0.31%	0.64%
S2	41.36%	0.48%	0.78%
S3	64.56%	0.75%	0.60%
S4	52.50%	0.45%	0.74%
S5	72.15%	2.13%	0.90%
S6	57.64%	0.48%	0.64%
S7	33.87%	0.60%	0.76%

Table 5.18: Usefulness of semantic features in identifying causal connectives.

categories. Thus, the S-command feature does not provide as much information to the classifier as the other command features.

5.6.5 Semantic features

A list of the usefulness of semantic features is included in Table 5.18. WordNet hypernyms (feature S5) seem to be the most helpful feature – it improves classification in more than 72% of the cases. The average increase is also high, reaching more than 2%, whilst the average decrease is only 0.90%. This feature is a very good method of improving recall, which increases in more than 78% of the cases by 2.13%. Precision, however, is increased in only 37% of feature combinations, with average increase and decrease of almost 2%.

The other semantic features do not help as much as S5. One thing to consider is that the features referring to whether or not a token is part of a named entity or an event (S1, S3, S6) and those which give the specific entity/event type (S2, S4, S7) are not independent. Since S1, S3 and S6 are binary features, it is only when they have the value of 1 that features S2, S4 and S7 have a value too. This value further specifies the type of the event, but it will be missing in case S1, S3 and S6 are 0. That said, the binary features improve classification in about 60% of the cases, whereas the multi-valued features in only 30-40%. In all cases, recall is improved much more by binary

ID	Usefulness	Av. increase	Av. decrease
P1	64.92%	0.21%	0.14%
P2	68.86%	0.23%	0.15%
P3	65.48%	0.20%	0.14%
P4	54.26%	0.15%	0.11%

Table 5.19: Usefulness of position features in identifying causal connectives.

features, whereas it is precision which increases in the case of the multi-valued ones.

5.6.6 Position features

The effect of the two position features is listed in Table 5.19. Feature P1, the index of the token in the sentence, is rather useful, as it tells a classifier that the further a token is located in the sentence, the less chances it had of being a trigger. It improves classification in about a third of the feature combinations, but its effect is limited: both average increase and decrease are very small, of 0.21% and 0.14%, respectively.

When the index is relative to the sentence length (feature P2), the usefulness increases to almost 69%, showing that there might be a data sparseness issue. The three-valued position in the sentence, P3, obtained similar scores to the previous two: over 65% usefulness, around 20% increase and 14% decrease averages.

Using the length of the sentence, P4, has an even smaller impact, improving classification in just over 54% of the cases. Its average increase and decrease are 0.15% and 0.11%, respectively, showing the low overall effect.

5.7 BioCause v. BioDRB

We have also evaluated our optimal feature set using the BioDRB corpus. This corpus differs from the BioCause corpus in one important aspect: it does not contain any semantic annotation related to named entities or events. This means that, for the purpose

Type	Europe PMC	NeMine	OSCAR
Gene	✓	✓	
Protein	✓	✓	
Gene—Protein	✓		
Disease	✓	✓	
Drug	✓	✓	
Metabolite	✓	✓	
Bacteria		✓	
Diagnostic process		✓	
General phenomenon		✓	
Indicator		✓	
Natural phenomenon		✓	
Organ		✓	
Pathologic function		✓	
Symptom		✓	
Therapeutic process		✓	
Chemical molecule			✓
Chemical adjective			✓
Enzyme			✓
Reaction			✓

Table 5.20: Named entity types and their source.

of conducting experiments on the BioDRB in a similar manner, we need to include a pre-processing step that recognises named entities.

For this, we used a simple method that augments the annotation with the named entities present in the output of three named entity recognition tools, i.e., MetaMap, NeMine and OSCAR. The types of entities in the output by each of the three tools, together with the NE types present in Europe PMC, are summarised in Table 5.20.

After augmenting the existing NEs by running the three NER tools on the corpus, the outputs were combined to give a single "silver" annotation list. This operation was performed by computing the mathematical union of the three individual annotation sets, as shown in Eq. 5.1.

$$\mathbb{A}_{\mathsf{Silver}} = \mathbb{A}_{\mathsf{MetaMap}} \cup \mathbb{A}_{\mathsf{Oscar}} \cup \mathbb{A}_{\mathsf{NeMine}} \cup \mathbb{A}_{\mathsf{UKPMC}} \tag{5.1}$$

Train	Test	P	R	F_1
BioCause	*10X*	84.64%	67.30%	74.98%
BioDRB	*10X*	85.52%	65.18%	73.97%
BioCause	BioDRB	69.58%	60.65%	64.80%
BioDRB	BioCause	75.50%	56.34%	64.52%
BioCause+BioDRB	*10X*	79.23%	66.21%	72.13%

Table 5.21: Results of the evaluation with BioDRB.

For reasons of fairness, the gold standard semantic annotation in the BioCause corpus has been removed and replaced with automatic NER results.

For the evaluation, we used the best performing algorithm and its parameter settings, i.e., CRF with all six types of features. We created different models and evaluated them in various ways, and the results of these tests are given in Table 5.21. The first two columns of the table show the training corpus and the test corpus, respectively, for that respective test. In the case of 10-fold cross validation, *10X* is used.

As can be observed, the model trained on the BioDRB corpus obtains a slightly higher precision than the one trained on BioCause. This is mainly due to the smaller set of unique connectives present in BioDRB. The recall is, however, 2% lower, and, overall, the F-score for the BioDRB model is 1% lower than the F-score for the BioCause model. The results obtained from BioCause are different from those given in Table 5.9 due to the fact that the semantic annotation is different. Instead of gold standard annotations, we employed automatic NE labels and the performance dropped significantly.

The second type of evaluation is a cross validation between the two corpora: training is carried out on one and testing on the other. In the first case, we trained a model on BioCause and tested it on BioDRB. The second case is the opposite, training on BioDRB and testing on BioCause. There are significant differences in precision and recall between the two tests, but the resulting F-scores are approximately equal. The precision is lower in the first case by 6%, whilst the recall is 4% lower in the second

because of the wider variety of causal triggers that are present in BioCause and do not occur in BioDRB.

Finally, we trained CRF on the combination of the BioCause and BioDRB corpora. The results of the 10-fold cross-validation are slightly worse than those achieved for each of the individual corpora, but much better than for the cross evaluation between the two corpora. It can also be noticed that both precision and recall are moderately lower than those obtained for each of the two corpora.

5.8 Effect of corpus size

We also hypothesised that an increase in the size of the training data increases the performance. To this end, we extracted random subsets of the combined corpus at various percentages. For each of the six corpus sizes, varying from 50% to 100% in intervals of 10%, we created five random subsets. These subsets have been 10-fold cross-validated using the best performing algorithm and its parameter settings, CRF with all six types of features.

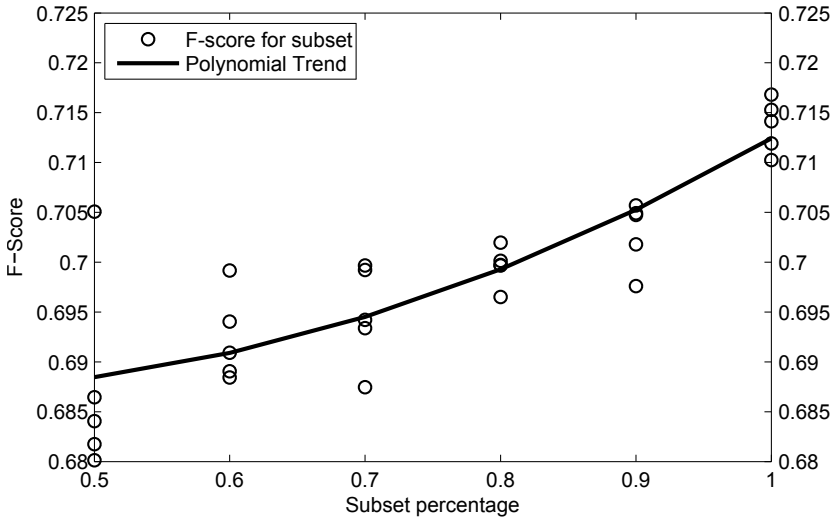


Figure 5.21: Distribution of F-scores for each evaluated subset of the combined BioCause and BioDRB corpus.

Figure 5.21 shows the F-score achieved for each of the 30 evaluated subsets with circles. As can be noticed, the results tend to have higher values as the amount of data increases. These results are also similar to those obtained in Section 5.5.3, where supervised classification is applied to subsets of a corpus containing BioCause and BioDRB.

Also depicted in the figure is a thick black line that shows the second-degree polynomial increase of the F-score trend. This is based on the averages obtained for each subset percentage. The co-efficient of determination, R^2, whose formula is given in Equation 5.2, and which shows how closely the trendline fits with the data points, has the value of 0.9761, indicating that the trend line is very reliable.

$$R^2 \equiv 1 - \frac{\sum_i (y_i - \hat{y})^2}{\sum_i (y_i - \bar{y})^2} \tag{5.2}$$

where $\hat{y}_i$ is the estimated value for a subset, whilst $\bar{y}$ is the average value for that subset.

Furthermore, we tested the statistical significance of this increase by using the Anova Single Factor test. At an α of 0.05, we obtained an $F_{statistic} = 15.12$, much larger than the corresponding $F_{crit} = 2.62$, a fact which rejects the null hypothesis that all the F-scores are equal in favour of the alternate hypothesis that at least two of the means are different. The resulting p-value is 9.53E-7, which again allows us to reject the null hypothesis. Taken together, these results strengthen our hypothesis that the more data there is, the better the system performs.

Figure 5.21, corroborated by Figures 4.8 and 5.18, shows that there exists an urgent need for more gold standard data.

5.9 Discussion

The learning models presented in this chapter come as a result of the objectives proposed in Chapter 1. The most important results are listed in Tables 5.7, 5.8 and 5.9, and Figures 5.12 and 5.18. They prove that causal triggers can be successfully recognised in biomedical scientific literature, with over 80% F-score.

There are three major factors to be considered when automatically recognising triggers: the chosen algorithm, the selection of features, and the corpus for training. They are all discussed in the following sections.

5.9.1 Comparison of algorithms

The experiments performed and discussed in the previous sections show that a semi-supervised approach yields the best F-score. More specifically, employing a supervised Conditional Random Fields reaches an F-score of 81.53%, whilst Random Forests and Support Vector Machines perform worse by almost 10%. In contrast, a semi-supervised approach produces slightly higher results. In our case, employing the self-training method on BioCause using CRFs results in a top F-score of 79.26%, whilst RFs and SVMs reach values of over 72%. If the learning is performed on unlabelled data, the performance increases to 83.47% in the case of CRFs, and to almost 74% in the case of RFs and SVMs.

These results are much lower than those that are obtained in the open domain. Pitler and Nenkova (2009), for instance, achieve results as high as 91% F-score using Naïve Bayes on automatic parses when identifying discourse triggers in general, whilst Lin et al. (2012) obtain 93.62% F-score. Ibn Faiz and Mercer (2013) further improve the results to 96.22% F-score.

However, assigning senses to the relations seems to be more difficult. The F-score of Lin et al. (2012) reaches only 80%, whilst Pitler and Nenkova (2009) perform a

level 1 type sense assignment and obtain 94% F-score. In the level 1 type classification, Causality is part of the Contingency class, together with Pragmatic Cause, Condition and Pragmatic Condition. Thus, if we consider these two steps as leading to the same goal as our task, then by multiplying the two results (93.62% and 80%) we get a performance of around 75%, less than the one described in this chapter. Nevertheless, when applying a model trained on BioDRB on the PDTB corpus, similarly worse results are obtained (Ramesh et al., 2012). This shows that in-domain classifiers outperform cross-domain classifiers and that biomedical scientific discourse is truly different and more difficult to capture automatically.

Both CRFs and SVMs have been used before in detecting biomedical discourse triggers, although they have not been trained on causality specifically. Ramesh et al. (2012) experimented with these two algorithms on BioDRB, and concluded that the CRF model outperformed the SVM model by 10%, producing a final F-score of 75.70%. This result is similar to the 73.97% F-score obtained by our CRF model cross-validated on BioDRB, considering that we focussed only on causal relations, which are more difficult to identify than general discourse relations. More recently, the same corpus has been used by Ibn Faiz and Mercer (2013), who applied their extended feature set with ME classifiers and achieved a performance of 82.36% F-score. Again, they make no distinction between the various discourse relations and treat them as a whole.

However, the Random Forest algorithm has not been used before for this task.

As regards semi-supervised approaches, the literature is not very vast, and does not contain any work on biomedical data. Our self-training method is, to the best of our knowledge, the first semi-supervised approach of this type applied to discourse connective recognition. A different approach is that of Hernault et al. (2010), who prove that feature vector extension is a promising method to improve classification accuracy for infrequent discourse relation types. Evaluating it on RSTDT and PDTB, the method increases the baseline F-score by more than three times in some cases for

discourse causality, to 18.7%. However, as the authors themselves admit, this method cannot be used by itself in discourse analysis due to its low performance.

Do et al. (2011) develop a minimally supervised event causality identification methodology, which employs a measure of cause-effect association between two given events and their arguments. They obtain an F-score of 38.60% on PDTB, but this increases to 41.70% when joint inference is performed with discourse relation predictions from ILP.

Having compared our results to the current state-of-the-art, we consider our supervised and semi-supervised Conditional Random Fields to improve on it in biomedical discourse causal trigger recognition.

5.9.2 Comparison of features

As concerns features, we noticed through our experiments that the best performance is obtained when using all types of features. This includes domain independent features, such as syntactic, dependency and command features, but also domain specific features, such as biomedical semantics. In fact, semantics plays a very significant role in the task of recognising causal triggers. They improve the classification in most feature combinations, and increase the performance by 2.13% on average.

Subba and Di Eugenio (2009) reach the same conclusion when experimenting on instructional texts. By adding semantics on top of their existing feature set, the performance of recognising cause:effect relations increases by 8.52%, to 19.05% F-score. Their semantic resources are VerbNet, for verbs, and CoreLex, for nouns. Although there are a couple of relations whose classification accuracy drops (act:reason, step1:step2), discourse relations generally benefit, to some extent, from this addition.

On biomedical text, Ramesh et al. (2012) employ mostly orthographic features and just a few syntactic features. They also include named entity information obtained

from UMLS and ABNER, but conclude that it damages the overall performance. More specifically, the F-score drops by between 1% and 7.5%, depending on the semantic feature source. In their case, recall is most affected, with variations of even 10%, whilst precision is relatively constant, but still falling with up to 3%. Ibn Faiz and Mercer (2013) suggest that the reason behind semantics damaging the performance of Ramesh et al. (2012) is the fact that ABNER already uses orthographic features, which thus get duplicated in the feature vector.

As Ibn Faiz and Mercer (2013) also suggest in their error analysis, there are cases of discourse triggers which cannot be captured by using only surface level and syntactic features, and instead need some sort of semantic understanding of the context. By checking the children of the dominant SBAR of the trigger for temporal senses, they manage to slightly increase the performance with 0.18%. Our richer semantic features add much more than that.

In conclusion, all feature types are needed and complement each other. Whilst lexical features are the most indicative of causal triggers, syntax and semantics permit generalisation over the grammatical flexibility and sense variability of language.

5.9.3 Comparison of corpus size

The size of the corpus is always a real problem for machine learning methods. As has been noticed in Section 5.8, the learning curve on a combined corpus of 43 full-text journal articles, containing more than 1300 causal relations, is increasing even when all data is given as input. This correlates with the results obtained by employing a self-learning algorithm.

Since most existing work has focussed on the general domain, the PDTB corpus has been the main resource for gold standard data. PDTB has 18459 manual annotations of discourse relations, which are triggered by only 100 unique trigger types. Unlike

it, BioCause contains only 800 explicit causal relations, whilst the unique trigger set comprises 381 phrases. These large differences pose significant issues when creating discourse parsers for a specialised domain such as biomedicine.

As Ramesh et al. (2012) mention, most errors arise from the fact that a large part of trigger phrases occur only a small number of times. The low frequency is not enough in order for the machine learner to create accurate models, especially in the case of 10-fold cross validation. For those triggers which occur only once, the trigger will be either in the training set, or in the test set, case which will result in low performance.

The main result of this experiment is the fact that more data is needed for such specialised domains.

5.10 Summary

The chapter presented our proposed framework in the area of identifying discourse causal triggers. We conducted the first detailed analysis of the problem of identifying discourse causality triggers in biomedical text given gold standard annotations.

Our analysis showed that the ability of a word or phrase to act as a causal trigger depends not only on the context and domain of text, but also on the annotation and information perspective (e.g., linguistic v. biological perspective).

In terms of feature selection, our results showed that lexical, syntactic and dependency features are more important, while command, semantic and position features are less significant. Nonetheless, the best results were achieved by a combination of all six types of features.

We have applied an array of algorithms, including rules, supervised machine learning and self-training. We discovered that, for this task, the Conditional Random Fields algorithm consistently outperforms the other learning algorithms. By combining the best solutions for each of the above aspects, we created a novel framework for the

identification of causal triggers.

We evaluated our system on the two open access corpora of discourse causality mentioned above. Our 10-fold cross-validated results on the BioDRB corpus were similar to those obtained on the BioCause corpus. The performance drops by approximately 10% when training a model on one corpus and testing it on the other. Furthermore, we trained a model on the combination of the two corpora, whose performance is slightly lower than that of the models trained on BioCause and BioDRB separately. This is mainly because of the lack of gold standard semantic annotations.

Finally, we proved the need for more annotated data by running a series of learning procedures using different sizes for training data. As we increase the amount of available data, the performance increases too.

Chapter 6

Argument detection

This chapter focusses on analysing and automatically identifying the spans of the two arguments of the previously recognised causal triggers.

First, we motivate our research by explaining the usefulness of capturing the causal arguments and demonstrating the difficulty of the task because of the numerous possibilities of expressing them.

Second, we describe the process of identifying the two arguments of the causal trigger, which is divided into three steps. In the first step, a classifier is built in order to determine whether the two arguments are located in the same sentence or not, based on the trigger. In the second step, based on the result of the previous step, two spans representing the arguments are located around the trigger, either in the same sentence or neighbouring sentences. The last step deals with giving a sense to the newly found causal relation by assigning roles to the two arguments: cause and effect.

In our approach, we employ multiple types of features, namely lexical, syntactic, dependency, command, semantic and positional. We describe each feature individually and justify its selection for our specific task. We then test these features for relevancy using automatic feature evaluators and retain only a subset of these for learning purposes.

Our experiment structure is similar to that used in Chapter 5. We employ rules, supervised machine learning, and semi-supervised machine learning. The performance of these three paradigms are analysed and compared to decide which is the best approach for this task.

We also investigated the effectiveness of the features that we created. We analyse, feature by feature, how useful they are in the classifications that the machine learners perform.

6.1 Motivation

The arguments of causal relations, cause and effect, are more difficult to recognise than causal triggers. This is due to multiple reasons, most of which are detailed in Section 4.9. We will briefly reiterate and exemplify them here, in order to explain the decisions we made.

First, the spans of text that make up the arguments are of arbitrary length, varying significantly from one case to another, as previously depicted in Figure 4.10. Arguments can go up to 100 tokens in length in the case of Cause, and up to 70 in the case of Effect.

Second, the position of the two arguments around the trigger can change, as shown in Table 4.11. Although most of the relations follow a Cause-Trigger-Effect pattern, there is an important percentage of relations, 20%, which do not obey this rule. Furthermore, Table 4.12 shows that almost half of all relations have one argument in a different sentence than that of the trigger. Thus, the search space increases significantly and, as a consequence, the difficulty of a correct recognition increases too.

This leads to the third reason, which concerns the distance between the trigger and the arguments. Figure 4.11 illustrates the number of sentences between that of the trigger and that of the independent argument, when it is located in a different sentence.

Require: trigger set T
Ensure: arguments for each trigger in T
1: **for all** trigger $t \in T$ **do**
2: Label t as SS or DS
3: **if** t is SS **then** {arguments in same sentence}
4: Split sentence in clauses
5: Label the immediate right clause of t as DepArg
6: Label the rest of the sentence as IndArg
7: **else** {arguments in different sentences}
8: Label sentence of t as DepArg
9: Identify IndArg around the sentence of t
10: **end if**
11: Identify argument roles
12: **end for**

Figure 6.1: Pseudocode for identifying causal arguments.

About half of the cases have the argument located in the previous sentence, but the rest spread up to the tenth previous sentence.

Because of these three reasons, we divide our process into three steps. Thus, we deal with only one of these problems at a given point of time. We are aware that if one of these steps produces noisy models, the erroneous classifications will be propagated further down in the pipeline. This can lead to lower final performance. However, in a real-life situation, this is the process that would occur.

6.2 Experimental setup

Figure 6.1 depicts the pseudocode for identifying the arguments of the previously recognised causal triggers. There are three steps, each based on the output of the previous. The first step is to determine the position of the arguments (AP). We are interested to know whether the two arguments are located in the same sentence or in different sentences. This is due to the fact that syntax plays an important role in the former case, but no role in the latter. This classification is based mainly on the causal trigger itself.

The second step then locates the actual spans of the two arguments (AS). Including the causal trigger as a feature for the system, the system first locates the syntactically dependent argument. This is the easier argument to detect, since it is bound syntactically to the trigger. The more elusive independent argument is then located by including both the trigger and dependent argument as features.

Finally, after both the spans of arguments are found, a role is given to each of them (AR). This is again done mostly having the trigger as a feature, but also semantic features based on the arguments.

Each of the three steps is tackled with various rule-based and machine learning algorithms and different settings. Similar to the trigger detection in Chapter 5, we have modelled each step of the argument recognition in three ways.

The first method is rule-based. Three different types of rules, based on lexical, dependency and syntactic features, are combined into five systems. These systems are evaluated on the whole of BioCause.

The second method approaches the problem as a supervised machine learning paradigm. For finding the argument spans, we experimented with CRF, considering the task a sequence labelling problem. We also employed SVM, RF and NB, when modelling the task as a classification problem. As in the previous chapter, we use the CRF-Suite implementation of CRF, LibSVM for SVM, and Weka for RF and NB. For deciding on the position and roles of the two arguments, we employ six classifiers belonging to different categories of learners. They are all implemented in the Weka framework.

Finally, semi-supervised learning is used to overcome the low amount of gold standard data. We evaluate the same classifiers for each of the three steps, using the best-performing classifiers to tag the unlabelled data after each step.

ID	Short desription	Values	AP	AS	AR
L1	t	8509	✓	✓	✓
L2	*lemma*(t)	5795	✓	✓	✓
L3	*isCapitalised*(t)	2	✓	✓	✓
L4	*neighbour*(t,[left,right],1..5)	8509	✓	✓	✓
L5	*lemma*(L4)	5795	✓	✓	✓

Table 6.1: Lexical features in identifying causal arguments.

6.3 Feature engineering

Based on our analysis of causal triggers, we engineered six types of features for the development of this causality model, i.e., lexical, syntactic, dependency, command, semantic and position in sentence. A more detailed description is given in subsequent sections. However, we describe only features that have not been introduced in the previous chapter, or for which we have a different motivation. All tables summarising the features in each category also show in which of the three steps the feature is used. In the case of the first and third steps (i.e. AP and AR), the features are constructed based on the trigger, whilst for the second step, AS, the features are constructed at the token level.

6.3.1 Lexical features

Several lexical features have been engineered for this classification task, and they are listed in Table 6.1.

One of the best features is the token or causal trigger itself, L1. For instance, when the trigger is the token *Thus* (i.e., *thus* with a capital first letter), it is highly probable that the current sentence is an effect of a previous sentence. Thus, the causal relation is marked as DS. For generalisation purposes, we also include the lemmatised form of the trigger, L2. Thus, *these results suggest that* is represented as *these result suggest that*.

ID	Short desription	Values	AP	AS	AR
X1	*partOfSpeech*(token)	47		✓	
X2	*syntCat*(token)	11		✓	
X3	*posString*(trigger)	228	✓		✓
X4	*syntCatString*(trigger)	165	✓		✓
X5	*posStringDupl*(trigger)	226	✓		✓
X6	*syntCatStringDupl*(trigger)	141	✓		✓
X7	*containsMainVerb*(trigger)	2	✓		✓
X8	*mainVerb*(sent)	896		✓	
X9	*voiceOfVerb*(trigger)	2			✓
X10	*pos*(L4)	47		✓	
X11	*syntCat*(L4)	11		✓	

Table 6.2: Syntactic features in identifying causal arguments.

Furthermore, a useful feature is a flag saying whether the trigger starts with a capital letter or not, L3. This again helps in the decision for the position of the trigger in the sentence. Finally, the neighbours of the triggers and their lemmata also count towards this decision, and are coded as L4 and L5, respectively.

6.3.2 Syntactic features

As for syntax, we include PoS and syntactic category strings representations of the causal triggers (X3 and X4, respectively). For instance, a trigger such as *These results show that* is represented as a PoS string *DT-NN-V-DT*. This adds a level of generalisation, where (usually) nouns and verbs can be replaced by their numerous synonyms.

These two features are then extended by creating other strings which do not contain duplicate consecutive PoS or syntactic category values, marked as X5 and X6. In other words, *DT-NN-V-V-DT* is reduced to *DT-NN-V-DT*. This simplifies the string representation and reduces the data sparsity. A sequence of adjectives or compound verb tenses should not affect the causal relation.

We also add a feature, X7, indicating whether the trigger contains the sentence's main verb. If it does, this is a good indicator that the arguments are located in different

ID	Short description	Values	AP	AS	AR
D1	*pas*(token)	3241		✓	
D2	*pas-role*(token)	2		✓	
D3	*pos*(D1)	28		✓	
D4	*distanceBetween*(token,D1)	11		✓	

Table 6.3: Dependency features used in identifying causal connectives.

sentences. Furthermore, feature X8 contains the main verb of the sentence. We are also interested in the voice of the verb, which is included as feature X9. This is helpful in determining the direction of the relation: which predicate affects which?

Finally, we extract the first two features for the neighbouring tokens as well. These are coded as X10-X11.

6.3.3 Dependency features

These features are constructed based on the dependency relations found by Enju in the sentence. Table 6.3 includes all dependency features employed in this study.

First, for each token, we extracted the predicate-argument structure and included the arguments as surface expression forms. We also included the PoS of these arguments, as well as the distance from the token.

6.3.4 Command features

Command features, built on the definition provided in the previous chapter, are included in Table 6.4.

Features C1-C3 indicate whether the current token c-commands a SBAR, VP or NP constituent, respectively. Features C4-C6 are similar, with the exception that the dominant node must be an S (sentence). In the case of features C7-C9, the dominant node must be a VP.

All mentioned features rely on the observation that a trigger c-commands at least

ID	Short description	Values	AP	AS	AR
C1	*c-commands*(token, SBAR)	2		✓	
C2	*c-commands*(token, VP)	2		✓	
C3	*c-commands*(token, NP)	2		✓	
C4	*S-commands*(token, SBAR)	2		✓	
C5	*S-commands*(token, VP)	2		✓	
C6	*S-commands*(token, NP)	2		✓	
C7	*VP-commands*(token, SBAR)	2		✓	
C8	*VP-commands*(token, VP)	2		✓	
C9	*VP-commands*(token, NP)	2		✓	

Table 6.4: Command features used in identifying causal connectives.

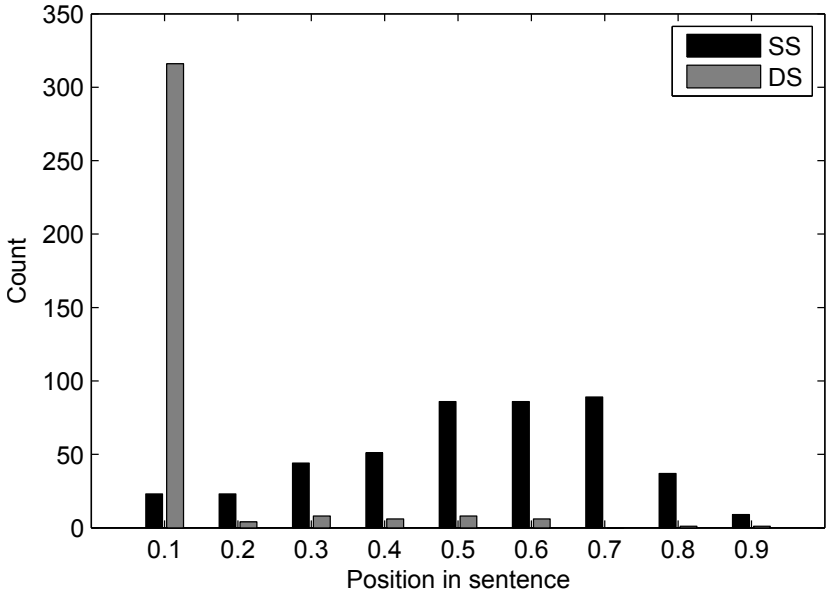


Figure 6.2: Procentual location of triggers in sentences showing the location of its two arguments.

one of its arguments (more specifically, the dependent argument). In most cases, trigger tokens S-command or VP-command argument tokens, whose superparent is usually an SBAR, VP, or NP.

6.3.5 Positional features

The position of the trigger in the sentence is also of great importance. As can be noticed in Figure 6.2, an initial trigger suggests that the arguments are located in different

ID	Short desription	Values	AP	AS	AR
P1	*indexInSent*(trigger)	56	✓	✓	✓
P2	*percentageInSent*(trigger)	295	✓	✓	✓
P3	*positionInSent*(trigger)	3	✓	✓	✓
P4	*length*(*sentence*(trigger))	72	✓	✓	✓

Table 6.5: Positional features in identifying causal arguments.

sentences, whilst a trigger in mid-sentence tends to have both arguments around it in the same sentence.

Thus, the position of the trigger in the sentence is a good indicator of the location of the arguments. Therefore, we include the position of the trigger in the sentence, P1. Furthermore, we added a feature which indicates, percentually, the position of the trigger in the sentence, P2, and one which discretises it into three values, Beginning, Middle, and End (P3). In order to relativise the position in the sentence, we included another feature containing the length of the sentence, P4.

6.3.6 Semantic features

The semantic features employed in detecting the arguments of a causal trigger are similar to those discussed in Chapter 5.

We exploit the same sources of semantic knowledge, namely the pre-existing gold standard annotation in BioCause, automatic named entity and event information from OSCAR, NeMine, and UKPMC, UMLS semantic types and WordNet hypernyms. In addition to these, we introduce another four semantic features, S8-S11.

The other two new features, S8 and S9, record the decisions made by the systems in previous steps. For instance, feature S8 is used in the second and third step of our pipeline, and shows whether or not a token has been marked as a trigger. Similarly, S9 is used only in the last step and shows whether or not a token has been marked as being part of the dependent argument.

ID	Short desription	Values	AP	AS	AR
S1	*isNamedEntity*(token)	2		✓	
S2	*namedEntityType*(token)	9		✓	
S3	*isEvent*(token)	2		✓	
S4	*eventType*(token)	8		✓	
S5	*wordnetHypernym*(token)	1158		✓	
S6	*isUMLSEntity*(token)	2		✓	
S7	*UMLSEntityType*(token)	126		✓	
S8	*isTrigger*(token)	2		✓	
S9	*isDA*(token)	2		✓	
S10	*type*(DA)	143			✓
S11	*type*(IA)	143			✓

Table 6.6: Semantic features in identifying causal arguments.

Finally, features S10 and S11 characterise the semantic properties of the dependent and independent argument, respectively. The semantics contained in the two arguments can help in determining which is the cause and which is the effect.

6.4 Feature analysis

To this end, we automatically analysed all binary features to decide which are relevant to our task. We have evaluated the entire feature space using two attribute evaluators, InfoGain and ChiSquare, which are implemented in Weka. The tables show the scores of InfoGain. Nevertheless, ChiSquare offers a similar set of top features, with slight order changes.

Table 6.7 shows the top features from an optimal set, for the task of determining the position of the arguments. As can be noticed, the most discriminant features are the trigger and its lemma. These are followed by the PoS string and its duplicateless version, as well as the syntactic category string and its duplicateless version. The information about the position of the trigger in the sentence is also of great relevance, both the absolute and percentual positions being present in the top ten. The flag corresponding to the capitalisation of the trigger also plays a significant role, as well as

Feature	InfoGain score
L1	0.90591
L2	0.84038
X1	0.76788
X5	0.76382
X2	0.68507
X6	0.64695
P2	0.63723
P1	0.62523
L3	0.58323
X1_*DT*	0.3258

Table 6.7: Top ten predictive features in identifying the position of arguments.

Feature	InfoGain score
X8_*suggest*	0.02020
P2	0.01897
P4	0.01719
P1	0.01570
X8_*indicate*	0.00909
X2_*PN*	0.00869
X6_*PN*	0.00869
X11_*l,PN*	0.00620
X11_*r,-EOS-*	0.00518
S8	0.00518

Table 6.8: Top ten predictive features in identifying the span of arguments.

whether the trigger contains a determiner.

In what regards the task of determining the span of the arguments, Table 6.8 shows the top ten features. It can be noticed that the top does not contain any lexical feature. The amount and diversity of tokens make it very difficult for these features to be discriminative. One of the best features is the main verb of the sentence, X8, when it is *suggest* or *indicate*. Positional features are also very important, as the absolute and percentual index of the token in the sentence, as well as the length of the sentence occupy the second, third and fourth places. Other important features are the presence of punctuation (*PN*) in the syntactic category string and its duplicateless version, and the presence of punctuation and end of sentence markers in the immediate left and right,

Feature	InfoGain score
X2_*P*	0.0836
X9_*active*	0.08212
L1_*by*	0.0602
L1_*due*	0.04742
L1_*suggest*	0.04307
X1_*IN*	0.04284
X5_*IN*	0.04284
X1_*V*	0.04199
L2_*be*	0.04199
L2_*that*	0.04124

Table 6.9: Top ten predictive features in identifying the role of arguments.

respectively, context. The tenth best feature is the flag which marks a token as part of a trigger.

For the task of determining the role of the arguments, Table 6.9 shows the top features. The features are spread across the lexical and syntactic types. The most important feature is the syntactic category *P* included in the trigger, immediately followed by the *active* voice of the verb in the trigger. The next three best features are lexical, and each flags the presence of *by*, *due* and *suggest*, respectively, in the trigger. These features can easily distinguish between the roles of arguments, since *due* will usually introduce the cause, whilst an *active* voice and *suggest* are specific to a following effect argument. The other features in this top refer to containing an *IN* or *V* in the part-of-speech string and its duplicateless version, as well as the lemmata *be* and *that* in the trigger.

6.5 Experimental results

We ran a series of experiments in order to systematically evaluate the effect of the numerous learning algorithms and features for all three steps. This section describes the results of our experiments.

Rule	P	R	F_1
Position	91.80%	90.59%	91.19%

Table 6.10: Performance of rules in classifying triggers as SS or DS.

6.5.1 Argument location identification

As we observed in the causal argument analysis section in Chapter 4, the relative position of the two arguments is roughly balanced. Table 4.12 showed that almost 56% of the causal relations in BioCause are intra-sentential, whilst the rest of 44% are inter-sentential. This means that syntactic dependency is an advantage in only half of the cases, where both arguments are located in the same sentence. In case the two arguments are located in different sentences, other types of features are needed, since syntax does not influence the position anymore. For instance, one could use semantics and position features to establish possible causal links. Therefore, it is necessary to decide whether the two arguments are located in the same sentence or not, as different methodologies would be applied afterwards. The following three subsections describe the three approaches to this task, i.e. rules, supervised learning and semi-supervised learning.

Rule-based

We have engineered a rule-based system to address the task of locating the two arguments. Table 6.10 lists its performance, which is detailed in what follows.

The position of the trigger in the sentence is a very good estimator of whether the two arguments are located in the same sentence of in different sentences. This is due to the fact that the trigger tends to be placed in between the two arguments. Thus, the first rule-based system decides that the arguments are located in different sentences if the trigger is at the beginning of the sentence, and in the same sentence otherwise. This approach leads to a very high F-score value of 91.19%. The main errors arise

Classifier	P	R	F_1
Naïve Bayes	91.85%	91.90%	91.87%
SVM	92.75%	92.55%	92.65%
JRip	93.20%	92.95%	93.07%
J48	93.00%	92.80%	92.90%
RandFor	92.70%	92.80%	92.75%
Vote	94.95%	94.65%	94.80%

Table 6.11: Performance of various algorithms in classifying triggers as SS or DS.

from the cases where the sentence begins with a trigger, and both arguments follow it in the same sentence. For instance, example (6.1) shows one case in which the trigger is followed by both arguments. This will result in a misclassification by our rule.

(6.1) $Since_T$ [Brucella is an intracellular facultative pathogen]$_{DA}$, [the bacteria could use these denitrification reactions to grow under low-oxygen condition by respiration of nitrate]$_{IA}$.

Supervised learning

To classify the trigger arguments into SS or DS, we have experimented with different types of algorithms implemented in Weka, ranging from simple probabilistic classifiers (Naïve Bayes) to decision trees (J48 and RF), rules (JRip) and Support Vector Machines (SMO). We have also employed the Vote meta-classifier, which is configured to consider the five previous classifiers, using an Average of Probabilities combination rule.

Table 6.11 shows the macro-averaged performance of the employed classifiers. As can be seen, the performances are very similar between all classifiers, their F-score ranging within just under 2% of 93%. Furthermore, the Vote meta-classifier improves the results only slightly, by 1.73% over JRip, which leads us to the conclusion that all

Features	P	R	F_1
L	93.45%	93.35%	93.40%
X	89.05%	89.50%	89.27%
P	92.85%	92.65%	92.75%
LX	94.05%	93.95%	94.00%
LP	94.40%	94.15%	94.27%
LXP	94.95%	94.65%	94.80%

Table 6.12: Performance of the Vote meta-classifier in classifying triggers as SS or DS.

classifiers make relatively the same decisions. Both precision and recall are balanced in the classification.

Table 6.12 shows the performance of the Vote meta-classifier when varying the feature set. The best performance is obtained when all feature types are employed, reaching an F-score value of 94.80%. This is closely followed by both LP and LX, which also reach values of over 94%. In fact, most combinations give F-score of over 92%. The worst performing feature set is when syntactic features are used by themselves, resulting in just over 89% F-score. This is because the variety of patterns leads to a sparse feature space, which results in the lower performance. For example, there are 228 different PoS patterns for the triggers, and 128 of these occur only once. Thus, there will be a significant amount of unseen data in the test fold in each of the ten folds.

The precision and recall are balanced, with precision being slightly (under 0.25%) higher than recall in all cases with the exception of syntactic features. Again, this is due to data sparseness, as it is difficult for the classifier to generalise when rare patterns occur.

JRip is the second best performing classifier for this task, obtaining an F-score of 1.73% less than Vote. As can be noticed from Table 6.13, the combination of all features again provides the best performance of 93.07%, but it is very closely followed by the lexical feature set, at 93.00%.

Features	P	R	F_1
L	93.10%	92.90%	93.00%
X	87.75%	88.15%	87.94%
P	92.85%	92.65%	92.75%
LX	92.85%	92.65%	92.75%
LP	93.05%	92.95%	92.99%
LXP	93.20%	92.95%	93.07%

Table 6.13: Performance of the JRip classifier in classifying triggers as SS or DS.

```
(P1 <= 4) and (X1_DT == 1) => DS (221/0)
(P2 <= 0.1) and (P4 <= 31) => DS (71/3)
(P2 <= 0.114286) and (P4 <= 42) => DS (35/8)
(X1_VBP == 1) and (L1_indicate == 1) => DS (8/2)
(P2 <= 0.25) and (X1_RB == 1) and (P4 >= 48) => DS (5/0)
(L1_is == 1) and (P4 <= 17) => DS (4/0)
(P2 <= 0.483871) and (P2 >= 0.481481) => DS (7/1)
=> SS (447/17)
```

Figure 6.3: Rules induced by the JRip classifier for argument location identification.

We investigated the output of this classifier to better understand how the features are used in the classification. Figure 6.3 shows the rules induced by JRip in the case of LXP feature set. The numbers in the parantheses at the end of each line stand for coverage / errors in the training data, which follows the standard convention of tree/rule induction. For instance, `(P2 <= 0.114286) and (P4 <= 42) => DS (35/8)` means that the rule `(P2 <= 0.114286) and (P4 <= 42) => DS` covers instances with total weights of 35, out of which there are instances with weights of 8 misclassified. In our case, each instance has a weight of 1, thus the rule applies to 35 instances, out of which 8 are misclassified. This shows the discriminant power of the employed feature set, which relies mostly on positional information. Special attention needs to be given to the first rule, which, by using only positional and PoS information (does the trigger contain a *DT*?), manages to correctly classify 221 instances with no misclassifications.

Classifier	P	R	F_1
Naïve Bayes	93.56%	96.42%	94.97%
SVM	93.50%	94.44%	93.97%
JRip	91.99%	91.57%	91.78%
J48	93.94%	93.00%	93.47%
RandFor	92.65%	90.04%	91.32%
Vote	93.97%	93.97%	93.97%

Table 6.14: Performance of various semi-supervised algorithms in classifying triggers as SS or DS.

Semi-supervised learning

In a semi-supervised framework, we have used the self-learning approach detailed in Section 5.5.3. This provides us with a means to overcome the data sparseness especially as regards the syntactic features.

We make use of the same process and unlabelled data for the learning process. Thus, we split BioCause into two equally sized sets, one used as seed data and one for final model evaluation. The unlabelled set is used for the self-learning step. For the purpose of feature extraction, the causal triggers in the unlabelled data set are automatically annotated using the best performing model created in Chapter 5, which is semi-supervised CRFs. Thus, the errors arising from automatic causal trigger recognition are propagated in the present step.

In case the system gets into the blocked state, we use the Position rule that was previously described. The rule is applied on the top 5% confident classifications.

Table 6.14 shows the best performance achieved by each of the classifiers used in the supervised setting. As can be observed, some F-scores achieved are slightly lower than those obtained in the supervised classification. This happens for the JRip, Random Forest and Vote classifiers and is due to two main reasons. First, the noisy data occurring in the unlabelled set confuses classifiers in their decisions. For instance,

one erroneously identified causal trigger is the word *DNA* in sentence (6.2) below.

(6.2) The Cre-mediated inverted band (6.5 kb) is evident in thymus *DNA* (thymoma).

Another reason is the low recall in recognising triggers. Whilst the precision is high, only a limited set of causal triggers is identified, due to data sparseness.

However, the Naïve Bayes, SVM, and J48 classifiers manage to improve both their precision and recall, which leads to an increased F-score for each of them. In fact, the recall of Naïve Bayes increases considerably, by almost 5%, whilst the precision is almost 2% higher. In the case of SVM, the increase is more moderate, of just 1% in the case of precision and 2% in the case of recall. The improvement of J48 is slightly less than that, with just under 1% for precision and 0.2% for recall.

We have experimented with various values for the τ parameter and the size of the seed data. As before, the τ parameter takes values from 0.6 to 0.9, in increments of 0.1, whilst the size of the seed data can vary between 12.5% and 100% in steps of 12.5%. The ratio between positive and negative instances in the seed data has not been included as a parameter, as the data set is roughly balanced. Since the seed data is selected randomly from the labelled set, we repeat each experiment ten times. The average of the obtained results is given for each of the six classifiers in Figures 6.4 - 6.9.

As can be noticed from Figure 6.4, the performance of the Naïve Bayes classifier remains relatively insensitive to the variance of τ and seed size. The amplitude of its F-score is just 1.50%, which is not seen in any of the other classifiers. This is partly due to the fact that this specific classifier offers probabilities for each of the two classes that are several orders of magnitude apart. When normalising them, this results in having a binary output, with 0 and 1 as the final probabilities.

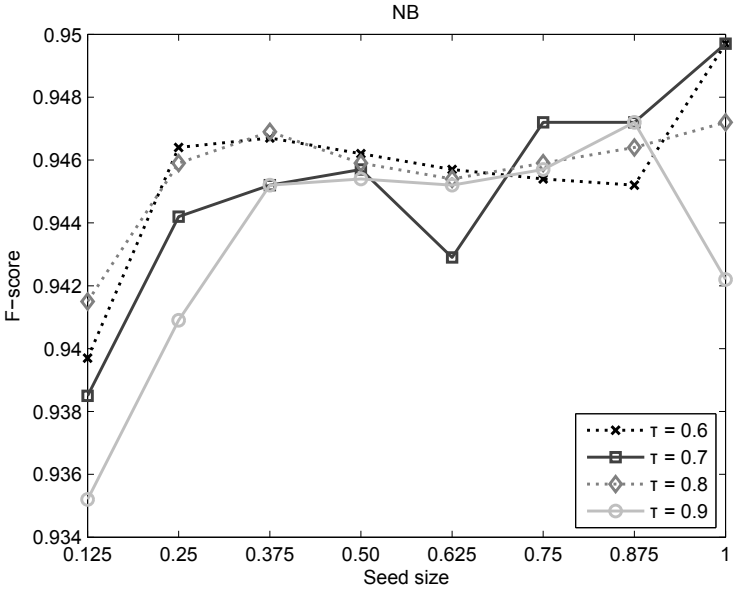


Figure 6.4: Self-training results for the argument location NB classifier when varying τ and the seed size.

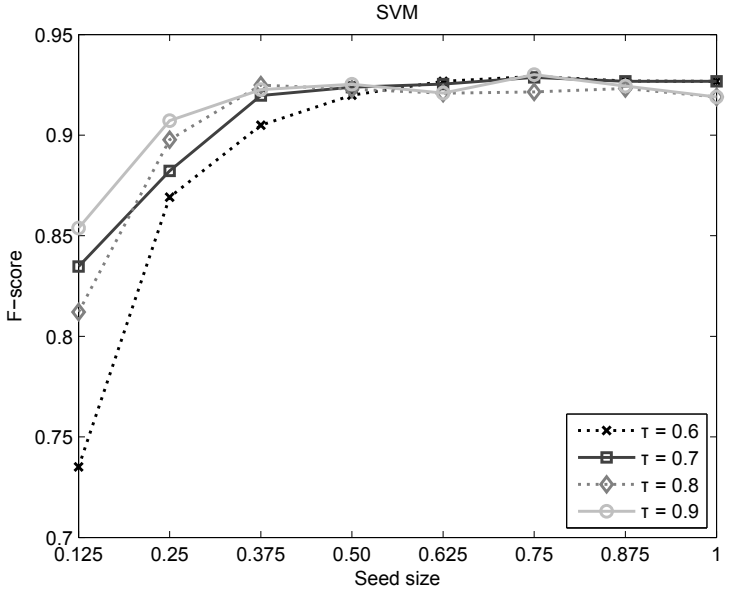


Figure 6.5: Self-training results for the argument location SVM classifier when varying τ and the seed size.

The SVM, RF and Vote classifiers suffer significantly when the size of the seed data is 12.50%. All three start at very low values, 61% in the case of RF and 72% in the case of SVM and Vote. The performance quickly increases to over 80% once more data joins the labelled set.

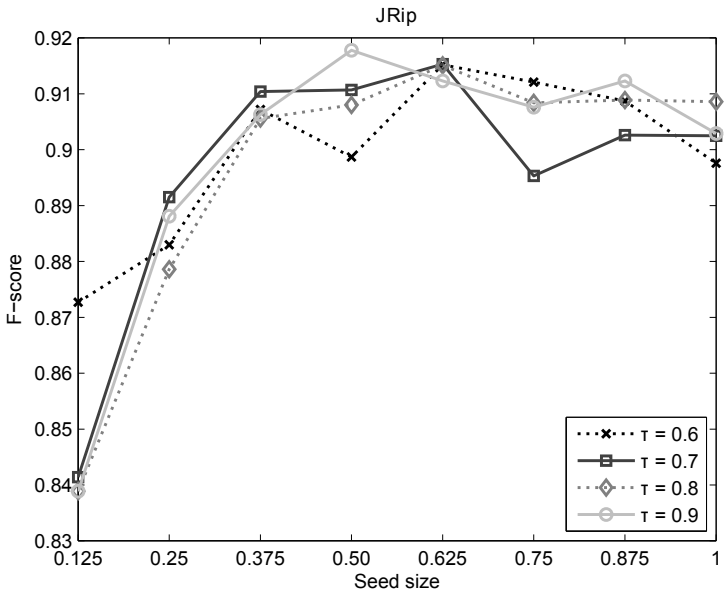


Figure 6.6: Self-training results for the argument location JRip classifier when varying τ and the seed size.

A similar trend is observed on JRip and J48, but to a much lesser degree. In fact, J48 behaves strangely at the other end of the seed size as well. The graph shows a decrease in F-score when 100% of the seed data is available for initial training, which is due to a decrease in precision, whilst the recall remains constant. This happens because of the high variability of low frequency triggers occurring many times non-causally, which allows for the production of many false positives.

The value of the τ parameter again does not seem to influence the performance of the classification, especially when more labelled data is available. The only classifier with a visibly separate line for the 60% confidence value for τ is Vote. In this case, the performance of the model at 60% confidence threshold is 1-2% lower than the other

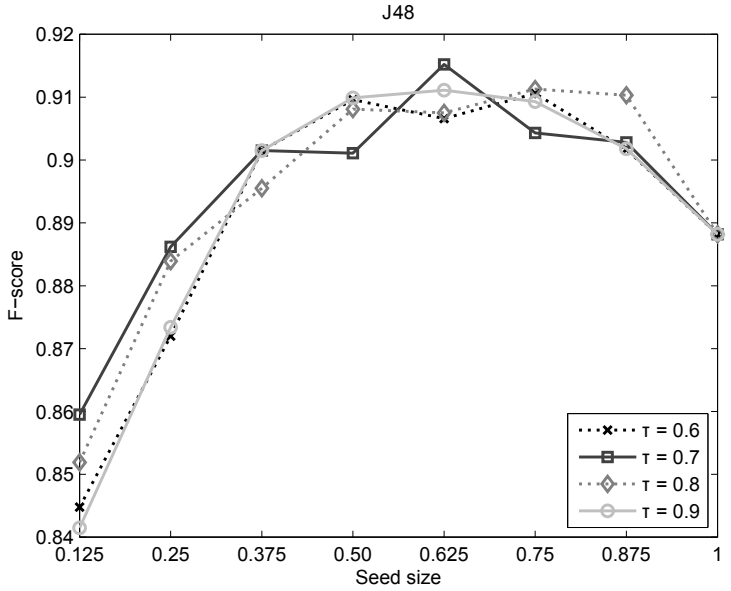


Figure 6.7: Self-training results for the argument location J48 classifier when varying τ and the seed size.

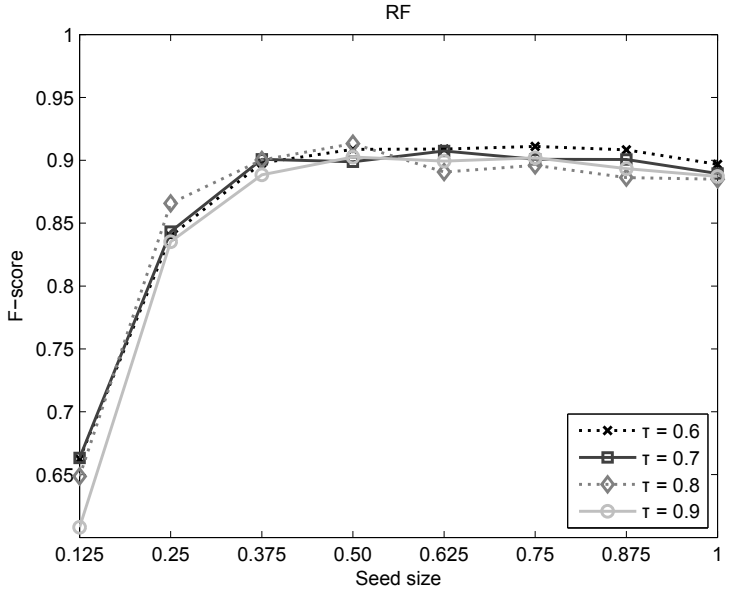


Figure 6.8: Self-training results for the argument location RF classifier when varying τ and the seed size.

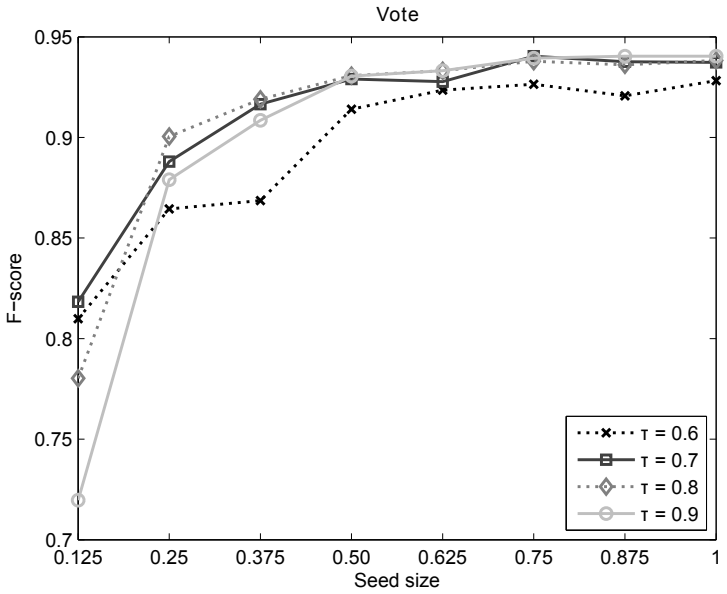


Figure 6.9: Self-training results for the argument location Vote meta-classifier when varying τ and the seed size.

confidence levels throughout all seed sizes.

6.5.2 Argument span identification

In the previous section we focussed on determining the position of the two arguments relative to the trigger. Whilst the syntactically dependent argument is always adjacent to the trigger, the independent argument can be located either in the same sentence with the causal trigger, or in a different one. This latter case is the most difficult to solve, as the search space becomes very large and syntactic dependency and constituency do not provide any help. Instead, we rely on semantics and positional features to provide discriminating information.

Rule-based

Our rule-based approach to determining the spans of the two arguments relies on the parse tree of sentences in the case of same-sentence arguments. For different-sentence

Argument-Case	P	R	F_1
DA-SS	74.36%	100%	85.29%
IA-SS	82.75%	96.53%	89.11%
DA-DS	83.79%	100%	91.18%
IA-DS	54.98%	60.58%	57.64%

Table 6.15: Performance of rules in identifying dependent (DA) and independent (IA) argument spans.

arguments, the decision is simpler and based on statistics. All engineered rules are described below.

In the case of same-sentence arguments, we employ a naïve rule which splits the sentence into two segments, each on either side of the trigger. The shortest segment that is contained within an S or S-REL constituent and immediately follows the trigger is marked as the dependent argument, whilst the segment preceding the trigger is marked as the independent argument. The performance of this simple rule is impressive, reaching values between 85% and 90%, as can be noticed from Table 6.15. More specifically, the evaluation result for dependent argument identification in the case of SS is F-score 85.29%. The recall reaches 100%, as all words after the token are marked as part of this argument. However, the precision only gets to almost 75%, showing that a better selection needs to be implemented to identify tokens that are not part of the argument. In contrast, the independent argument identification in the case of SS reaches a higher F-score of 89.11%. The recall is less than 100% because there is a handful of cases where the independent argument is located after both the trigger and the dependent argument, as mentioned in Table 4.11.

In case the two arguments of the causal trigger are classified as being in distinct sentences, we mark the entire sentence containing the trigger as the dependent argument. Thus, the dependent argument is marked as starting from the end of trigger to the end of the sentence. This leads to an F-score of 91.18%, with 100% recall and almost 84% precision.

Classifier	P	R	F_1
CRF	85.98%	79.98%	82.87%
SVM	80.48%	74.49%	77.37%
Random Forest	79.42%	75.14%	77.22%
Naïve Bayes	63.42%	66.11%	64.73%

Table 6.16: Overall performance of various classifiers in identifying dependent (DA) and independent (IA) argument spans.

The independent argument is marked as the preceding sentence to that containing the trigger. This results in an F-score of 57.64%, much lower than those of other arguments. The two main reasons for a lower score are the syntactic independency and the large search space. Practically any sentence preceding or following the trigger sentence can play the role of the independent argument. The only possibility to improve the accuracy of identifying it is by employing deep semantics and other discourse features, which are used by machine learning approaches.

Supervised learning

We have experimented with four different classifiers, namely Support Vector Machines, Random Forests, Naïve Bayes and Conditional Random Fields, in a supervised setting. Table 6.16 shows the overall performance of the four classifiers that we have employed. The following four tables, 6.17 - 6.20, contain the results specific to each of the four cases of arguments.

As can be noticed, the overall best performing classifier is CRF. It overperforms both SVM and RF by almost 10% in terms of F-score. Naïve Bayes is even further away, at over 15% distance.

Although the performance of SVM and RF is similar, the better precision is obtained by SVM, whilst the better recall by RF. In fact, the precision of SVM is very close to that of the CRF-based model, with only 3% difference. Naïve Bayes, although producing a very high recall, manages to recognise correctly only just over 55% of its

Classifier	P	R	F_1
CRF	91.79%	88.22%	89.97%
SVM	88.82%	81.04%	84.75%
Random Forest	85.78%	82.20%	83.95%
Naïve Bayes	67.12%	73.40%	70.12%

Table 6.17: Performance of various classifiers in identifying DA-SS argument spans.

Classifier	P	R	F_1
CRF	93.11%	83.48%	88.03%
SVM	87.95%	81.25%	84.47%
Random Forest	87.75%	81.75%	84.18%
Naïve Bayes	69.95%	68.66%	69.30%

Table 6.18: Performance of various classifiers in identifying IA-SS argument spans.

output. We discuss in what follows each of the four cases, and then the best features of each individual algorithm.

The same-sentence dependent argument (DA-SS) obtains the best results amongst all four cases. These are listed in Table 6.17. This is due to the syntactic dependency that exists between itself and the trigger. As such, CRF reaches an impressive F-score value of 90%, whilst SVM and RF immediately follow at almost 85% F-score. Naïve Bayes again performs the worst, with an F-score of 70%.

The second best case is that of the same-sentence independent argument (IA-SS), whose results are given in Table 6.18. In fact, the performance of CRF is not much lower than that of the DA-SS case. The difference between these two cases is just under 2% in terms of F-score. However, the precision of the IA-SS is higher than that of DA-SS, whilst the recall is lower. This is due to the fact that the classifiers tend to mark extra tokens as part of the DA-SS argument, which results in many false positives. SVM, RF, and NB maintain their relative distance to CRF as previously.

The third case is that of the different-sentence dependent argument, presented in Table 6.19. The results are similar to the previous case of dependent argument, but

Classifier	P	R	F_1
CRF	86.45%	82.25%	84.30%
SVM	80.55%	76.30%	78.37%
Random Forest	80.05%	76.45%	78.21%
Naïve Bayes	63.45%	68.25%	65.76%

Table 6.19: Performance of various classifiers in identifying DA-DS argument spans.

Classifier	P	R	F_1
CRF	72.58%	65.95%	69.11%
SVM	64.58%	59.36%	61.86%
Random Forest	64.08%	60.15%	62.05%
Naïve Bayes	53.15%	54.12%	53.63%

Table 6.20: Performance of various classifiers in identifying IA-DS argument spans.

slightly lower for all algorithms. In the case of CRF, the F-score drops by approximately 5%, whilst in the case of SVM and RF the decrease is of almost 7%.

Finally, the case of independent arguments in different sentences obtains the worst results. This is due to the complete syntactic independence between this argument and the trigger. The only real support for the identification of this argument comes from the lexical and semantic features. CRF again obtains the best precision and recall, reaching to 69% F-score. SVM and RF perform slightly worse, reaching almost 62% F-score. Even lower results are obtained by Naïve Bayes, which gets to almost 54% F-score.

Tables 6.21 - 6.24 show the best results for each of the four classifiers with various combinations of feature types. As can be noticed, the CRF and SVM classifiers obtain their best scores when all types of features are used, whilst RF and Naïve Bayes perform best when excluding syntactic features.

In the case of Random Forest, shown in Table 6.23, the addition of syntactic features to the model increases the recall slightly, by 0.13%, but decreases the precision by 0.23%, thus resulting in an F-score lower by 0.04%. For Naïve Bayes (Table 6.24), the addition of syntactic features boosts the recall by 0.44%, but seriously affects the

Features	P	R	F_1
L	83.65%	77.14%	80.26%
LX	83.94%	77.78%	80.74%
LP	86.22%	78.02%	81.92%
LXS	84.33%	78.67%	81.40%
LXDCP	86.05%	78.85%	82.29%
LXDCPS	85.98%	79.98%	82.87%

Table 6.21: Performance of the CRF classifier in identifying dependent (DA) and independent (IA) argument spans.

Features	P	R	F_1
L	78.28%	72.75%	75.41%
LX	78.65%	73.04%	75.74%
LP	81.05%	73.35%	77.01%
LXS	79.80%	74.00%	76.79%
LXDCP	80.98%	74.04%	77.35%
LXDCPS	80.48%	74.49%	77.37%

Table 6.22: Performance of the SVM classifier in identifying dependent (DA) and independent (IA) argument spans.

Features	P	R	F_1
L	78.23%	72.33%	75.16%
LX	79.02%	72.85%	75.81%
LP	80.12%	74.42%	77.16%
LXS	79.25%	74.90%	77.01%
LDCPS	79.65%	75.01%	77.26%
LXDCPS	79.42%	75.14%	77.22%

Table 6.23: Performance of the Random Forest classifier in identifying dependent (DA) and independent (IA) argument spans.

precision, which drops by 1.1%, thus resulting in an F-score lower by 0.36%.

Furthermore, it can be observed that semantic features improve the F-score in all combinations they are added. They work by increasing the recall in most cases, although a drop in precision occurs as a consequence of that. Nevertheless, there are cases where both precision and recall increase by adding semantic information to the feature set. This is to be expected, as semantics generalises knowledge very well.

Features	P	R	F_1
L	62.05%	64.25%	63.13%
LX	62.58%	65.04%	63.79%
LP	63.98%	65.97%	64.96%
LXS	63.02%	65.88%	64.42%
LDCPS	64.52%	65.67%	65.09%
LXDCPS	63.42%	66.11%	64.73%

Table 6.24: Performance of the Naïve Bayes classifier in identifying dependent (DA) and independent (IA) argument spans.

Classifier	P	R	F_1
CRF	84.52%	79.58%	81.98%
SVM	75.85%	77.95%	76.89%
Random Forest	76.95%	76.50%	76.72%
Naïve Bayes	63.30%	67.35%	65.26%

Table 6.25: Performance of semi-supervised various classifiers in identifying dependent (DA) and independent (IA) argument spans.

Semi-supervised learning

Similar to the previous experiments, we use half of the BioCause corpus as seed data, 24 full text articles as learning data, and the other half of BioCause for final model evaluation. The two halves are then swapped and the experiments repeated. The automatic annotations of triggers over the learning data are enhanced with new information regarding the location of the two arguments, obtained from the best performing classifier detailed in the previous section.

Table 6.25 shows the results that were obtained with the same classifiers in a semi-supervised setting. As can be noticed, CRF leads the performance results, with almost 82% of the arguments identified correctly. SVM and RF are situated at around 5% lower than CRF, whilst NB manages to obtain just 65% F-score.

The results of the first three classifiers are slightly lower than in the case of the supervised method. Whilst the F-score of CRF drops by 1%, the scores of SVM and

RF decrease by only 0.5%. In contrast, the NB classifier manages to improve its performance by almost 0.5%, due to an 1.2% increase in recall.

The slight decrease is due to the errors arising from the automatic annotation of the unlabelled data by using the models from previous steps. There are several cases in which a same-sentence trigger is erroneously classified as different-sentence, such as the one in example (6.3). This type of errors is due to the order of the causal constituents, T-E-C in this case (order occurring in only 1.29% of all relations in BioCause). Since the trigger is the first token in the sentence, the algorithm decides that the arguments are located in distinct sentences.

(6.3) *Since* [Brucella is an intracellular facultative pathogen]$_{DA}$, [the bacteria could use these denitrification reactions to grow under low-oxygen condition by respiration of nitrate]$_{IA}$.

The reverse occurs as well: there are several cases where different-sentence triggers are classified as being same-sentence, as shown in example (6.4). This happens when the trigger is located mid-sentence and the majority of its occurrences are in fact same-sentence.

(6.4) [The fact that PmrB is likely to sense changes in pH directly]$_{D}$*A is supported by* multiple findings.
First, [the mild acid pH-dependent activation of the PmrA-regulated gene pbgP was dramatically reduced in a strain lacking pmrB]$_{I}$*A*.

Figures 6.10 - 6.13 depict the change in the obtained F-score when varying the seed size and confidence threshold for each of the four classifiers.

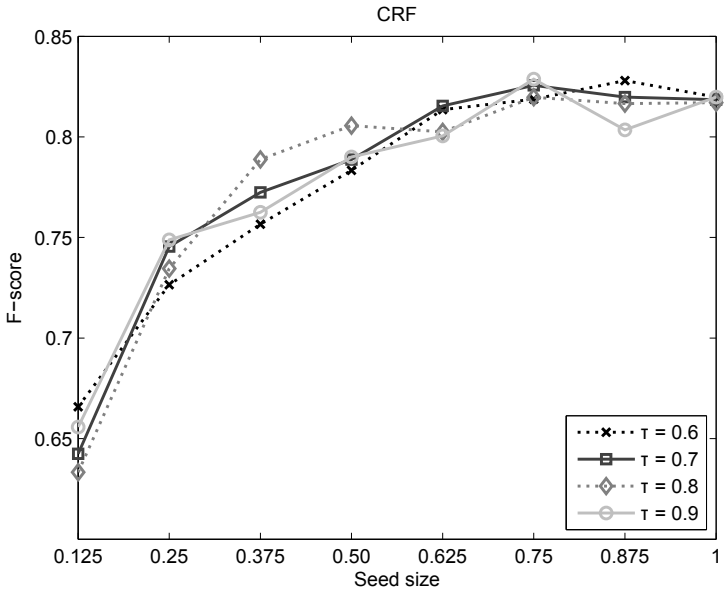


Figure 6.10: Self-training results for the argument span CRF classifier when varying τ and the seed size.

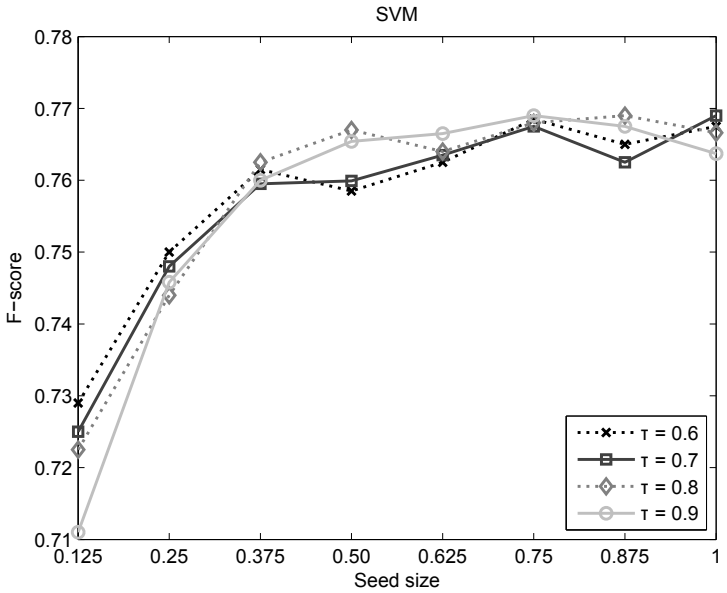


Figure 6.11: Self-training results for the argument span SVM classifier when varying τ and the seed size.

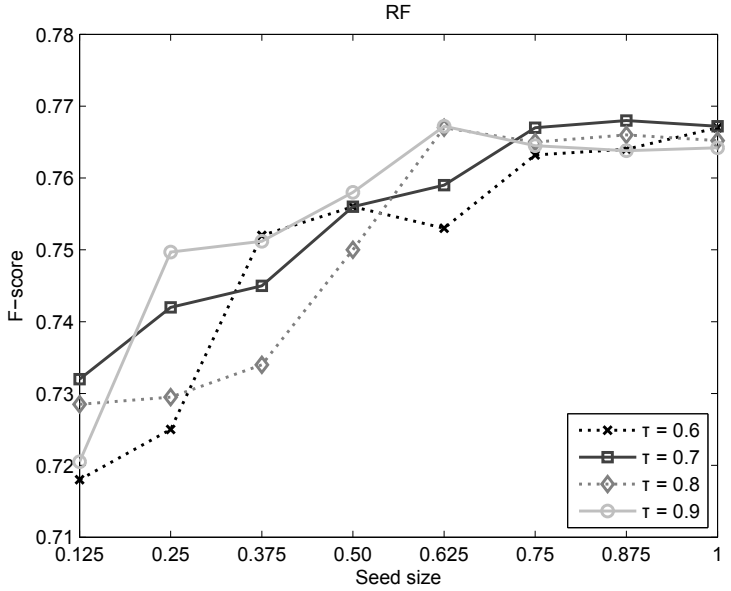


Figure 6.12: Self-training results for the argument span RF classifier when varying τ and the seed size.

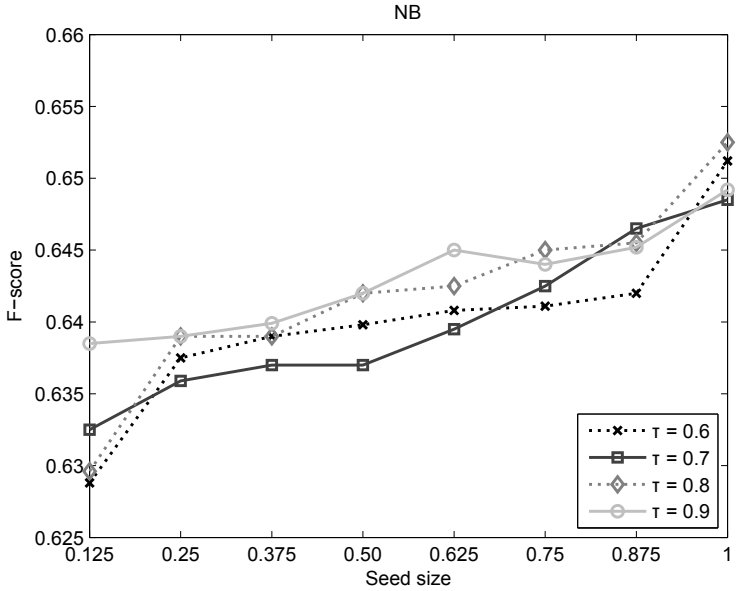


Figure 6.13: Self-training results for the argument span NB classifier when varying τ and the seed size.

As noticed before, Figure 6.13 shows that the Naïve Bayes classifier has a very small amplitude in the F-score curve, of just over 2%. In contrast, the other three algorithms increase their performance by approximately 5% when changing the size of the seed data from 12.5% to 100%. All classifiers are, however, insensitive to the modification of the confidence threshold, especially when higher amounts of seed data are available.

6.5.3 Relation direction identification

The final step in the causality recognition pipeline is to detect which argument plays which semantic role. Each of the previously identified arguments must be assigned one of the two possible roles, Cause and Effect. For this task, we have explored different possibilities to detect whether a causal relation is of the form C-T-E or E-T-C. The other three possibilities existing in BioCause have been excluded from the classification, as their number is insufficient for training purposes.

One aspect that has to be taken into consideration is the skewed data. The E-T-C to C-T-E ratio is 1:7.54.

Rule-based

Table 6.26 lists the results of all the rules that we have created for this third step of assigning roles to the arguments.

The most obvious and simplest rule that can be developed is the majority class rule, especially under the circumstances of such a high skew. All instances are marked with the majority class label (C-T-E), which, for that class, will produce a complete recall, and a slightly lower precision. However, the zero precision and recall for the minority class will halve those numbers for the macro F-score. In this case, the macro-average precision is 44.15%, whilst the recall is 50%. Thus, the macro-average F-score reaches

Rule	P	R	F_1
Majority	44.15%	50%	46.89%

Table 6.26: Performance of rules in identifying causal direction.

Classifier	P	R	F_1
Naïve Bayes	69.85%	83.80%	73.40%
SVM	81.70%	79.90%	80.80%
JRip	81.60%	80.35%	80.95%
J48	83.40%	79.15%	81.10%
RandFor	83.70%	72.55%	76.60%
Vote	85.25%	83.55%	84.35%

Table 6.27: Performance of various classifiers in identifying causal direction.

only 46.89%.

Supervised learning

Similar to the previous steps, we have experimented with multiple algorithms, ranging from simple probabilistic classifiers (e.g., Naïve Bayes) to trees (e.g., J48 and Random Forests), rules (JRip) and support vector machines (SVM). We have also used the Vote meta-classifier, which considers the five previous classifiers, and decides using an Average of Probabilities combination rule. All of the mentioned algorithms are used as implemented in Weka.

The macro-averaged results are provided in Table 6.27. Under the circumstances of the skewed data set, the best classifier, Vote, reaches an F-score of 96.40% in the case of C-T-E and of 72.30% in the case of E-T-C, resulting in a macro-average F-score of 84.35%.

This improves significantly over J48, the decision tree-based classifier, which is second best in terms of F-score, at more than 3% distance. In fact, the precision of Vote is increased with almost 2% over that of J48 and Random Forest, whilst the recall is similar to that of Naïve Bayes and much higher than that of the other classifiers. This

Features	P	R	F_1
L	85.85%	79.05%	82.30%
X	79.50%	76.75%	78.10%
LP	83.10%	80.75%	81.90%
LXS	84.45%	82.75%	83.59%
LPS	85.65%	81.55%	83.55%
LXPS	85.25%	83.55%	84.35%

Table 6.28: Performance of the Vote meta-classifier in identifying causal direction.

Features	P	R	F_1
L	81.75%	74.80%	78.12%
X	74.15%	72.05%	73.08%
LP	81.60%	75.85%	78.62%
LXS	81.70%	77.35%	79.46%
LPS	82.75%	76.65%	79.58%
LXPS	81.60%	80.35%	80.95%

Table 6.29: Performance of the JRip classifier in identifying causal direction.

shows that Vote exploits the individual strengths of each of the five classifiers. Repeating the experiment with a Majority Voting combination rule instead of the Average of Probabilities results in a similar output.

The most useful features in this classification, according to InfoGain and ChiSquare attribute evaluators, have proven to be the actual trigger, its lemmatised form, part-of-speech, syntactic category, its neighbours, the presence of the words *by*, *due* and pronouns, and the voice of the verb.

Table 6.28 shows the performance of combinations of features for the Vote meta-classifier. As can be noticed, combining all feature types leads to the best overall F-score, and also the best recall. However, the best precision is obtained by using only lexical features (85.85%).

Table 6.29 includes various combinations of feature types and their performance against the data. It is again noticeable that all feature types lead to the best recall and F-score. The best precision, however, excludes syntactic features. This is because

```
(X2_P >= 1) and (L1_by >= 1) => ETC (23/4)
(X2_V <= 0) and (X2_P >= 1) => ETC (33/10)
(L2_be >= 1) and (X7 = NN) and (L1_it <= 0) => ETC (14/4)
(X2 = SC) => ETC (11/1)
(L1_due >= 1) => ETC (8/2)
(X2 = ADV) and (P1 >= 20) => ETC (3/0)
(L2_report >= 1) => ETC (4/1)
 => CTE (682/17)
```

Figure 6.14: Rules induced by the JRip classifier for relation direction identification.

Classifier	**P**	**R**	**F_1**
Naïve Bayes	70.45%	80.05%	74.94%
SVM	82.50%	80.05%	81.25%
JRip	84.65%	80.90%	82.73%
J48	83.10%	79.20%	81.10%
RandFor	79.85%	74.20%	76.92%
Vote	84.55%	83.05%	83.79%

Table 6.30: Performance of various semi-supervised classifiers in identifying causal direction.

the syntactic patterns that we engineered help generalise and increase recall, with the downside of lowering the obtained precision.

Figure 6.14 shows the rules that are produced by the JRip classifier in determining the direction of the relation. As can be noticed, the most prominent feature is X2, the syntactic category of the trigger. More specifically, many rules contain a test whether the syntactic category of the trigger contains certain categories, such as *P*, *V*, *SC* or *ADV*. Other rules refer to lexical features, be it as surface expression or lemmatised forms, and positional information.

Semi-supervised learning

For the semi-supervised learning approach, the data is split similar to the previously described semi-supervised experiments. In addition, the argument spans are automatically detected using the best performing classifier described in the previous step.

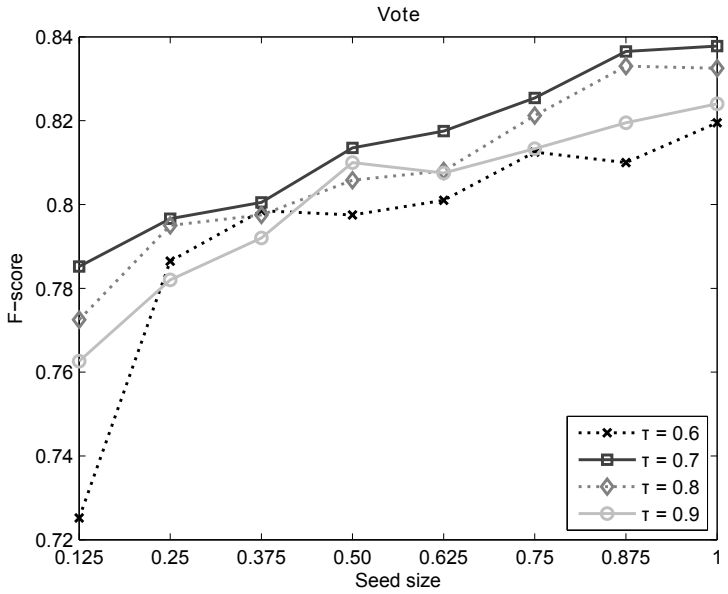


Figure 6.15: Self-training results for the argument role Vote meta-classifier when varying τ and the seed size.

Table 6.30 lists the results obtained by the six classifiers used as learning algorithms. The Vote meta-classifier has obtained the best performance, an F-score of 83.79%. However, it is still slightly lower than that obtained in a supervised setting. This is due to the propagation of errors from the previous two steps.

Besides the errors regarding the classification of the trigger into SS or DS, exemplified in the previous section, the current step inherited inaccurate spans for the arguments. Most common is the case of selecting the wrong span for the arguments located in a different sentence by choosing a completely wrong sentence. Another possibility is only the partial match for an argument, where the classifier also selects false positives and leaves out false negatives.

Figures 6.15 - 6.20 show the variation in F-score when changing the seed size and confidence threshold. As can be noticed, most classifiers have a generally increasing trend, with a high slope for small amounts of seed data. As this size increases, the slope of the F-score curve decreases and almost plateaus towards 100% of the seed

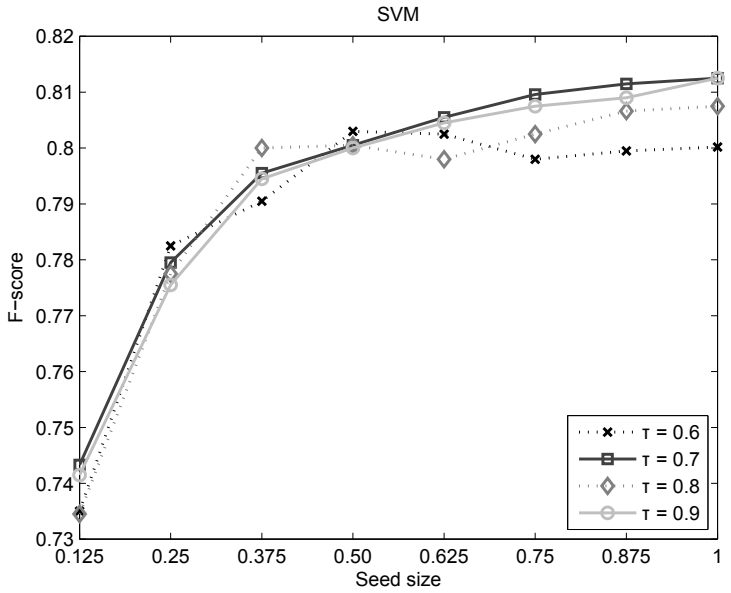


Figure 6.16: Self-training results for the argument role SVM classifier when varying τ and the seed size.

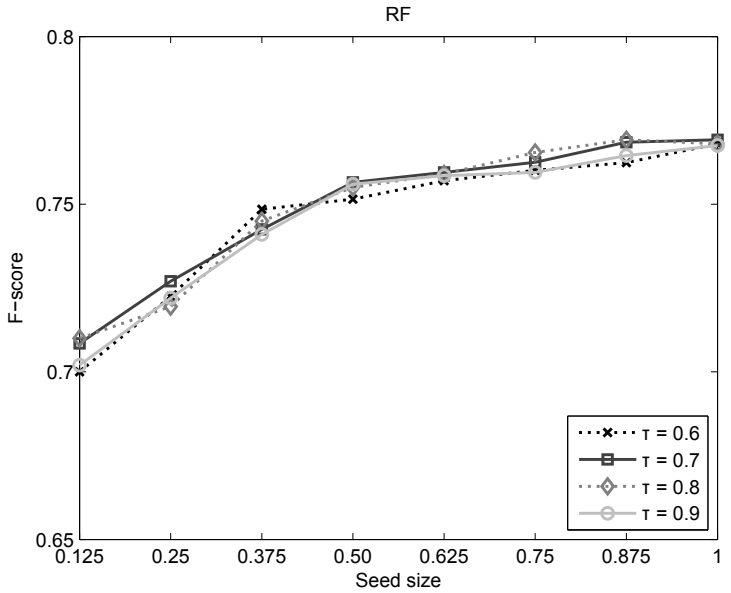


Figure 6.17: Self-training results for the argument role RF classifier when varying τ and the seed size.

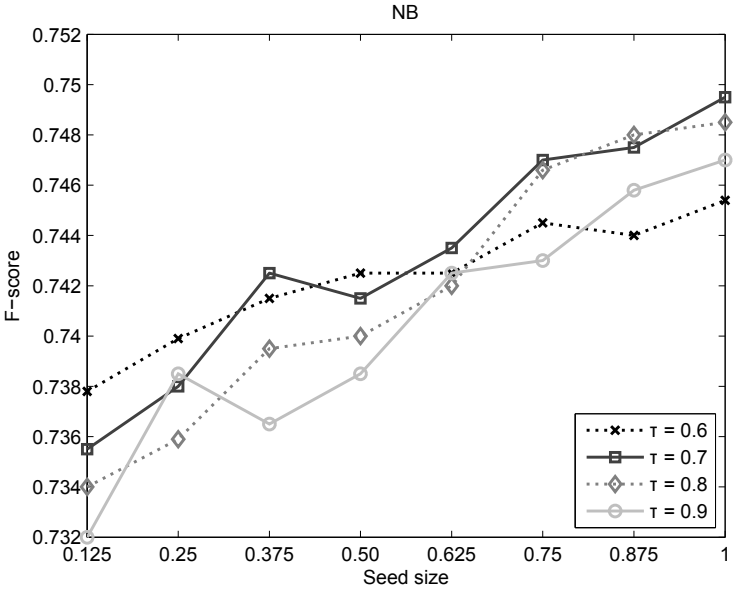


Figure 6.18: Self-training results for the argument role NB classifier when varying τ and the seed size.

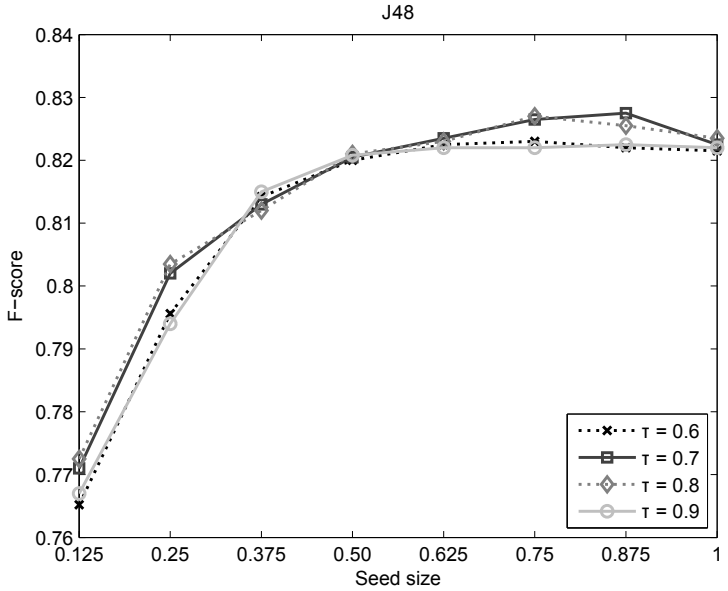


Figure 6.19: Self-training results for the argument role J48 classifier when varying τ and the seed size.

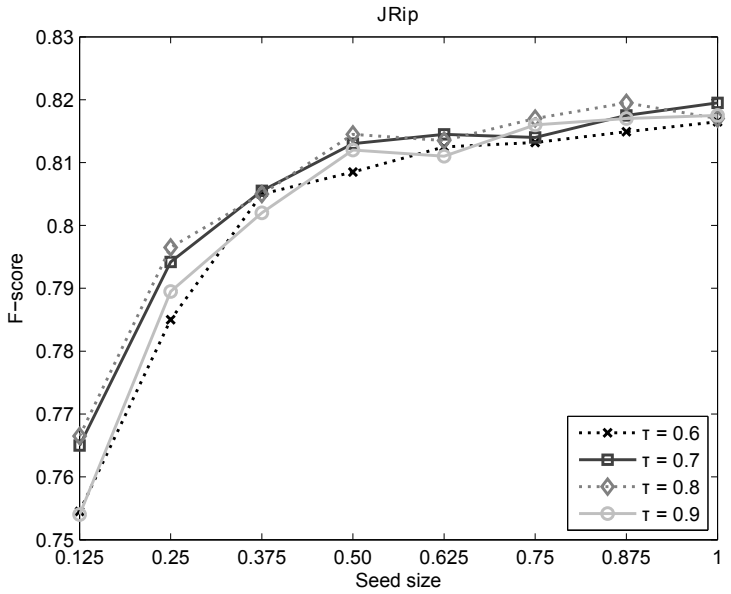


Figure 6.20: Self-training results for the argument role JRip classifier when varying τ and the seed size.

data. Naïve Bayes is, in contrast to all other classifiers, fairly constant throughout different seed sizes. However, its performance is the worst, at almost 10% distance from Vote.

The confidence threshold τ does not generally influence the performance of the algorithms. Notable cases are the value of 60% confidence, which obtains a low F-score for the Vote classifier at seed size 12.5% and for SVM at high seed sizes.

6.6 Effect of features

We have also investigated the usefulness of the numerous features that we have engineered. Whilst in the previous section we showed the effect of some combinations of different feature types on the various classifiers, we will now analyse how helpful are the features for all three steps and for the overall best performing algorithm. The following subsections discuss the behaviour of each feature type and its interaction with the other features types. The tables show the percentage of feature combinations where

ID	Usefulness	Av. increase	Av. decrease
L1	73.58%	0.77%	0.39%
L2	73.16%	0.73%	0.35%
L3	65.66%	0.54%	0.22%
L4	75.36%	0.79%	0.41%

Table 6.31: Usefulness of lexical features in identifying causal arguments.

by adding that specific feature the performance is improved in terms of F-score. Not included in the table are the individual values for precision and recall; these will be commented in the text. Furthermore, both the average increase and average decrease in performance by adding a feature are statistically significant for an a = 0.05 with a p-value of less than 0.001, unless otherwise stated in the text, using Student's t-test.

6.6.1 Lexical features

Lexical features are of great use in the task of recognising causal arguments. Table 6.31 shows the usefulness of each lexical feature, together with its average increase and decrease of F-score in feature combinations.

Feature L3, the capitalisation flag, is the least useful feature in this set. This is because there are many instances of tokens which are part of triggers, but they are not necessarily capitalised. The only such case occurs when a sentence starts with a trigger, and then only the first token of the trigger will be capitalised.

The number of cases where an increase in precision is observed when using these features is very high, exceeding 90%. However, an increase in recall occurs in around 50% of the cases. Also, the average increase in precision is 1.25%, whilst the decrease is only 0.19%. Recall is more balanced, with 0.51% average increase and 0.48% average decrease.

ID	Usefulness	Av. increase	Av. decrease
X1	59.68%	0.41%	0.37%
X2	58.57%	0.37%	0.33%
X3	75.54%	1.35%	0.38%
X4	71.87%	1.28%	0.35%
X5	78.64%	1.72%	0.36%
X6	78.35%	1.70%	0.39%
X7	68.92%	0.30%	0.19%
X8	4.28%	0.29%	1.91%
X9	73.68%	1.18%	0.58%
X10	56.33%	0.25%	0.11%
X11	55.67%	0.21%	0.09%

Table 6.32: Usefulness of syntactic features in identifying causal arguments.

6.6.2 Syntactic features

Syntactic features are listed in Table 6.32 together with the percentage of feature combinations they improve. As can be noticed, about half of the created features improve the classification significantly, whilst the other half just slightly. The only exception is feature X8, the main verb of the sentence, which does not generalise very well over the data.

Both the part of speech and syntactic category features are moderately useful in recognising argument spans. Their addition improves the F-score in around 59% of feature combinations, with low increase and decrease averages. The difference between the two features lies in the fact that X1 increase mostly recall, in more than 66% of cases, to the disadvantage of precision (less than 40% of cases), whereas X2 is roughly balanced between precision and recall, at around 57% of feature combinations.

Features X3 and X4, string representations of the part-of-speech and syntactic category of triggers, are used in the first and third steps. These two features provide a significant amount of discriminative information to the classifiers such that they improve the F-score in more than 71% of cases. Furthermore, the average increase is of around 1.30%, whilst the decrease is of only 0.35%.

Even more informative are features X5 and X6, which remove contiguous duplicate PoSs or syntactic categories from the string representations. This is due to the fact that the removal of duplicates reduces the number of possible values and therefore data sparsity. At 78% of cases with an F-score increased by 1.70% on average, these features are amongst the most helpful for the first and last steps.

Regarding the main verb of the sentence, checking whether the trigger contains it (X7) proves to be a good feature, increasing the F-score in more than two thirds of cases, although the average improvement is very small – 0.3%. The recall is the one that is boosted by this feature, in more than 67% of cases, whilst precision benefits in just 60% of cases.

In contrast, the actual main verb of the sentence (X8) mostly affects the performance. It improves only 4% of feature combinations, by 0.29% on average. The decrease, however, reaches an impressive value of 1.91%.

Feature X9, the voice of the verb in the trigger, if there is any, has proven its efficiency by increasing the F-score in almost 74% of feature combinations for the third step of our pipeline. It increases the F-score by more than 1% on average, and recall benefits the most from this feature, increasing in more than 80% of cases.

Finally, the last two features, X10 and X11, include the PoS and syntactic category of the neighbouring tokens for the span detection. They are slightly less useful than the first two features, X1 and X2. The F-score is increased in around 56% of feature combinations, and the impact of these features is very small. The increase and decrease averages are of only 0.20% and 0.10%, respectively.

6.6.3 Dependency features

Table 6.33 includes the usefulness of all dependency features employed in this study. Feature D1, the surface expression of the arguments which are dependent on the current

ID	Usefulness	Av. increase	Av. decrease
D1	69.24%	0.33%	0.38%
D2	62.69%	0.13%	0.15%
D3	69.95%	0.35%	0.37%
D4	61.45%	0.07%	0.15%

Table 6.33: Usefulness of dependency features in identifying causal arguments.

ID	Usefulness	Av. increase	Av. decrease
C1	70.52%	0.95%	0.57%
C2	79.96%	0.99%	0.50%
C3	72.29%	0.96%	0.45%
C4	59.85%	0.84%	0.45%
C5	61.25%	0.78%	0.41%
C6	62.03%	0.87%	0.53%
C7	72.48%	0.85%	0.42%
C8	76.67%	1.05%	0.56%
C9	75.87%	0.98%	0.52%

Table 6.34: Usefulness of command features in identifying causal arguments.

token, is rather helpful, increasing both precision and recall in around 69% of the cases.

Similar values are obtained for the PoS of these arguments. In contrast, the distance between the arguments and the token is not that helpful. It increases the performance in about 61% of the cases, but the average decrease is much higher than the increase.

6.6.4 Command features

The nine command features provide significant information to the classifier, according to the data in Table 6.34. As can be observed, the most useful features are c-command and VP-command, where the commanded constituent has the syntactic category VP or NP. These help in more than 70% of the feature combinations, and have high average increase and low average decrease values, similar to what has been observed in the trigger recognition.

The S-command features (C4-C6) also help in the classification task, but not as much as the rest. It improves only 59-62% of cases by about 0.82%. This can be

ID	Usefulness	Av. increase	Av. decrease
P1	74.85%	1.53%	0.52%
P2	78.25%	1.72%	0.54%
P3	76.95%	1.45%	0.38%
P4	69.54%	0.74%	0.29%

Table 6.35: Usefulness of positional features in identifying causal arguments.

explained by the fact that, although a high proportion of triggers S-commands SBARs, VPs or NPs, there are also many non-triggers which S-command the same syntactic categories. Thus, the S-command feature does not provide as much information to the classifier as the other command features.

6.6.5 Positional features

Table 6.35 lists the effect of position features. These features are useful especially in the first and third steps, where the classifications are made based on the trigger. The second step considers all tokens in the text, and the position is not as relevant for this argument span task.

The index of the token in the sentence, feature P1, is useful in almost 75% of cases. The precision benefits most from this feature, with an average increase of 1.42% in almost 95% of cases. The recall, however, is increased much less, 0.34%, and in only 45% of cases.

Even more useful than P1 is feature P2, the percentual position in the sentence. The F-score is increased in 78.25% of the feature combinations. Precision is improved in almost all cases by around 1.80%, whilst recall improves in about 50% of cases by 0.40%.

The third feature, which discretises the position in three values, sits in between the first two with respect to its usefulness. The length of the sentence, P4, is the least informative amongst positional features, increasing the F-score in only 69.54%

ID	Usefulness	Av. increase	Av. decrease
S1	89.90%	2.94%	0.17%
S2	89.50%	2.78%	0.23%
S3	75.82%	1.68%	0.54%
S4	68.24%	1.13%	0.45%
S5	69.30%	1.85%	0.73%
S6	60.71%	0.25%	0.19%
S7	29.04%	0.28%	0.43%
S8	84.89%	2.56%	0.08%
S9	82.59%	2.38%	0.11%
S10	75.42%	2.10%	0.25%
S11	77.23%	2.15%	0.24%

Table 6.36: Usefulness of semantic features in identifying causal arguments.

of cases. Even in these cases, the average increase is of only 0.74%, which is about half of the other features.

6.6.6 Semantic features

Table 6.36 summarises the effect of semantics across the various combinations with other features.

The features that concern the named entity information are the most informative for the entire task. They increase the F-score in almost 90% of the feature combinations, with the average increase of around 2.80%. Additionally, the decrease is very small, of around 0.20%. Recall is the one that benefits from these features, in more than 90% of cases, whilst precision suffers by decreasing in about 70% of cases.

Event information is less useful than named entity information, mostly due to the fact there is less of it present in the corpus. The binary feature S3 increases the F-score in almost 76% of cases, with an average of 1.68%. The explicit event type feature, S4, is less helpful, because of the sparsity of the data, improving the performance in 68% of feature combinations.

WordNet hypernyms, feature S5, helps slightly less than in the case of triggers.

For arguments, the feature combinations are improved in 69% of cases, by an average of 1.85%. The decrease is quite high as well, though, reaching an average of 0.73%. Recall is improved most, by 2.1% in more than 75% of cases, whilst precision is mostly affected, as 74% of cases decrease by an average of 1.5%.

Features S6 and S7, related to the mapping to UMLS types, behave differently from the previous ones. Whilst the binary S6 improves classification in more than 60% of cases, the multi-valued feature S7 manages to do so in only 30% of the cases. Furthermore, whilst the average increase is similar, the decrease of S7 is more than double that of S6.

The trigger annotation that is used for detecting both dependent and independent arguments, feature S8, is very helpful for two reasons. First, it excludes a number of tokens from being wrongly marked as arguments, thus reducing the number of false positives. Second, it indicates the approximate position of the dependent arguments due to the adjacency relation with the trigger. Thus, this feature increases the F-score in almost 85% of cases by an average of 2.56%, whilst the decrease is minimal, at 0.08% on average.

Feature S9, which flags tokens marked as dependent arguments, is also very helpful in determining the span of independent arguments. It improves the F-score in 82.59% of cases, by almost 2.40%, whilst the decrease is just 0.11%.

Finally, features S10 and S11, which determine whether and what type of semantic information the two arguments contain, prove very important. Whilst the recall is relatively insensitive to these two features, the precision is improved by around 4% when they are added in about 80% of cases.

6.7 Discussion

We have presented in this chapter several experiments that complete and show the viability of the task of recognising causal relations in biomedical scientific discourse. We proved that causal arguments can be extracted successfully, in a cascaded pipeline.

The two major factors influencing the automatic identification of causal arguments, the algorithms and features, are discussed in the following subsections.

6.7.1 Comparison of algorithms

Our experiments have shown that causal arguments are best detected in a semi-supervised setting for the argument position and span, whilst the argument role is better identified in a supervised manner. This is due to the fact that the errors occurring in previous steps are propagated and affect the performance of semi-supervised systems. Nevertheless, the performance between the supervised and semi-supervised is comparable at this last stage, even with error propagation.

For the first and third steps, we employed six different classifiers, one of them making its decisions based on the result of the other five. The wide spectrum of algorithms, ranging from Naïve Bayes to decision rules, decision trees and Support Vector Machines, provide complementary results which lead the Vote meta-classifier to outperform them by 2% for the first step and 3% for the third step.

For the second step, we modelled the task as a sequence labelling task using CRFs, and as a classification task using SVMs, RFs and NB. CRF performed best in this case, surpassing SVM and RF by approximately 5%, and NB by 16%.

The literature is very restricted from this point of view: most research is either based on CRFs, when researchers perform a token-level identification (Ghosh et al., 2011a; Stepanov and Riccardi, 2013), or on ME classifiers when they wish to obtain syntactic constituents that span the arguments (Lin et al., 2012; Xu et al., 2012).

6.7.2 Comparison of features

With respect to features, in all the experiments that we described, using features from all types produced the best results. This includes both domain-independent features, such as lexical, syntactic, dependency and command and positional features, and features specific to the biomedical domain, such as biomedical semantics. Semantics has proven to play a major role especially in the argument span and role recognition, where they improve the F-score by 3% on average.

The task of detecting the arguments of causal relations, and, more generally, discourse relations, has not been as studied as recognising triggers. Thus, the variety of features that have been employed until now is fairly limited. Do et al. (2011) use a complex semantic feature, measuring the similarity between two predicates, including their arguments, in the general domain, for the task of deciding whether or not the pair of predicates are in a causal relation. Their method takes into consideration just co-occurrence and various distances between the two predicates, but it manages to improve the F-score by 15% over that obtained by classical PMI, to 38%. It is recall that is increased significantly in this case, from 26% to 62%, when tested on PDTB.

Other methods restrict themselves to lexical and syntactic features. Ghosh et al. (2011b), Lin et al. (2012) and Xu et al. (2012) engineer a similar feature set to each other in their own approaches. Whilst Ghosh et al. (2011b) use a feature set composed of lexical features (surface expression and lemmata of tokens) and morpho-syntactic features (PoS, inflection, main verb of sentence, path from root to token in parse tree), Lin et al. (2012) extend it by adding information about the neighbouring tokens. Xu et al. (2012) enrich the set even more, considering the position of the token relative to the trigger (left or right), and its position in the sentence as a binary class (before the middle or after the middle of the sentence). Thus, they manage to reach 46% F-score in recognising both arguments when they employ automatic parses for feature extraction.

On biomedical text, the relevant literature is extremely limited. To the best of our knowledge, Ibn Faiz and Mercer (2013) describe the only method that identifies argument head words in the style of Wellner and Pustejovsky (2007). However, no decision is made on argument spans. Of note is the fact that their system has been built having the general domain in mind, and just applied on biomedical data. Thus, the framework does not use biomedically specific processing or features specific to the biomedical domain.

In conclusion, all feature types are needed for a better performance in discourse argument identification, as they complement each other. Whilst lexical and positional features increase precision, semantic and syntactic information boost recall.

6.8 Summary

In this chapter, we have presented our approach towards the automatic recognition of the arguments of discourse causal triggers. This is, to the best of our knowledge, the first detailed study of the problem of identifying the argument of discourse causal triggers in biomedical text, given a corpus of gold standard annotations.

As regards features, our experiments have shown that it is very important to employ features from various levels, i.e. lexical, syntactic, dependency, command, semantic and positional. These complement each other, with lexical and positional features ensuring high precision, and syntactic and semantic features providing generalisation and boosting recall.

We have split this task into three cascading steps, and applied an array of rules and machine learning algorithms for each of them. The first step, which tackles the position of the arguments, is cast as a binary classification problem, where classifiers decide whether or not the arguments are located in the same sentence. The Vote meta-classifier, which considers the output of other five classifiers, performs best, at almost

95% F-score. The second step regards the marking of the span of text which constitute the arguments. In this case, CRF outperforms the other classifiers, reaching 82.87% F-score. Finally, after the two argument spans are extracted, a decision is made with respect to the role they play in the causal relation. The roles of cause and effect are best assigned again by the Vote meta-classifier, with an F-score of 84.35%.

The numerous models that we have created have been evaluated on the BioCause corpus, in both a 10-fold cross validation supervised setting, and a self-learning semi-supervised setting. Due to the errors being propagated in the cascaded pipeline, the semi-supervised models for argument role recognition achieve slightly lower results than the supervised ones. However, the difference between the two approaches is of only around 1%.

Chapter 7

Metaknowledge of causality

Statements regarding causal associations have been long studied in general language, mostly as part of more complex tasks, such as question answering (Girju, 2003; Blanco et al., 2008) and textual entailment (Ríos Gaona et al., 2010). In spite of the more focussed and powerful analysis methods available today, typical discourse annotation efforts only focus on identifying the causal trigger and the two arguments that play the roles of Cause and Effect, and do not take into consideration the information regarding the context of discourse relations, although this is essential for their correct interpretation. However, more information is needed for the correct interpretation of these relations, and this is often present in discourse. For instance, negation plays an important role in contradiction detection, whilst determining the certainty level provides information about the confidence of authors. Additionally, it is necessary to automatically discover the novel parts of articles, as well as whether they are hypotheses, experiments, evaluations or results. The goal of capturing this type of interpretative information, explicitly or implicitly available in text, termed *meta-knowledge (MK)* (Thompson et al., 2011b), is to extract as much useful information as possible about causal associations in their textual context. This will further support the development

of information retrieval and extraction systems, the automatic discovery of new knowledge and the detection of contradictions.

In this chapter, we adapt an existing meta-knowledge annotation scheme (Thompson et al., 2011b) from biomolecular events to biomedical discourse relations, apply it to the causal associations existing in the BioCause corpus and analyse the resulting annotations. Furthermore, we train classifiers to automatically recognise meta-knowledge information and evaluate their performance based on the human annotations. To our best knowledge, our method is the first that is able to automatically identify and classify meta-knowledge information about causality in biomedical scientific discourse.

7.1 Related work

There exist several distinct efforts to capture various meta-knowledge dimensions in biomedical text, such as certainty (Kilicoglu and Bergler, 2008; Vincze et al., 2008), negation (Vincze et al., 2008; Nawaz et al., 2013a) or source (Liakata et al., 2010; Sándor and de Waard, 2012; Nawaz et al., 2013b), most of them related to biomedical events.

Regarding discourse, researchers have looked at articles as networks of hypotheses and evidence, and tried to identify the argumentation contained within a paper and the relationships between hypotheses, claims and evidence expressed in the article (de Waard et al., 2009). Others classified the discourse into discourse zones specific to scientific articles (e.g., background, methods, results) (Sándor, 2007). Another annotation scheme considers more than one aspect of meta-knowledge. For example, the ART corpus and its CoreSC annotation scheme Liakata and Soldatova (2009); Liakata et al. (2010) augment general information content categories with additional attributes, such as *New* and *Old*, to denote current or previous work.

Considering the mentioned work, we decided to create a resource of biomedical discourse causality enriched with relevant meta-knowledge information.

7.2 Annotation scheme

The original meta-knowledge annotation scheme is depicted in Figure 7.1. As can be noticed, it contains six dimensions, depicted in dark grey, which are centred on a biomedical event. These are *Knowledge type*, *Certainty*, *Polarity*, Source, and *Manner*.

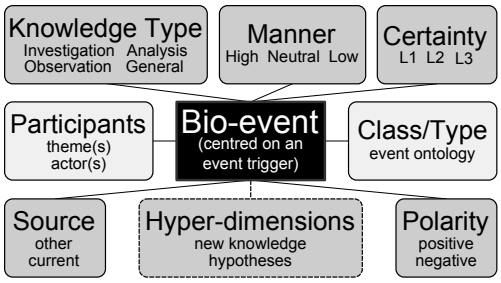


Figure 7.1: Meta-knowledge dimensions (from Thompson et al. (2011b)).

We adapted this meta-knowledge annotation scheme to the characteristics of discourse relations. All dimensions have been kept, with the exception of *Manner*, which is used to describe the change in intensity or speed of a biological process and does not have a correspondent in discourse. In what follows, we describe the adapted dimensions and categories.

7.2.1 Knowledge type

The *Knowledge Type* (*KT*) captures the general information about the content of the causal association, classifying it into five categories:

- *analysis*: inferences, interpretations, speculations or other types of cognitive analysis, always accompanied by lexical clues, typical examples of which include *suggest*, *indicate*, *therefore* and *conclude*.

- *fact*: events that describe general facts and well-established knowledge, and sometimes accompanied by lexical clues such as *known*.

- *investigation*: enquiries or investigations, which have either already been conducted or are planned for the future, typically accompanied by lexical clues like *examined*, *investigated* and *studied*.

- *observation*: direct observations, sometimes represented by lexical clues like *found*, *observed* and *report*, etc.

- *other*: the default category, assigned to associations that either do not fit into one of the above categories, do not express complete information, or whose *KT* is unclear or is unassignable from the context.

The original meta-knowledge *KT* dimension also includes a *Method* category, that is used to describe experimental methods, with clue words such as *stimulate* and *inactivate*. This category is not suitable for discourse, as intensity or speed does not apply to causality or other discourse relations.

7.2.2 Certainty

This dimension encodes the confidence or certainty level ascribed to the association in the given text. The epistemic scale is partitioned into three distinct levels:

- *L1*: explicit indication of either low confidence or considerable speculation towards the association or the association occurs infrequently or only some of the time.

- *L2*: explicit indication of either high (but not complete) confidence or slight speculation towards the association or the association occurs frequently, but not all of the time.

- *L3*: the default category. No explicit expression that either there is uncertainty or speculation towards the associations or that the association does not occur all of the time.

7.2.3 Source

The source of the knowledge expressed by the causal association is encoded as:

- *current*: the association makes an assertion that can be attributed to the current study. This is the default category, and is assigned in the absence of explicit lexical or contextual clues, although explicit clues such as *the present study* may be encountered.

- *other*: the association is attributed to a previous study. Explicit clues are usually present either as citations, or by using words such as *previously* and *recent studies*.

7.2.4 Polarity

This dimension identifies the truth value of the asserted causal association. A negated causal association is defined as one describing the non-existence or absence of a causal link between two spans of text. The recognition of such associations is vital, as it can lead to the correct interpretation of a causal association, completely opposite to that of a non-negated one.

- *positive*: no explicit negation of the causality. This is the default category, as most causal associations are expected to be positive.

- *negative*: the association has been negated according to the description above. The negation may be indicated through lexical clues such as *no*, *not* or *fail*.

MK subdim.	Kappa
Knowledge type	0.88
Certainty	0.89
Polarity	0.95
Source	0.94

Table 7.1: Inter-annotator agreement per MK dimensions.

7.3 Annotation process

We have applied the adapted meta-knowledge annotation scheme to all 19 full papers in the BioCause corpus, previously annotated with discourse causality associations. Previous studies have shown that the annotator background does not affect the consistency of the resulting annotations of meta-knowledge (Thompson et al., 2011b). Therefore, two annotators with background in computational linguistics and experience in meta-knowledge annotation have undertaken the annotation task. All causal associations have been annotated with meta-knowledge information. The two annotators have undergone a short training period, in which they have become accustomed to the annotation tool and guidelines and improved the agreement between them.

High levels of inter-annotator agreement have been achieved, falling in the range of 0.88 - 0.95 Kappa, depending on the MK dimension. The Kappa scores for each MK dimension are given in Table 7.1. The lowest Kappa occurs in the case of *KT*, as it is the most complex dimension to annotate. The five possible values can be confusing with specific relations which lie at the border between labels. The highest score is obtained in the case of *Polarity*, as it is fairly easy to recognise whether a relation is negated or not. The few problems that arose were in cases where the negation is implicit to the trigger itself. All disagreements have been discussed after the annotation and a final option has been agreed for each such disagreement by both annotators.

KT subdim.	Instances
Analysis	663
Fact	52
Investigation	2
Observation	62
Other	21

Table 7.2: Distribution of knowledge types in BioCause.

7.4 Manual analysis

Here we provide some key statistics regarding the causality annotation produced, together with a discussion of the characteristics of the corpus.

7.4.1 Knowledge type

Table 7.2 shows the number of causal relations of each category annotated with the *Knowledge Type* dimension. The most frequent annotated value is by far *Analysis*, constituting more than 82% of the total number of causal associations. This is not surprising, since most causal associations are the result of inference or interpretation of experimental results. Two other categories, *Observation* and *Fact*, are less frequently annotated, occurring in just over 6.5% of all annotations. *Investigation* appears even less, with only two instances in the entire corpus. The number of *Other* relations is 21 (2.63%)

There are several lexical clues that mark this MK category, as shown in Table 7.3. The most common is *suggest*, which occurs in almost 39% of the *Analysis* cases. The second most common is *indicate*, which occurs in almost 22% of the *Analysis* cases. Other clues include *demonstrate*, *thus* and *therefore*.

7.4.2 Certainty

The distribution of *Certainty* annotations is listed in Table 7.4.

KT subdim.	Frequent clues
Analysis	suggest (38.86%), indicate (21.68%)
Fact	shown to (60%), known to (20%)
Investigation	illuminate (100%)
Observation	observe (45%), report (30%)

Table 7.3: Most frequent clues for each KT category with their respective relative frequency (computed over the number of explicit clues) for that category.

Cert subdim.	Instances
L1	78
L2	382
L3	340

Table 7.4: Distribution of certainty levels in BioCause.

More than half of the causal associations in the corpus are expressed with some degree of uncertainty. That is, 50.63% of associations have been annotated with uncertainty clues, whilst 49.37% are certain or lack any uncertainty clue.

Under the speculated category, almost 92.10% (46.62% per total) of associations are reported with slight speculation (*L2*), whilst just under 8% (4% per total) are annotated as having a high level of speculation (*L1*). This is again an expected result, since most authors express their analyses with a high level of confidence.

The most frequent clues that lead to uncertainty are verbs, such as *suggest* and *indicate*, and modals, e.g., *may* and *might*. Nevertheless, there are several other types of uncertainty clues, such as adverbs (*likely*, *maybe* and *perhaps*), as shown in Table 7.5.

An interesting observation is that most of the uncertain associations (96.30%) belong to the *KT* type *Analysis*. There are very few instances of uncertain relation pertaining to other knowledge types. *Fact* has two relations (3.84%), whilst *Observation* has 12 relations (19.35%). Thus, almost 67% of all associations annotated as *Analysis* also have some degree of uncertainty.

Speculated relations are mostly part of the *Current* value of the *Source* dimension, and there are four negated speculated relations (44% of all negations).

Cert subdim.	Frequent clues
L1	may (45%), might (30%), perhaps (8%)
L2	suggest (51.8%), indicate (29.2%)
L3	definitely (40%), firmly (30%)

Table 7.5: Most frequent clues for each Certainty category with their respective relative frequency (computed over the number of explicit clues) for that category.

Source subdim.	Instances
Current	723
Other	77

Table 7.6: Distribution of source types in BioCause.

7.4.3 Source

Very few associations belong to the *Other* category, when compared to *Current*, as can be seen in Table 7.6. Just over 15% of all associations have their source in other articles, whilst 85% express knowledge created by the authors themselves.

Clues that are specific only to the *Other* category are citations to other articles, as shown in Table 7.7. Other clues are phrases such as *previously reported* and *X proposes that*, where X substitutes the names of researchers.

Source subdim.	Frequent clues
Current	in this study (67%), in this paper (17%)
Other	citations (84%), previously (8%)

Table 7.7: Most frequent clues for each Source category with their respective relative frequency (computed over the number of explicit clues) for that category.

Causal relations that have their source in other research are all positive from a *Polarity* point of view. However, they are not all completely certain: there are four instances which have *L1* as their *Certainty* level, whilst another 16 are *L2*. The rest of 57 are marked as *L3*.

The knowledge type of the causal relations is almost evenly split between *Analysis*

Pol subdim.	Instances
Positive	791
Negative	9

Table 7.8: Distribution of polarity values in BioCause.

(42 relations) and *Fact* (33 relations). There are one *Observation* and one *Other Knowledge type* relations from other sources. This fact is quite intuitive – most work already published tends to be treated as a fact or is analysed in connection with the research described in the current work.

7.4.4 Polarity

Table 7.8 shows the distribution of positive and negative causal relations in the BioCause corpus. As can be noticed, a small number of associations have been annotated with a *Negative* category in the *Polarity* dimension. Just over 1% of the annotations are marked as expressing a negated causality. This is to be expected, since, in scientific discourse, authors tend to present their positive results instead of negative ones. Nevertheless, it is vital to detect such information, since a simple negation completely changes the meaning of a causal relation.

Table 7.9 lists the most common cue expressions for negated relations. As can be observed, clues for negations are varied, some belonging to closed-class parts-of-speech, e.g. determiners (*no*), adverbs (*not*) or prepositions (*against*), whilst others belong to open-class parts-of-speech, such as verbs (*rule out*) and adjectives (*impossible*). Nevertheless, the adverbial *not* is the most frequent, accounting for almost two thirds of negated causal associations.

Negated causal relations always have the *Source* dimension set to *Current*. It is very unlikely that authors of one study directly contradict causal relations described in other research.

Pol subdim.	Frequent clues
Negative	not (62.5%), no (15%), against (7%), rule out (4%)

Table 7.9: Most frequent clues for each MK category with their respective relative frequency (computed over the number of explicit clues) for that category.

Furthermore, five out of the nine negated relations have the *Certainty* level set to *L3*. Two relation is set to *L1*, and another two to *L2*.

Looking at negated relations from a *Knowledge Type* perspective, seven relations are of type *Analysis*, whilst two are marked as *Observation*. The lack of occurrence of negative instances amongst the other types of *Knowledge Type* is to be expected, as it is usual that researchers investigate why events occur, and not why they do not.

7.5 Automatic identification of meta-knowledge

We have experimented with several supervised machine learning algorithms in the task of automatically classifying causal discourse relations from the point of view of each MK dimension. The learners have been trained on a large feature set, including the clues mentioned above. Lexical features are the most important, as they provide direct information to classifiers. Having binary features that flag the presence of negation particles or modal verbs helps ML algorithms make better decisions.

Furthermore, syntax provides good support for the generalisation of triggers and their associated meta-knowledge. These are extracted from automatic parses created by the Enju system (Miyao and Tsujii, 2008) trained on GENIA. Syntactic features include PoS, syntactic category, dependency, constituency and c-command information. They are similar to those used for trigger detection and previously described in Section 5.3.2.

Besides lexical and syntactic features, the algorithms have learned using a semantic layer of annotations. These come from the gold standard named entities and events in

BioCause, as well as UMLS, OSCAR, NeMine, and Europe PMC. Furthermore, features have been extracted from a context window spanning the full sentence in which the trigger is located.

We built separate models for each MK dimension. We have used seven different classifiers, from various categories: SVM, RF, NB, JRip and J48 as classifiers, Vote as a meta-classifier based on the previous five, and a rule for the baseline. The baseline for each MK dimension is the majority class rule, which tags all instances as belonging to the class with most instances.

Algorithm	**KT**	**Certainty**	**Polarity**	**Source**
Majority	18.15%	21.53%	49.72%	47.51%
SVM	36.31%	87.40%	84.02%	68.25%
Random Forest	34.76%	83.53%	79.97%	73.77%
JRip	29.28%	77.92%	84.02%	71.52%
J48	25.45%	83.75%	49.72%	47.51%
Naïve Bayes	32.96%	77.49%	61.17%	62.35%
Vote	41.69%	84.62%	79.97%	70.87%

Table 7.10: Macro-average F-scores achieved by various learners per each MK dimension.

The overall macro-average F-score results are given in Table 7.10. All results have been 10-fold cross validated. Details on the performance and error analysis for each dimension are given in the following subsections. Due to the highly skewed data, we present both macro- and micro-average F-scores in the detailed tables.

7.5.1 Knowledge type

Table 7.11 lists the detailed performance of the employed classifiers in the task of detecting the *Knowledge Type* of causal relations. It includes the macro-average precision, recall and F-score, as well as the micro-average F-score. The large difference between the two scores comes from the fact that this is a five-way classification, corresponding to the five subdimensions of *KT*, and that the data is very skewed across

Algorithm	ma P	ma R	ma F_1	mi F_1
Majority	16.62%	20.00%	18.15%	75.40%
SVM	39.94%	33.38%	36.31%	82.60%
Random Forest	39.12%	31.28%	34.76%	82.20%
JRip	40.96%	22.78%	29.28%	77.60%
J48	27.50%	23.68%	25.45%	77.90%
Naïve Bayes	30.52%	35.82%	32.96%	74.50%
Vote	54.64%	33.70%	41.69%	83.80%

Table 7.11: Performance of various classifiers in identifying the *Knowledge Type* of causal relations.

these five subdimensions.

As can be noticed, all classifiers perform better than the baseline in a macro-average setting. However, in a micro-average context, Naïve Bayes is confused by the data imbalance and is outperformed by the Majority rule by almost 1%. The best performing classifier is the Vote meta-classifier, which reaches 83.80% micro-average F-score and 41.69% macro-average F-score. It also obtains the best precision and recall amongst all classifiers, in both macro- and micro-average settings.

Most errors arise because of the skewed distribution of the labels. For instance, for Vote, there are only eight false negatives for the *Analysis* label, but 82 false positives are generated. The two instances in the *Investigation* label are erroneously assigned to *Analysis*. This proves the tendency of the classifiers to assign most instances from minority classes to the majority class.

7.5.2 Certainty

A detailed account of the performance of the classifiers is given in Table 7.12. Unlike in the case of *Knowledge type*, the difference between macro- and micro-average is much smaller. This is due to the fact that there are only three possible labels that a classifier can assign.

The best results are obtained by the SVM classifier, which reaches 90.90% micro

Algorithm	ma P	ma R	ma F_1	mi F_1
Majority	15.90%	33.33%	21.53%	47.70%
SVM	89.20%	85.67%	87.40%	90.90%
Random Forest	88.00%	79.50%	83.53%	87.70%
JRip	91.40%	67.90%	77.92%	87.70%
J48	87.27%	80.50%	83.75%	81.30%
Naïve Bayes	76.03%	79.00%	77.49%	83.70%
Vote	87.33%	82.07%	84.62%	88.60%

Table 7.12: Performance of various classifiers in identifying the *Certainty* of causal relations.

F-score and 87.40% macro F-score. Class *L1* is recognised with the lowest precision and recall amongst the three classes, due to its low number of instances. The low scores of Naïve Bayes and J48 damages the performance of the Vote meta-classifier, which is the second best amongst all algorithms.

The most important features for this dimension are, as expected, the certainty clues previously described. The fact that triggers contain words such as *may*, *probably*, *suggest* or *can* is a good indicator for the correct certainty level.

Many of the error cases happen between the two uncertain classes, *L1* and *L2*. It is usually the case that *L1* relations are wrongly classified as *L2*. Furthermore, there are several instances of mostly *L2*, but also *L1*, classified as *L3* and vice-versa. For instance, in example (7.1), the causal relation is speculated, but the model decided that it is certain and belongs to *L3*.

(7.1) [32] has shown that mutation of phosphotransferase system (PST) in extraintestinal pathogenic E. coli (ExPEC) *can cause* the loss of its colonization ability in extraintestinal organs, and bacteria are cleared rapidly from the bloodstream.

Algorithm	ma P	ma R	ma F_1	mi F_1
Majority	49.45%	50.00%	49.72%	98.30%
SVM	91.40%	77.75%	84.02%	99.30%
Random Forest	89.70%	72.15%	79.97%	99.10%
JRip	91.40%	77.75%	84.02%	99.30%
J48	49.45%	50.00%	49.72%	98.30%
Naïve Bayes	58.90%	81.65%	61.17%	97.30%
Vote	89.70%	72.15%	79.97%	99.10%

Table 7.13: Performance of various classifiers in identifying the *Polarity* of causal relations.

7.5.3 Polarity

The *Polarity* of causal relations is the most correctly recognised MK dimension amongst all four in terms of micro-average F-score, and the results for it are shown in Table 7.13. This is due to the fact that this dimension has the most skewed label distribution of all: 9 negative to 791 positive instances. As a consequence, the baseline is very high as well, reaching 98.30% micro F-score, but just under 50% macro F-score.

The best overall results are obtained by SVM and JRip, in both macro- and micro-average settings. However, amongst all classifiers, Naïve Bayes manages to identify correctly most of the minority class instances, reaching a recall of 66.67%. In contrast, its recall for positive instances and precision for negative instances are the lowest, a fact which affects the final micro-F-score, making it perform worse than the baseline rule in a micro setting. In addition, the low performance of Naïve Bayes, as well as that of J48, influence negatively the result of the Vote meta-classifier, which gets the second best result.

The most salient features are the placement of negation particles in the vicinity of triggers. This leads to some error cases arising from those triggers which are negated not by the use of negating particles (e.g., *not*), but by using inherently negative triggers, such as in example (7.2). The verb *rule out* implicitly suggests a negative polarity. However, the sparse data regarding relations negated by such means affects its correct

recognition.

(7.2) Therefore, the DNA-induced resistance of biofilms requires both the cultivation and challenge under cation-limiting conditions.

These latter two observations *rule out* the possibility that negatively charged DNA simply interacts with cationic antimicrobial peptides and prevents their access to bacterial cells.

7.5.4 Source

The results of the classifiers in the case of the *Source* of causal relations are shown in Table 7.14. The best micro performance is achieved by the JRip classifier, at 90.20% F-score, whilst the best macro result is obtained by Random Forest, at 73.77%. The difference between these two classifiers is not that large, being less than 2% for macro and just 0.40% for the micro F-score. The main problem of these two classifiers is the low recall for the *Other* label, which is under-represented when compared to the *Current* label. The best recall for this class is achieved by Naïve Bayes, which captures 47.40% of its instances. However, the precision drops significantly to only 24%, whilst JRip and Random Forest reach up to 80%.

Most errors occur when instances of *Other* are classified as *Current*.

7.6 Summary

This chapter has described our approach to the enrichment of the BioCause corpus, which contains discourse causality associations, with meta-knowledge information.

Algorithm	ma P	ma R	ma F_1	mi F_1
Majority	45.25%	50.00%	47.51%	86.00%
SVM	77.30%	61.10%	68.25%	89.60%
Random Forest	86.15%	64.50%	73.77%	89.80%
JRip	84.30%	62.10%	71.52%	90.20%
J48	45.25%	50.00%	47.51%	86.00%
Naïve Bayes	59.00%	66.11%	62.35%	83.40%
Vote	84.85%	60.85%	70.87%	89.90%

Table 7.14: Performance of various classifiers in identifying the *Source* of causal relations.

This type of contextual information regarding causal relations is crucial for their correct interpretation. Modifiers such as *not* and *might* completely alter the meaning and certainty of a relation, especially when placed in the context of a network of causal relations. Furthermore, it is important to recognise what type of knowledge the causal relations refer to and whether it is new or old knowledge. This helps the creation of new, testable hypotheses and the assignment of literature support to those relations which contain references.

We have adapted an existing meta-knowledge annotation scheme designed for biomedical events to the needs of discourse analysis. The annotation has been performed by two humans, and the inter-annotator agreement between them is high, ranging between 0.89 and 0.95 Kappa.

A manual analysis of how causality associations are expressed in the biomedical domain has been performed. This shed light into what phrases are used to convey negation, uncertainty, various knowledge types and source of statements.

Additionally, machine learners have been trained to automatically identify the value of each MK dimension for each causal relation. The algorithms base their decisions on a mixture of lexical, syntactic and semantic features, most of which are produced from automatic parses by off-the-shelf systems. Considering the skewness of the data, the classifiers perform reasonably well. SVM obtains the best scores in the case of

Certainty and *Polarity*, whilst Random Forest is the best at recognising the *Source* dimension. The best model for *Knowledge Type* considers all five algorithms combined by the Vote meta-classifier. Since the data is so sparse for some dimensions, more would be welcomed in order to be able to create more accurate models.

Chapter 8

Question generation using discourse causality

The previous chapters have dealt with the recognition of causal relations from biomedical scientific discourse. These relations, however, are of limited use if they are not leveraged in other real-world tasks and applications, where they can reduce the effort of users such as biomedical researchers.

We now focus our attention on the generation of questions based on discourse relations. The applications of questions generated in this manner are numerous. For instance, it is a novel way of allowing users to query large collections of scientific papers, looking for facts rather than documents. By putting together ordinary search terms (e.g., proteins, genes or drugs), queries are generated and evaluated and only the facts that match the query terms within individual document sentences are returned to the user. However, many submitted queries are incomplete, especially when the desired response is not well defined in the mind of the users. Thus, by creating a query suggestion mechanism in the form of questions proves to be of great help in the searching process, since this shows, in natural language, the most common associations of the already input terms. Other uses for question generation are the automatic creation

of multiple-choice tests and experimental hypothesis production.

Since BioCause contains causal relations, the obvious questions that can be generated are *Why*-questions. This type of question is one of the more complex that can be asked (Graesser et al., 2009), as they require more logical thinking in both creating the question, but also in finding the answer to them.

There have been numerous attempts at developing methodologies for automatically generating questions, either independent or as part of shared tasks (Rus and Graesser, 2009; Rus et al., 2010). Most research focusses on the general domain, although effort has been invested in domain-specific applications too. The approaches in the literature can be split into three main approaches: template-based, syntax-based and semantics-based. Template-based methods, such as those described by Mostow and Chen (2009) and Chen et al. (2009), have been developed to produce questions from children stories and informational text. This method has been chosen due to the restricted variability of the desired questions and the closed-domain in which they were applied. Approaches based on syntax (Wyse and Piwek, 2009; Heilman and Smith, 2009) are based on the manipulation of parse trees, using transformation rules manually designed by linguists. Semantics has been less explored compared to the previous two approaches. The method of Schwartz et al. (2004) represents semantic relationships in a logical form and uses these for generating *Wh*-questions, whilst Yao et al. (2012) decompose and simplify complex sentences and rank multiple question candidates.

Since our work focusses specifically on biomedicine and the question types are based on causal discourse relations, a combination of rules and syntactic transformations is suitable to generate natural language questions.

8.1 Automatic question generation

The process of question generation is a two-step process. The first task refers to selecting the content that will form the actual question. Following this, the second step deals with formulating the question, ensuring its grammaticality. Both steps are individually discussed in the following subsections.

8.1.1 Content selection

Content selection is a major problem in the task of natural language generation, including question generation. In the specific case of question generation, the content is the section of text over which the question has to be asked. The size of the content ranges significantly depending on the type of question to be asked, from a single phrase or clause for very specific questions to entire paragraphs for more general questions.

As this work deals with recognising discourse causal relations, this is the basis for the process of content selection. The causal relations in BioCause are used to identify the possible target content for questions. The targets are select from the two arguments of each relation. Either the cause or the effect can become a question, thus resulting in two question types: causal antecedent and causal consequence questions.

After the content has been selected, it is passed to the question formulation module, which identifies the appropriate question type and transforms the statement into a question.

8.1.2 Question formulation

Formulating the question correctly depends largely on the type of causal relation. Although the BioCause corpus contains only causal relations, two types of questions can be created: *Why*-questions and *What*-questions. After the question type is established, the content needs to be transformed from its statement format into a question format.

Question type identification

The trigger of the causal relation influences the type of question that is to be asked. Based on lexical and syntactic patterns, we develop a simple heuristic which decides on what question type will be used.

For instance, triggers of the type *X suggests Y* can easily produce *what*-questions: *What suggests Y?*. Many questions can be in fact *What*-questions, and they are formulated from triggers that contain a VP. They are usually centred around keywords such as *suggest*, *indicate* and *demonstrate*.

In contrast, causal triggers like *because* and *since* are suitable to be transformed only into *Why*-questions. These triggers cannot be used to formulate any other question type so that the final question reads naturally and sounds idiomatic.

An important aspect to note is that the questions produced in the first case can have their type changed for the same causal relation. Instead of creating a *What*-question based on the causal trigger, it is possible to create a *Why*-question based on the argument of the trigger that has been selected, in a similar manner to the second case. However, *What*-questions do not require a main verb transformation, since the triggers already contain a VPs to serve as the main verb of the question. Thus, for such available triggers, it is preferable to create a *What*-question, as the main verb transformation can introduce errors into the pipeline.

Main verb transformation

This stage deals with identifying the verb complex of the argument that will become the question, and transforming it in order to make it suitable for a question.

There are several transformation steps for adapting the verb of a statement into that of a question. First, the verb complex must be identified from the dependency parse of the argument. Such verb complexes can consist of the main verb, along with any

modals and auxiliaries that accompany it. Extracting the entire verb complex is an important step, as an incomplete extraction will result in an ungrammatical structure of the sentence.

An argument containing a modal is shown in example (8.1). In the second sentence, which contains both the causal trigger and the *Effect* argument, the verb complex *may involve* needs to be separated in order to ensure a correct syntactic structure. Other modal verbs are *might*, *would*, *could*, *should*, *will* and *can*.

(8.1) That PmrB is likely to sense changes in pH directly is supported by three findings: (i) the mild acid pH-dependent activation of the PmrA-regulated gene pbgP was dramatically reduced in a strain lacking pmrB.
Therefore, regulation of PmrB activity **may involve** protonation of one or more of these amino acids.

Any auxiliaries, such as *is* in the sentence given in example (8.2), need to be separated in a similar manner to modals. Other lemmatised auxiliary verbs are *have* and *do*, but their possible inflections also occur in text.

(8.2) Upregulation of the actP and acs genes in the flea, which direct the uptake of acetate and its conversion to acetyl-CoA, also *suggests that* insufficient acetyl-CoA **is produced** by glycolysis to potentiate the TCA cycle.

Finally, the presence of other particles affecting the verb, such as negation particles, illustrated in example (8.3), should not influence the verb transformation of the question.

PoS	Aux
VBD	did
VBZ	does
VBP	do

Table 8.1: Addition of the support verb *do* in questions.

(8.3) This acid pH-promoted increase appears to be specific to a subset of PhoP-activated genes (our unpublished results) that includes pmrD *because* expression of the PhoP-regulated slyA gene and the PhoP-independent corA gene **was not affected** by the pH of the medium.

If the verb complex that has been extracted contains modals or auxiliaries, these need to be separated and pre-pended to the argument. These will form the main verb of the question to be generated. If both modals and auxiliaries are present, only the modals get extracted from the verb complex, whilst the auxiliaries remain in the complex. Regardless of whether it is a modal or auxiliary that is extracted, the token must be completely removed from inside the argument.

Otherwise, if the verb complex does not include auxiliaries or modals, a support verb needs to be added instead. However, this verb needs to agree syntactically with the subject of the verb complex, whilst the verb complex needs to be changed to its lemmatised form. Thus, the PoS of the verb complex is analysed and a set of rules rely on this to decide on the tense and number of the support verb. Table 8.1 lists the rules for the addition of the support verb. A past tense complex will result in the addition of *did*, whilst a present tense singular third person verb complex will pre-pend *does*. The default rule is the addition of *do*, which is added in all other cases.

Finally, there are two more additions that need to be performed in order to complete the question. First, the question type needs to be pre-pended in front of the transformed text. Second, a question mark needs to be appended to the question, thus finalising the

process of question generation.

8.2 Question evaluation

The evaluation of automatically generated natural language text is a very difficult task. Thus, we employ two human evaluators for scoring the questions output by the system. The two humans have nearly native English proficiency.

The grading system uses a scale of 1 to 4, with 1 being the lowest and 4 being the highest grade. The evaluators assign a grade to check the correctness of the syntactic structure and another one for the semantic content. Although the causal relations are manually annotated by domain experts, the semantics can suffer after going through the steps of question transformation. Thus, each question is marked out of eight points.

The syntactic correctness is evaluated to ensure the grammaticality of the output, as well as the fluency of the question. The syntactic correctness and fluency are evaluated as follows:

- 4: the question is grammatically correct and reads naturally. For instance, the question *Why is the rv3612c-rv3616c gene cluster regulated by PhoP?* would be marked with this score.

- 3: the question is grammatically correct, but does not read naturally. An example of such a question is *Why is, like GAS M1 Mac [7,8], SeMac a cysteine endopeptidase?*.

- 2: there are some grammatical problems. The following sentence has a missing determiner for the noun *group*: *Why did application of the CLR algorithm identify very interesting group of genes that are co-regulated with SPI-2 and were horizontally transferred to Salmonella?*.

- 1: the quality of the grammar is unacceptable. For example, the sentence *Why can use multiple carbon sources and terminal electron acceptors?* does not have a subject.

The quality of the semantics is evaluated in a similar manner, using the following criteria:

- 4: the question is semantically correct and reads naturally. For instance, *Why does the DeltavicK mutant retain the ability of S. equi to resist to phagocytosis by PMNs?*.

- 3: the question is semantically correct and close to the text or other questions. One such case is *What confirms that salKR had been deleted from the bacterial chromosome?*, where the identity of the bacterium under study is mentioned previously in discourse.

- 2: there are some semantic issues. An example of such as question is *Why has SeMac other unknown function?*, where some information for complete understanding is missing.

- 1: the semantics of the question is unacceptable. For example, the following sentence includes an unresolved anaphor: *Why is chelation a general property of this negatively charged polymer?*

A total of 555 questions have been scored independently by the two human evaluators. The agreement between the two annotators has been measured using Cohen's κ. The value of κ is 0.92, indicating a high level of agreement.

Figures 8.1 and 8.2 show the distribution of scores for the two evaluators for the syntactic and semantic dimensions, respectively. As can be noticed, the evaluators have similar distributions across the two dimensions.

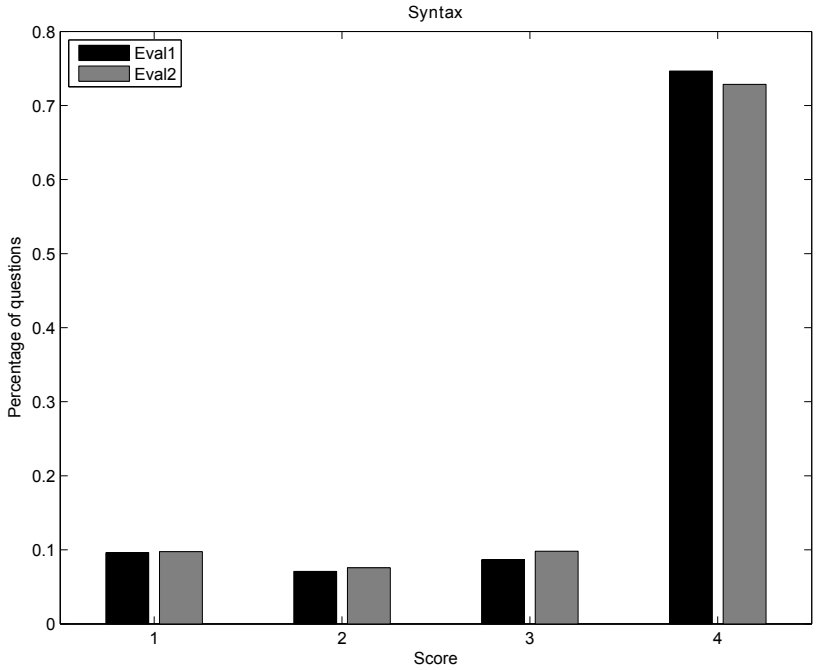


Figure 8.1: Syntactic evaluation of the generated questions by the two evaluators.

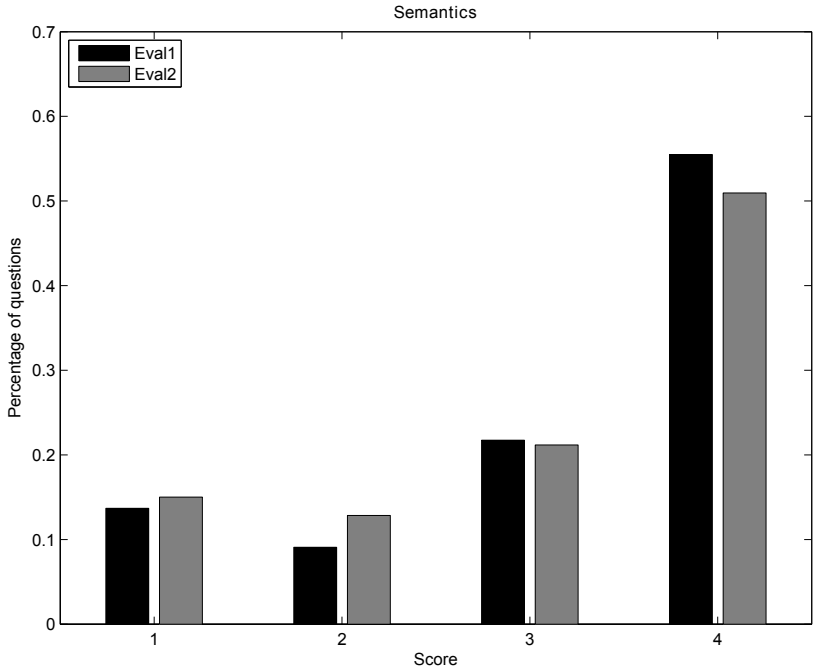


Figure 8.2: Semantic evaluation of the generated questions by the two evaluators.

Dimension	Syntax	Semantics
Eval1	3.48	3.19
Eval2	3.46	3.08
Final	3.47	3.13

Table 8.2: Average scores as assigned by the two evaluators and final scores after adjudication for each dimension.

Table 8.2 shows the average scores assigned by the two evaluators for each dimension. It can be noticed that syntax obtained higher scores than semantics. This can be explained by the high degree of specialisation of our task. Generating questions from discourse relations by using simple rules which only change the main verb should result in a correct syntax in most cases. Semantics, however, reaches a lower average score, mostly due to the fact that the discourse relations contain a large number of coreferential links, which are not tackled in our proposed method. Additionally, the difference between the two annotators is larger in the case of semantics, as this dimension is more subjective. After adjudication, the average score for syntax is 3.47, whilst the quality of semantics is graded as 3.13.

Looking at the disagreements between the two annotators, the largest proportion of differences occurs between the second and third categories of each of the scoring dimensions. This is to be expected, since these are the most subjective categories. Although it is easy to decide whether a question is fully correct or completely wrong syntactically and semantically, deciding on the gravity of the error in intermediate cases is more difficult and depends on each person's interpretation.

Example (8.4) shows one question in the case of which the two evaluators did not agree on the score. One evaluator assigned the maximum score, whilst the other penalised the inclusion of the subordinate clause, as it overloads the question.

(8.4) Why may horse IgG3 be also cleavable by SeMac, while the other five horse

IgG subgroups may not be cleaved?

In the case of disagreement on semantics, example (8.5) includes such a question. One annotator assigned a score of 2, whilst the other marked it as 3.

(8.5) What demonstrates that an ATP-saturated form of dimeric Rv2623 (composed of 2 bound ATP molecules per monomer) constitutes at least half of the purified sample?

8.3 Error analysis

There are several types of errors that have occurred in the question generation process. These are either related to poor syntax or to poor semantics. However, an incorrect syntactic structure is the predominant production error.

The use of coreferences is a common property of natural language, ensuring discourse cohesion and coherence. However, extracting pieces of text containing coreferential expressions without properly resolving these coreferences as an initial step to extraction will result in an incorrect semantic structure. The unresolved anaphors thus lead to a lower semantic rating for the questions. Take, for instance, the text in example (8.6).

(8.6) Interestingly, [both glutamate and GABA are important neurotransmitters at the neuromuscular junction of insects, and the concentration of glutamate is very low in insect hemolymph]$_{Cause}$, *suggesting that* [it is converted to glutamine before

it is absorbed]$_{Effect}$.

The question generated from this text is included in example (8.7).

(8.7) What suggests that it is converted to glutamine before it is absorbed?

Although syntactically correct, the question cannot be understood and interpreted correctly since the included context is not sufficient.

Another source of errors is the fact that humans produce errors when writing the text. Although this problem could be tackled automatically, it is beyond the scope of this application. One of the most common human mistakes is the lack of determiners for nouns. Take, for instance, the text in example (8.8), where a determiner is necessary for the bolded noun *property*.

(8.8) Thus, [the ability of extracellular DNA to chelate magnesium is independent of origin and molecular weight]$_{Cause}$, *indicating that* [chelation is general **property** of this negatively charged polymer]$_{Effect}$.

The question generated from this text is included in example (8.9).

(8.9) What indicates that chelation is general property of this negatively charged polymer?

The problem of lacking idiomaticity and naturalism for the produced questions also occurs, but is again not tackled by our methodology. The removal of predicate

adjuncts, which overload a question, can be performed by pruning the parse tree of the argument prior to the main verb transformation. Take, for instance, the text in example (8.10).

(8.10) [S. equi DeltavicK mutant does not grow as well as the wild-type strain in both THY and blood, suggesting that the vicK deletion causes defect in growth, a plausible reason that likely contributes to the attenuation of S. equi virulence in the mouse infection models]$_{Effect}$.
This suggestion is further supported by the observations that [both the wild-type and DeltavicK mutant strains are resistant to phagocytosis by PMNs]$_{Cause}$, which suggest that VicRK is not required for the evasion of S. equi to the innate immunity.

The question generated from this text is included in example (8.11).

(8.11) Why does S. equi DeltavicK mutant not grow as well as the wild-type strain in both THY and blood, suggesting that the vicK deletion causes defect in growth, a plausible reason that likely contributes to the attenuation of S. equi virulence in the mouse infection models?

8.4 Summary

This chapter has focussed on proving the viability of an application of recognising discourse causality from biomedical scientific text. The chosen application, question generation, can be used in numerous tasks and fields, ranging from improving the user experience in searching to test question creation and hypothesis production.

The process of generating natural language questions from natural language statements has been split into two main steps, both of which are based on rules. The first step deals with deciding, based on the trigger of the causal relation, on the type of question to be asked, i.e. *What* or *Why*-questions. Second, we have engineered several heuristic rules to transform the text from its statement format to the appropriate question format.

The automatically generated questions have been manually evaluated by two English-speaking humans, who scored their correctness from both syntactic and semantic points of view. The inter-annotator agreement between the two evaluators reaches a κ score of 0.92, suggesting a high degree of agreement. The 555 questions that have been evaluated obtained, on average, a score of 3.47 for syntactic correctness and 3.13 for semantic correctness. Most syntactic errors arise from including too many predicate adjuncts in the question, whilst semantics is affected mainly by unresolved anaphora. Both these issues can be addressed automatically by improving the processing in the second step.

Chapter 9

Concluding remarks

This chapter summarises this study and provides an outline of further research directions. First, we evaluate the progress against research objectives and hypotheses established in the beginning of the project. We review the contributions of this study and summarise the main findings described in the preceding chapters. The chapter concludes with a discussion on the main areas of future work. We provide an insight into how our contributions can be applied to the further development of biomedical discourse causality recognition and other related fields.

9.1 Review of the contributions

The main goal of the research described in this thesis has been to investigate the use of NLP techniques with the purpose of automatically recognising causal relations in biomedical scientific discourse. To achieve this goal, five objectives were established at the beginning of the project, each corresponding to a research question. Their accomplishment is evaluated individually in what follows.

9.1.1 Objective 1

O_1 to develop a manually annotated corpus of biomedical scientific literature with relevant discourse causality information.

This objective has been achieved by developing a manually annotated corpus of discourse causal relations in biomedical scientific articles, i.e. the BioCause corpus. We trained two independent annotators with high levels of education and experience in biomedical sciences, as well as extensive annotation experience, to perform causal relation annotations. The BioCause corpus was created by adding the new layer of annotations on top of existing biomedical named entity and event information in the BioNLP Shared Task on Infectious Diseases. The corpus contains 19 full-text open-access journal articles, and has been enriched with 850 causal relations, of which 800 are explicit and 50 are implicit.

The results of the annotation process, including the annotation scheme and evaluation, as well as a thorough characterisation of causality in biomedical text, have been published in the journal BMC Bioinformatics (Mihăilă et al., 2013).

9.1.2 Objective 2

O_2 to develop a methodology that can recognise discourse causality in biomedical literature.

The problem of recognising discourse causality has been split into two steps, namely identifying causal triggers and extracting their arguments. Whilst trigger detection is a fairly straightforward task, recognising the arguments has been further split into three cascaded sub-steps: finding their position, their spans and, finally, their role.

All four low-level tasks have been tackled by multiple approaches. Firstly, we engineered rules and heuristics to establish baselines for each task. Secondly, we developed supervised models using an extensive array of classifiers, modelling the tasks

as sequence labelling or supervised classification. These models obtain superior results to those in the case of rules. Finally, to overcome the small and sparse dataset, we enriched the labelled data with a large amount of unlabelled data. Thus, the performance of these latter models increases even more, by several percentage points over the supervised ones.

All approaches have been published in various conference and journal articles: trigger detection and comparison to BioDRB (Mihăilă and Ananiadou, 2013a,b), a hybrid approach for trigger and argument detection (Mihăilă and Ananiadou, 2013c), and the semi-supervised approach (Mihăilă and Ananiadou, In press).

9.1.3 Objective 3

O_3 to identify useful features for recognising biomedical discourse causality.

Following the numerous experiments performed to accomplish objective O_2, we performed an extensive analysis to identify the most useful features for recognising biomedical discourse causality. Each feature that has been created in this research has been evaluated individually to assess its contribution to the task in which it has been used. We analysed the interaction with other feature sets, and measured the average increase and decrease in precision, recall and F-score.

An evaluation of features used at various steps in the process of recognising causality has been published at various conferences and in various journal articles, as they have been introduced (Mihăilă and Ananiadou, 2013a,b,c, In press).

9.1.4 Objective 4

O_4 to develop a manually annotated corpus of biomedical discourse causality with meta-knowledge information.

This objective has been achieved by enriching the previously created BioCause corpus with manual annotations regarding meta-knowledge information for all existing causal relations. We trained two independent annotators with extensive annotation experience to perform meta-knowledge annotations. The inter-annotator agreement is high for all four dimensions, i.e. polarity, certainty, knowledge type and source.

The results of the annotation process, including the annotation scheme and evaluation, of meta-knowledge for causal relations in biomedical discourse, have been published at the 9th Conference on Language Resources and Evaluation (Mihăilă and Ananiadou, 2014).

9.1.5 Objective 5

O_5 to investigate the automatic recognition of the meta-knowledge information of biomedical causal relations.

Based on the annotations from objective O_4, we trained four machine learning models, one for each meta-knowledge dimension. The various learners employed have been trained on a large feature set, including lexical, syntactic and semantic features. The high performance, reaching over 88% F-score, proves the feasibility of identifying such types of information.

The methodology for and results of the automatic recognition of meta-knowledge of causal relations in biomedical discourse have been published at the 9th Conference on Language Resources and Evaluation (Mihăilă and Ananiadou, 2014).

9.2 Review of hypothesis

H_0 Discourse causality in biomedical scientific literature exhibits significant and measurable differences, which can be captured through statistical and linguistic

indicators.

Having accomplished all objectives that were initially proposed, it can be stated that the hypothesis is proven. Discourse causality does exhibit significant and measurable differences at various levels of analysis, i.e., lexical, syntactic, dependency, command, semantic and positional. These differences can be successfully leveraged by machine learners to satisfactorily identify causal relations in text.

9.3 Future directions

The work presented in this thesis leaves unexplored certain aspects of discourse causal relations. Investigating these aspects could lead to an improvement in performance. Furthermore, multiple threads of research can be created based on this work.

One obvious extension is resolving the anaphora that occurs in the discourse. There are two main types of anaphoric expressions whose resolving could have a positive impact on the detection of causal relations. Firstly, there is the issue of anaphoric shell nouns, which frequently occur in causal triggers. These anaphors usually point to the independent argument, so the low performance in recognising these arguments could be significantly improved. Nevertheless, this topic is still emerging and scientists are still discussing on the definition of and annotation schemata for anaphoric shell nouns (Kolhatkar et al., 2013). Secondly, resolving nominal anaphors could increase the performance in multiple places in the processing pipeline. For instance, the semantic type of their antecedents will provide more information to machine learners, which can then better recognise argument spans and classify the argument roles. Furthermore, this will significantly improve both the syntactic and semantic quality of the generated questions. By recognising its antecedent, an anaphoric expression can then be easily replaced in the argument that forms the base for the question, thus removing semantic

ambiguity. Resolving zero anaphora will ameliorate both syntactic and semantic ambiguities.

Another idea is to extract multiple argument candidates for each trigger and then submit them to a ranking algorithm. Such an algorithm can select the best candidate based on various criteria and filters, that can be suited to user needs or specified by users themselves. A good starting point is the work of Wellner (2009), which can be extended by integrating one or more causal measures into the ranking mechanism.

An interesting extension would be the annotation of such discourse causal relations in other biomedical sublanguages. Whilst BioCause contains only articles on infectious diseases, a different subdomain, such as Neurology or Neoplasms, might shed some light onto whether causality is expressed differently between subdomains. Although it is now known for a fact that there are significant differences at lexical, syntactic and semantic levels, discourse has not been investigated from this point of view.

Besides the research in recognising discourse causality, other biomedical and NLP fields can benefit from this work. As a fundamental discourse relation, causality plays an important role in many daily life applications.

For instance, in the biomedical domain, epidemiologists study the patterns, as well as the causes and effects of health and disease conditions in specific populations. Epidemiology is thus the centrepiece of public health, being pivotal to health policy decision making and evidence-based practice by identifying targets for preventive healthcare and risk factors for diseases. Being able to quickly analyse large amounts of documents and correctly recognise causal relations between relevant facts can improve significantly both the speed and quality of making decisions affecting the public. In order to achieve this, it is necessary to be able to recognise, extract and analyse, in an automatic manner, the patterns that occur in defined populations. For this mechanism to function in a realistic manner, several sources of information need to be brought

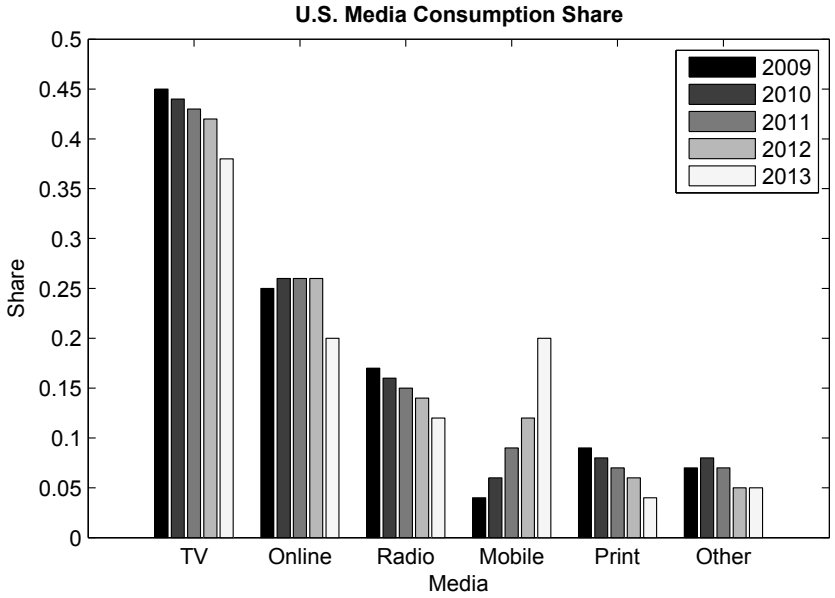


Figure 9.1: Media consumption share in the United States (from Danova (2014)).

together and combined. It is insufficient to consider studying only scientific articles, as these are published with several months' delay since the first observations were made and usually describe laboratory experiments performed under controlled conditions to test only specific aspects of larger problems. An integration with social media, such as Facebook and Twitter, is mandatory, since these environments are able to provide the most up-to-date situation in the real world. Users can supply first-hand information regarding their health status, primary or adverse effects of medication they are taking, public loci of infection etc. This step is important especially in the context of the recent steep increase in the mobile share of media consumption, as shown in Figure 9.1, which is likely to continue in the following years (Danova, 2014). More than 30% of the current usage is dedicated to social media, and this has doubled over the past two years (Danova, 2014). Analysing social media provides the most current view on the state of affairs of larger populations, having the capability of showing their health and disease situation. Nevertheless, one needs to bear in mind that reliability becomes an important issue that needs to be taken into account.

However, most effects, whether they are diseases or even death, are not caused by

a single cause, but by a chain or, in most cases, a web of many causal components. Take, for example, still incurable diseases such as cancer, for which a single cause does not exist. More specifically, in the case of pulmonary cancer, although smoking plays an important role, the disease cannot be attributed just to this factor. Thus, the interlinking of the various sources to analyse causal relations will eventually lead to the automatic creation of complex causal networks with various degrees of granularity. These networks can explain, to a certain degree or granularity, the aspects of everyday life. At a high, abstract level, the networks are addressed mostly to the general public, to advocate for both personal measures, like diet changing, and corporate measures, such as the of taxation of junk food and banning its advertising. At a low, molecular level, causal networks are mostly useful for research performed in biochemistry, molecular biology, epigenetics etc. Molecular and signalling pathways can be created and curated automatically, and linked to supporting evidence in the literature.

The vast amount of literature that is available proves to be a major impediment in the advancement of knowledge. One possible mitigating solution is the automatic production of summaries from documents, containing only the causal relations most relevant to a certain query. Although automatic summarisation is a well-studied field in NLP, focussing on specific discourse relations, including causality, has not been studied to the same extent. The causality extraction framework that has been described in this work can be easily integrated into a method for creating automatic summaries from journal articles. This can be further extended to create multi-document summaries too on the basis of causal networks, enhanced with the source dimension of meta-knowledge. The addition of other meta-knowledge information, such as polarity and certainty, can lead to an easier identification of contradictions and creation of hypotheses.

Bibliography

Sophia Ananiadou and John McNaught, editors. *Text Mining for Biology And Biomedicine*. Artech House, Inc., 2006. ISBN 158053984X.

Sophia Ananiadou, Sampo Pyysalo, Jun'ichi Tsujii, and Douglas B. Kell. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology*, 28(7):381–390, 2010. doi: 10.1016/j.tibtech.2010.04.005.

Cecilia Arighi, Zhiyong Lu, Martin Krallinger, Kevin Cohen, John Wilbur, Alfonso Valencia, Lynette Hirschman, and Cathy Wu. Overview of the BioCreative III Workshop. *BMC Bioinformatics*, 12(Suppl 8):S1, 2011. doi: 10.1186/1471-2105-12-S8-S1.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature genetics*, 25:25–29, 2000.

Chris Barker and Geoffrey K. Pullum. A theory of command relations. *Linguistics and Philosophy*, 13(1):1–34, 1990.

Riza Theresa B. Batista-Navarro and Sophia Ananiadou. Building a coreference-annotated corpus from the domain of biochemistry. In *Proceedings of BioNLP 2011*, pages 83–91, 2011.

Steven Bethard, William Corvey, Sara Klingenstein, and James H. Martin. Building a corpus of temporal-causal structure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 908–915. European Language Resources Association, 2008. ISBN 2-9517408-4-0.

Douglas Biber. *Variation across speech and writing*. Cambridge University Press, Cambridge, 1988.

Eduardo Blanco, Nuria Castell, and Dan Moldovan. Causal relation extraction. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*, Marrakech, Morocco, 2008. European Language Resources Association (ELRA). ISBN 2-9517408-4-0.

Olivier Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl 1):D267–D270, 2004. doi: 10.1093/nar/gkh061.

Austin Bradford-Hill. The environment and disease: association or causation? *Proceedings of the Royal Society of Medicine*, 58:295–300, 1965.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Ted Briscoe, John Carroll, and Rebecca Watson. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 77–80, 2006.

Evelyn Camon, Michele Magrane, Daniel Barrell, Vivian Lee, Emily Dimmer, John Maslen, David Binns, Nicola Harte, Rodrigo Lopez, and Rolf Apweiler. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Research*, 32(suppl 1):D262–266, 2004. doi: 10.1093/nar/gkh021.

Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, pages 1–10, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1118078.1118083.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Du-Seong Chang and Key-Sun Choi. Causal relation extraction using cue phrase and lexical pair probabilities. In Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee, and Oi Yee Kwong, editors, *IJCNLP*, volume 3248 of *Lecture Notes in Computer Science*, pages 61–70. Springer, 2004. ISBN 3-540-24475-1.

Du-Seong Chang and Key-Sun Choi. Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. *Information Processing & Management*, 42(3):662 – 678, 2006. doi: http://dx.doi.org/10.1016/j.ipm.2005.04.004.

Wei Chen, Gregory Aist, and Jack Mostow. Generating questions automatically from

informational text. In *Proceedings of the 2nd Workshop on Question Generation*, 2009.

Kevin Bretonnel Cohen and Lawrence Hunter. Getting started in text mining. *PLoS Computational Biology*, 4(1):e20, 2008. doi: 10.1371/journal.pcbi.0040020.

Kevin Bretonnel Cohen, Helen Johnson, Karin Verspoor, Christophe Roeder, and Lawrence Hunter. The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, 11(1):492, 2010.

Peter Corbett and Ann Copestake. Cascaded classifiers for confidence-based chemical named entity recognition. *BMC Bioinformatics*, 9(Suppl 11):S4, 2008.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1023/A:1022627411411.

Tony Danova. The mobile revolution is the biggest tech shift in years, and companies are in a race to keep up. *Business Insider*, 2014. URL http://www.businessinsider.com/mobile-media-consumption-grows-2-2014-1.

Anita de Waard, Simon Buckingham Shum, Annamaria Carusi, Jack Park, Matthias Samwald, and Ágnes Sándor. Hypotheses, evidence and relationships: The hyper approach for representing scientific knowledge claims. In *Proceedings 8th International Semantic Web Conference, Workshop on Semantic Web Applications in Scientific Discourse*, Lecture Notes in Computer Science, Berlin, 2009. Springer Verlag.

Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Attribution and the (non-) alignment of syntactic and discourse arguments of connectives. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 29–36. Association for Computational Linguistics, 2005.

Quang Xuan Do, Yee Seng Chan, and Dan Roth. Minimally supervised event causality identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 294–303, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

Robert Elwell and Jason Baldridge. Discourse connective argument identification with connective specific rankers. In *IEEE International Conference on Semantic Computing 2008*, pages 198–205. IEEE, 2008.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998. ISBN 978-0-262-06197-1.

Christiane Fellbaum, Udo Hahn, and Barry Smith. Towards new information resources for public health - from WordNet to MedicalWordNet. *Journal of Biomedical Informatics*, 39(3):321–332, 2006. doi: 10.1016/j.jbi.2005.09.004.

Joseph L. Fleiss. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, New York, 1981.

Carol Friedman, Pauline Kra, and Andrey Rzhetsky. Two biomedical sublanguages: a description based on the theories of Zellig Harris. *Journal of Biomedical Informatics*, 35(4):222–235, 2002.

Ken-ichiro Fukuda, Tatsuhiko Tsunoda, Atsushi Tamura, and Toshihisa Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 707–718, 1998.

Akira Funahashi, Yukiko Matsuoka, Akiya Jouraku, Mineo Morohashi, Norihiro Kikuchi, and Hiroaki Kitano. CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, 96(8):1254–1265, 2008.

Samik Ghosh, Yukiko Matsuoka, Yoshiyuki Asai, Kun-Yi Hsin, and Hiroaki Kitano. Software for systems biology: from tools to integrated platforms. *Nature Reviews Genetics*, 2011a.

Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. Shallow discourse parsing with conditional random fields. pages 1071–1079, 2011b.

Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. Improving the recall of a discourse parser by constraint-based postprocessing. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Roxana Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering - Volume 12*, MultiSumQA '03, pages 76–83, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

Roxana Girju and Dan Moldovan. Mining answers for causation questions. In *AAAI symposium on mining answers from texts and knowledge bases*, pages 15–25, 2002.

Gene H. Golub and Charles F. van Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences) (3rd Edition)*. The Johns Hopkins University Press, 3rd edition, 1996. ISBN 0801854148.

Art Graesser, José Otero, Albert Corbett, Dan Flickinger, Aravind Joshi, and Lucy Vanderwende. Guidelines for question generation shared task and evaluation campaigns. In *The Question Generation Task and Evaluation Challenge*, Memphis, TN, 2009. Institute for Intelligent Systems. ISBN 978-0-615-27428-7.

Ralph Grishman. Adaptive information extraction and sublanguage analysis. In *Proceedings of Workshop on Adaptive Text Extraction and Mining at Seventeenth International Joint Conference on Artificial Intelligence*, pages 1–4, 2001.

Cécile Grivaz. Human judgements on causation in French texts. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2626–2631. European Language Resources Association, 2010. ISBN 2-9517408-6-7.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, 2009. doi: 10.1145/1656274.1656278.

Zellig Harris. *Mathematical Structures of Language*. John Wiley and Son, New York, 1968.

Michael Heilman and Noah A. Smith. Question generation via overgenerating transformations and ranking. Technical report, Language TEchnologies Institute, Carnegie Mellon University, 2009.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 399–409. Association for Computational Linguistics, 2010.

Jerry R. Hobbs. *Literature and Cognition*. CSLI lecture notes. Center for the Study of Language and Information, 1990. ISBN 9780937073520.

George Hripcsak and Adam S. Rothschild. Agreement, the f-measure, and reliability

in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005. doi: 10.1197/jamia.M1733.

Michael Hucka, Andrew Finney, Herbert M. Sauro, Hamid Bolouri, John C. Doyle, and Hiroaki Kitano et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4): 524–531, 2003. doi: 10.1093/bioinformatics/btg015.

David Hume and Lewis Amherst Selby-Bigge. *A treatise of human nature*. Clarendon Press, Oxford, 1896.

Syeed Ibn Faiz and Robert E. Mercer. Identifying explicit discourse connectives in text. In *Advances in Artificial Intelligence*, volume 7884 of *Lecture Notes in Computer Science*, pages 64–76. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38456-1.

David Jessop, Sam Adams, Egon Willighagen, Lezan Hawizy, and Peter Murray-Rust. OSCAR4: a flexible architecture for chemical text-mining. *Journal of Cheminformatics*, 3(1):41, 2011. doi: 10.1186/1758-2946-3-41.

Jing Jiang. A Literature Survey on Domain Adaptation of Statistical Classifiers, 2008. URL http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/da_survey.pdf.

Leo Joskowsicz, Tomasz Ksiezyk, and Ralph Grishman. Deep domain models for discourse analysis. In *The Annual AI Systems in Government Conference*, pages 195–200. IEEE Computer Society, 1989.

Yoshinobu Kano, William Baumgartner, Luke McCrohon, Sophia Ananiadou, Kevin Bretonnel Cohen, Larry Hunter, and Jun'ichi Tsujii. U-Compare: share and compare text mining tools with UIMA. *Bioinformatics*, 25(15):1997–1998, 2009.

Sophia Katrenko and Pieter Adriaans. Learning relations from biomedical corpora using dependency trees. In *Knowledge Discovery and Emergent Complexity in*

Bioinformatics, volume 4366 of *Lecture Notes in Computer Science*, pages 61–80. Springer Berlin / Heidelberg, 2007.

Brian Kemper, Takuya Matsuzaki, Yukiko Matsuoka, Yoshimasa Tsuruoka, Hiroaki Kitano, Sophia Ananiadou, and Jun'ichi Tsujii. PathText: a text mining integrator for biological pathway visualizations. *Bioinformatics*, 26(12):i374–i381, 2010.

Halil Kilicoglu and Sabine Bergler. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9 (Suppl 11):S10, 2008. doi: 10.1186/1471-2105-9-S11-S10.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10, 2008.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. Overview of BioNLP Shared Task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

Samantha Kleinberg and George Hripcsak. A review of causal inference for biomedical informatics. *Journal of Biomedical Informatics*, 44(6):1102 – 1112, 2011. doi: 10.1016/j.jbi.2011.07.001.

Alistair Knott and Ted J.M. Sanders. The classification of coherence relations and their lingusitic markers: an exploration of two languages. *Journal of Pragmatics*, 30:135–175, 1998.

Varada Kolhatkar, Heike Zinsmeister, and Graeme Hirst. Annotating anaphoric shell nouns with their antecedents. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 112–121, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

BalaKrishna Kolluru, Lezan Hawizy, Peter Murray-Rust, Jun'ichi Tsujii, and Sophia Ananiadou. Using workflows to explore and optimise named entity recognition for chemistry. *PLoS ONE*, 6(5):e20181, 2011. doi: doi:10.1371/journal.pone.0020181.

Georgios Kontonatsios, Ioannis Korkontzelos, BalaKrishna Kolluru, Paul Thompson, and Sophia Ananiadou. Deploying and sharing U-Compare workflows as web services. *Journal of Biomedical Semantics*, 4:7, 2013. doi: 10.1186/2041-1480-4-7.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, and Lyle Ungar. 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*, pages 61–68, 2004.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

Ronald W. Langacker. *On Pronominalization and the Chain of Command*. San Diego, 1966.

David Lewis. *Counterfactuals*. Blackwell Publishers, Malden, Mass, 2001. ISBN 0631224254.

Maria Liakata and Larisa Soldatova. ART corpus, 2009. http://hdl.handle.net/2160/1979.

Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. Corpora for the conceptualisation and zoning of scientific papers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, FirstView:1–34, 2012. doi: 10.1017/S1351324912000307.

Thomas Lippincott, Diarmuid Seaghdha, and Anna Korhonen. Exploring subdomain variation in biomedical language. *BMC Bioinformatics*, 12(1):212, 2011. doi: 10.1186/1471-2105-12-212.

Boon-Toh Low, Ki Chan, Lei-Lei Choi, Man-Yee Chin, and Sin-Ling Lay. Semantic expectation-based causation knowledge extraction: A study on hong kong stock movement analysis. In *Advances in Knowledge Discovery and Data Mining*, volume 2035 of *Lecture Notes in Computer Science*, pages 114–123. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-41910-5. doi: 10.1007/3-540-45357-1_15.

Beatriz Maeireizo, Diane Litman, and Rebecca Hwa. Co-training for predicting emotions with spoken dialogue data. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 28. Association for Computational Linguistics, 2004.

William C. Mann and Sandra A. Thompson. Rhetorical structure theory: towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA, 2000. ISBN 0262133725.

Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375. Association for Computer Linguistics, 2002.

Huaiyu Mi and Paul Thomas. PANTHER pathway: an ontology-based pathway database coupled with data analysis tools. *Methods Mol. Biol*, 563:123–140, 2009.

Claudiu Mihăilă and Sophia Ananiadou. What causes a causal relation? detecting causal triggers in biomedical scientific discourse. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 38–45. Association for Computational Linguistics, 2013a.

Claudiu Mihăilă and Sophia Ananiadou. Recognising discourse causality triggers in the biomedical domain. *Journal of Bioinformatics and Computational Biology*, 11 (6):1343008, 2013b. doi: 10.1142/S0219720013430087.

Claudiu Mihăilă and Sophia Ananiadou. A hybrid approach to recognising discourse causality in the biomedical domain. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2013*, pages 361–366. IEEE, 2013c. ISBN 978-1-4799-1309-1.

Claudiu Mihăilă and Sophia Ananiadou. The meta-knowledge of causality in biomedical scientific discourse. In *Proceedings of the 9th Conference on Language Resources and Evaluation*, pages 1984–1991. ELRA, 2014.

Claudiu Mihăilă and Sophia Ananiadou. Semi-supervised learning of causal relations in biomedical scientific discourse. *BMC Medical Informatics and Decision Making*, In press.

Claudiu Mihăilă and Riza Theresa Batista-Navarro. What's in a name? Entity type variation across two biomedical subdomains. In *EACL*, pages 38–45. The Association for Computer Linguistics, 2012. ISBN 978-1-937284-19-0.

Claudiu Mihăilă, Riza Theresa Batista-Navarro, and Sophia Ananiadou. Analysing entity type variation across biomedical subdomains. In *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2012)*, pages 1–7, 2012.

Claudiu Mihăilă, Tomoko Ohta, Sampo Pyysalo, and Sophia Ananiadou. BioCause: Annotating and analysing causality in the biomedical domain. *BMC Bioinformatics*, 14(1):2, 2013.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. Protein-protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, 78(12):e39–e46, 2009. doi: 10.1016/j.ijmedinf.2009.04.010.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun'ichi Tsujii. Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology*, 8(1):131–146, 2010.

Makoto Miwa, Paul Thompson, and Sophia Ananiadou. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics*, 28(13):1759–1765, 2012a. doi: 10.1093/bioinformatics/bts237.

Makoto Miwa, Paul Thompson, John McNaught, Douglas B. Kell, and Sophia Ananiadou. Extracting semantically enriched events from biomedical literature. *BMC Bioinformatics*, 13:108, 2012b. doi: 10.1186/1471-2105-13-108.

Yusuke Miyao and Jun'ichi Tsujii. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80, 2008.

Yusuke Miyao, Tomoko Ohta, Katsuya Masuda, Yoshimasa Tsuruoka, Kazuhiro Yoshida, Takashi Ninomiya, and Jun'ichi Tsujii. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In *ACL*, 2006.

Jack Mostow and Wei Chen. Generating instruction automatically for the reading strategy of self-questioning. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education*, pages 465–472, Amsterdam, The Netherlands, 2009. IOC Press.

Rutu Mulkar-Mehta, Andrew S. Gordon, Jerry R. Hobbs, and Eduard Hovy. Causal markers across domains and genres of discourse. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 183–184, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0396-5.

Raheel Nawaz, Paul Thompson, and Sophia Ananiadou. Negated bioevents: Analysis and identification. *BMC Bioinformatics*, 14(1):14, 2013a. doi: doi:10.1186/1471-2105-14-14.

Raheel Nawaz, Paul Thompson, and Sophia Ananiadou. Something old, something new: identifying knowledge source in bio-events. In *International Journal of Computational Linguistics and Applications*, pages 129–144, 2013b.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum, editors. *Proceedings of the BioNLP Shared Task 2013 Workshop*. Association for Computational Linguistics, Sofia, Bulgaria, 2013a.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. Overview of BioNLP Shared Task 2013. In

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 1–7, Sofia, Bulgaria, 2013b. Association for Computational Linguistics.

Chikashi Nobata, Yutaka Sasaki, Naoaki Okazaki, C.J. Rupp, Jun'ichi Tsujii, and Sophia Ananiadou. Semantic search on digital document repositories based on text mining results. In *International Conferences on Digital Libraries and the Semantic Web 2009 (ICSD2009)*, pages 34–48, 2009.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. From pathways to biomolecular events: Opportunities and challenges. In *Proceedings of BioNLP 2011 Workshop*, pages 105–113, Portland, Oregon, USA, 2011a. Association for Computational Linguistics.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. Overview of the epigenetics and post-translational modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 16–25, 2011b.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

Emily Pitler and Ani Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *ACL/AFNLP (Short Papers)*, pages 13–16, 2009.

Michael Poprat, Elena Beisswanger, and Udo Hahn. Building a BioWordNet by using WordNet's data formats and WordNet's software infrastructure: a failure story. In *SETQA-NLP '08: Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 31–39, Morristown, NJ, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-10-7.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the*

Sixth International Conference on Language Resources and Evaluation, Marrakech, Morocco, 2008. European Language Resources Association. ISBN 2-9517408-4-0.

Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Exploiting scope for shallow discourse parsing. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. The biomedical discourse relation bank. *BMC Bioinformatics*, 12(1):188, 2011.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50, 2007.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the infectious diseases (ID) task of BioNLP shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 26–35, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

Rafal Rak, Andrew Rowley, William J. Black, and Sophia Ananiadou. Argo: an integrative, interactive, text mining-based workbench supporting curation. *Database: The Journal of Biological Databases and Curation*, 2012, 2012.

Polepalli Balaji Ramesh, Rashmi Prasad, Tim Miller, Brian Harrington, and Hong Yu. Automatic discourse connective detection in biomedical text. *Journal of the*

American Medical Informatics Association, 19(5):800–808, 2012. doi: 10.1136/amiajnl-2011-000775.

Tanya M. Reinhart. *The Syntactic Domain of Anaphora*. PhD thesis, Massachusetts Institute of Technology, 1976.

Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics, 2003.

Miguel Angel Ríos Gaona, Alexander Gelbukh, and Sivaji Bandyopadhyay. Recognizing textual entailment using a machine learning approach. In *Advances in Soft Computing*, volume 6438 of *Lecture Notes in Computer Science*, pages 177–185. Springer Berlin / Heidelberg, 2010.

Vasile Rus and Art Graesser. *The Question Generation Task and Evaluation Challenge*. Institute for Intelligent Systems, Memphis, TN, 2009. ISBN 978-0-615-27428-7.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. The first question generation shared task evaluation challenge. In *Proceedings of the Sixth International Natural Language Generation Conference (INLG 2010)*, 2010.

Andrey Rzhetsky, Ivan Iossifov, Tomohiro Koike, Michael Krauthammer, Pauline Kra, Mitzi Morris, Hong Yu, Ariel Pablo Duboué, Wubin Weng, W.John Wilbur, Vasileios Hatzivassiloglou, and Carol Friedman. GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *Journal of Biomedical Informatics*, 37(1):43 – 53, 2004. doi: 10.1016/j.jbi.2003.10.001.

Rune Sætre, Kazuhiro Yoshida, Akane Yakushiji, Yusuke Miyao, Yuichiro Matsubayashi, and Tomoko Ohta. AKANE system: Protein-protein interaction pairs in the

BioCreAtIvE2 Challenge, PPI-IPS subtask. In *Second BioCreative Challenge Evaluation Workshop*, 2007.

Naomi Sager, Carol Friedman, and Margaret Lyman. *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley, Reading, MA, 1987.

Ágnes Sándor. Modeling metadiscourse conveying the author's rhetorical strategy in biomedical research abstracts. *Revue Française de Linguistique Appliquée*, 200(2): 97–109, 2007.

Ágnes Sándor and Anita de Waard. Identifying claimed knowledge updates in biomedical research articles. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, ACL '12, pages 10–17, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. How to make the most of NE dictionaries in statistical NER. *BMC Bioinformatics*, 9(Suppl 11):S5, 2008.

Guergana K. Savova, Wendy W. Chapman, Jiaping Zheng, and Rebecca S. Crowley. Anaphoric relations in the clinical narrative: corpus creation. *Journal of the American Medical Informatics Association*, 18(4):459–465, 2011. doi: 10.1136/amiajnl-2011-000108.

Deborah Schiffrin. *Discourse Markers*. Studies in Interactional Sociolinguistics. Cambridge University Press, 1988. ISBN 9780521357180.

Martijn J. Schuemie, M. Weeber, Bob J. A. Schijvenaars, Erik M. van Mulligen, C. C. van der Eijk, R. Jelier, B. Mons, and Jan A. Kors. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, 20(16):2597–2604, 2004. doi: 10.1093/bioinformatics/bth291.

Lee Schwartz, Takako Aikawa, and Michel Pahud. Dynamic language learning tools. In *Proceedings of the InSTIL/ICALL Symposium*, 2004.

Burr Settles. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

Parantu Shah, Carolina Perez-Iratxeta, Peer Bork, and Miguel Andrade. Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, 4(1):20, 2003. doi: 10.1186/1471-2105-4-20.

Ernest Sosa. *Causation and conditionals*. Oxford readings in philosophy. Oxford University Press, 1975. ISBN 9780198750307.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

Evgeny A. Stepanov and Giuseppe Riccardi. Comparative evaluation of argument extraction algorithms in discourse relation parsing. In *13th International Conference on Parsing Technologies (IWPT 2013)*, pages 36–44, 2013.

Peter D. Stetson, Stephen B. Johnson, Matthew Scotch, and George Hripcsak. The sublanguage of cross-coverage. *Proceedings of the AMIA Symposium*, pages 742–746, 2002.

Rajen Subba and Barbara Di Eugenio. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009*

Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 566–574. Association for Computational Linguistics, 2009.

Lorraine Tanabe, Natalie Xie, Lynne Thom, Wayne Matten, and W. John Wilbur. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S3, 2005. doi: 10.1186/1471-2105-6-S1-S3.

Paul Thompson, Syed Iqbal, John McNaught, and Sophia Ananiadou. Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinformatics*, 10(1):349, 2009.

Paul Thompson, John McNaught, Simonetta Montemagni, Nicoletta Calzolari, Riccardo del Gratta, Vivian Lee, Simone Marchi, Monica Monachini, Piotr Pezik, Valeria Quochi, C.J. Rupp, Yutaka Sasaki, Giulia Venturi, Dietrich Rebholz-Schuhmann, and Sophia Ananiadou. The BioLexicon: a large-scale terminological resource for biomedical text mining. *BMC Bioinformatics*, 12:397, 2011a. doi: 10.1186/1471-2105-12-397.

Paul Thompson, Raheel Nawaz, John McNaught, and Sophia Ananiadou. Enriching a biomedical event corpus with meta-knowledge annotation. *BMC Bioinformatics*, 12:393, 2011b. doi: 10.1186/1471-2105-12-393.

Jun'ichi Tsujii, editor. *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*. Association for Computational Linguistics, Boulder, Colorado, USA, 2009.

Jun'ichi Tsujii, Jin-Dong Kim, and Sampo Pyysalo, editors. *Proceedings of BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, Portland, Oregon, USA, 2011.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics - 10th Panhellenic Conference on Informatics*, volume 3746 of *LNCS*, pages 382–392. Springer-Verlag, Volos, Greece, 2005.

Yoshimasa Tsuruoka, Makoto Miwa, Kaisei Hamamoto, Jun'ichi Tsujii, and Sophia Ananiadou. Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics*, 27(13):i111–i119, 2011. doi: 10.1093/bioinformatics/btr214.

Karin Verspoor, Kevin B. Cohen, Ben Goertzel, and Inderjeet Mani. Introduction to BioNLP'06. In Karin Verspoor, Kevin B. Cohen, Ben Goertzel, and Inderjeet Mani, editors, *BioNLP '06: Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pages iii–iv, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Karin Verspoor, Kevin Bretonnel Cohen, and Lawrence Hunter. The textual characteristics of traditional and open access scientific journals are similar. *BMC Bioinformatics*, 10(1):183, 2009. doi: 10.1186/1471-2105-10-183.

Veronika Vincze, Gyorgy Szarvas, Richard Farkas, Gyorgy Mora, and Janos Csirik. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11):S9, 2008. doi: 10.1186/1471-2105-9-S11-S9.

Tuangthong Wattarujeekrit, Parantu Shah, and Nigel Collier. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, 5(1):155, 2004.

Ben Wellner. *Sequence Models and Ranking Methods for Discourse Parsing*. PhD thesis, Brandeis University, 2009.

Ben Wellner and James Pustejovsky. Automatically identifying the arguments of discourse connectives. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 92–101, Prague, Czech Republic, 2007. Association for Computational Linguistics.

Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, 2005.

Brendan Wyse and Paul Piwek. Generating questions from openlearn study units. In *Proceedings of the 2nd Workshop on Question Generation*, 2009.

Fan Xu, Qiaoming Zhu, and Guodong Zhou. A unified framework for discourse argument identification via shallow semantic parsing. In *COLING (Posters)*, pages 1331–1340, 2012.

Xuchen Yao, Gosse Bouma, and Yi Zhang. Semantics-based question generation and implementation. *Dialogue & Discourse*, 3(2):11–42, 2012.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.

Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B. Cohen. Frontiers of biomedical text mining: current progress. *Briefings in Bioinformatics*, 8(5): 358–375, 2007. doi: 10.1093/bib/bbm045.

Appendix A

BioCause annotation guidelines

These guidelines have been developed to define the scope of annotation with regard to causal relationships that exist in biomedical scientific discourse. The task is described in the following sections.

A.1 Pre-annotated named entities and events

The articles that are subject to our task have been previously used for biomedical text mining purposes. Thus, they already contain some annotations which are manually added by domain experts. These include biomedical named entity and event information.

Figure A.1 shows an example of such annotations.

A.2 Recognising causal relations

This section will define the concept of *causality*, as well as exemplifying the annotation process.

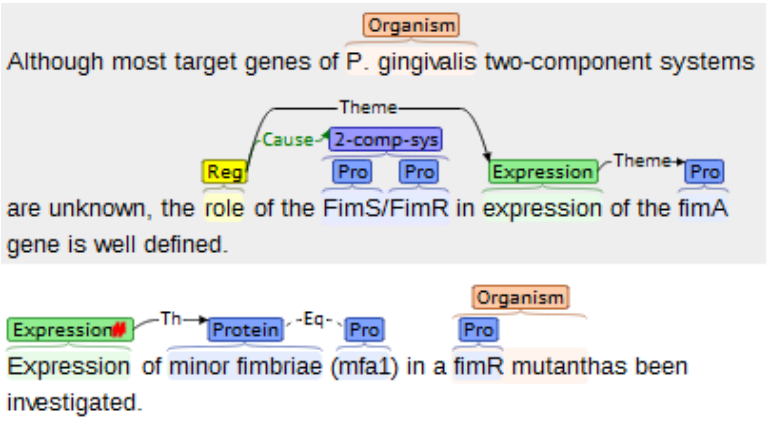


Figure A.1: Annotations already included in the BioCause corpus.

A.2.1 Definition of causality

Causality is a vague term whose definition varies depending on the domain and sometimes within domains. Thus, we cannot provide a strict formulation of what causality is. Instead, we offer several guidelines or tests that can help in your decisions.

One essential condition of causality is the temporal asymmetry between the cause and the effect. More specifically, the cause must always precede the effect.

Furthermore, it is important to understand whether the effect would still have happened had the cause not occurred in the first place. If such is the case, then it is possible that the relation that exists between these two statements is not causality.

If necessary, please try rewording the sentences and performing linguistic tests (e.g., placing an explicit causal marker and deciding whether the resulting text is an accurate rephrasing of the original).

A.2.2 Annotating causal triggers

The first step towards annotating causal relation is to identify the causal triggers. When a new causal relation is found, identify what expression in the text leads to you the conclusion that the two predicates are in a causal relation.

For instance, example (A.1) contains an instance of an annotated causal trigger.

(A.1) Here it is demonstrated that FimR binds directly to the promoter region of the mfa1 gene, *suggesting* a direct role of FimR in activation of mfa1 expression.

Causal trigger can be expressed in numerous ways in text, ranging from single words, such as conjunctions and adverbials (e.g., *because*, *thus*) to full verb phrases (e.g., *these results suggest that*).

There will be cases where a causal trigger is not explicitly given in text, but suggested by authors. In this case, please annotate the trigger as an empty span of text in the location where you consider that it should be placed. Example (A.2) shows how an implicit trigger should be annotated.

(A.2) Mlc repressed hilE in a direct manner {} by binding to two distinct sites in the hilE P3 promoter region.

A.2.3 Annotating causal arguments

After annotating the causal triggers, it is necessary to mark up its two arguments. The arguments can be located at any place in the text. Either they are both in the same sentence with the trigger, or one of them is in a different sentence. One argument must always be placed in the same sentence with the trigger.

An example of same sentence arguments is provided below, in sentence (A.3), whilst separate triggers are provided in example (A.4).

(A.3) Here it is demonstrated that [FimR binds directly to the promoter region of the mfa1 gene]$_{Cause}$, *suggesting* [a direct role of FimR in activation of mfa1 expression]$_{Efect}$.

(A.4) [Y. pseudotuberculosis, in contrast to Y. pestis, has been shown to be orally toxic to flea]$_{Cause}$.
This suggests that loss of one or more insect gut toxins is a critical step in the change of the Y. pestis lifestyle compared with the Y. pseudotuberculosis and thus in evolution of flea-borne transmission]$_{Efect}$.

A.3 Other items

This section provides further directions that are not entirely relevant to the annotation procedure, such as actions to be taken in case of mistyped words, grammatical errors, and concerns.

A.3.1 Spelling or grammatical errors

Ignore any spelling mistakes or grammatical errors that you encounter whilst reading the texts. Take, for instance, example (A.5), where the word *through* has been mistyped as *throug*. As these are published articles, they must be taken as is, and must not suffer any alteration. Editing the text is forbidden and has been disabled for your convenience.

(A.5) It is hypothesized that the FimS/FimR system regulates expression of each fimbrial gene **throug** a unique mechanism.

A.3.2 Points for discussion

If at any point during your annotation you come across something that you wish to discuss with other annotators or task creator, please mark-up the relevant part of text and use the NOTES area of your annotation dialogue to describe your concern. Figure A.2 shows the location of the NOTES area in the annotation dialogue.

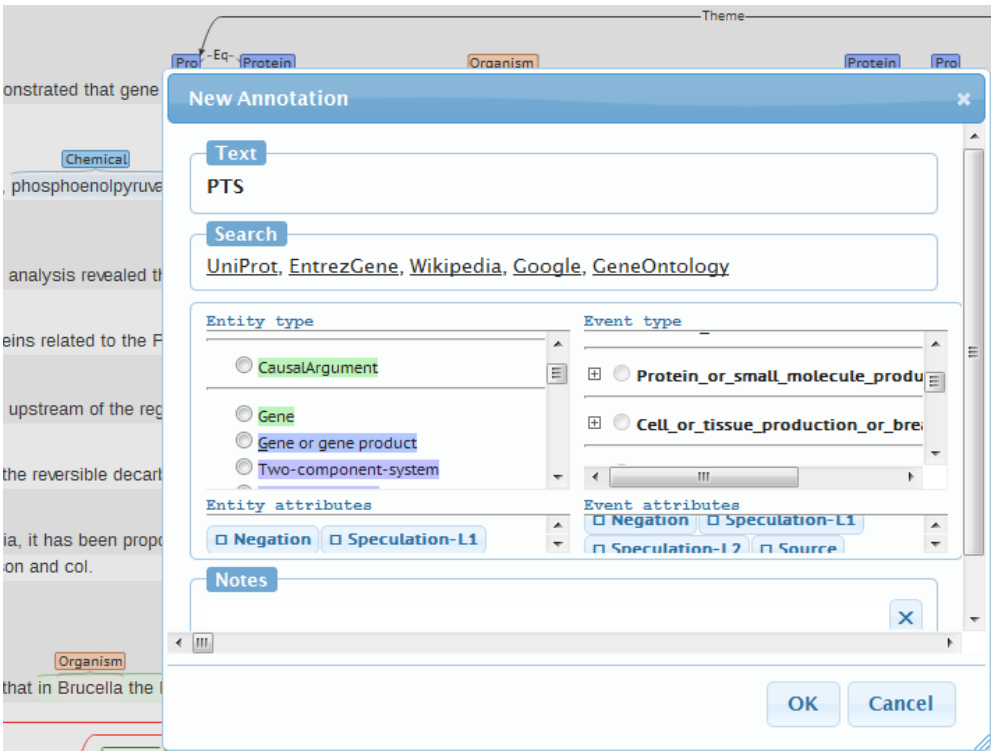


Figure A.2: NOTES area in the annotation dialogue

Appendix B

BioCause+MK annotation guidelines

These guidelines have been developed to define the scope of annotation with regard to the meta-knowledge of causal relationships that exist in biomedical scientific discourse. The task is described in the following sections.

B.1 Pre-annotated named entities, events, and causal relations

The documents to undergo annotation in this task have already been annotated with biomedical named entities, events and discourse causal relations. Figure B.1 shows an example of such annotations.

Even if mistakes are discovered during the annotation of meta-knowledge information, they are to be ignored. These are out of scope for the present task.

B.2 Meta-knowledge

The meta-knowledge of discourse causal relations that we are interested in has four dimensions: polarity, certainty, source and knowledge type. All four dimensions must be

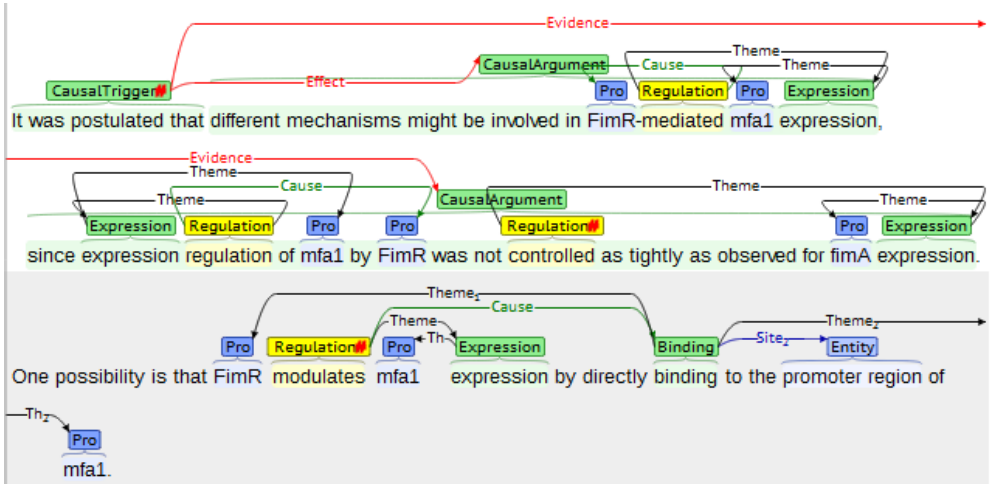


Figure B.1: Annotations already included in the BioCause corpus.

annotated for each causal relation that exists in the BioCause corpus. For the comfort of annotators, each dimension has a default value assigned to it, which is the majority value that we expect. Thus, annotators must change the values of only some dimensions for only a subset of causal relations. All four dimensions are defined in what follows, and examples are given for each value.

Polarity

This dimension identifies the truth value of the asserted causal association. A negated causal association is defined as one describing the non-existence or absence of a causal link between two spans of text. The recognition of such associations is vital, as it can lead to the correct interpretation of a causal association, completely opposite to that of a non-negated one.

- *positive*: no explicit negation of the causality. This is the default category, as most causal associations are expected to be positive.

- *negative*: the association has been negated according to the description above. The negation may be indicated through lexical clues such as *no*, *not* or *fail*.

Certainty

This dimension encodes the confidence or certainty level ascribed to the association in the given text. The epistemic scale is partitioned into three distinct levels:

- *L1*: explicit indication of either low confidence or considerable speculation towards the association or the association occurs infrequently or only some of the time.

- *L2*: explicit indication of either high (but not complete) confidence or slight speculation towards the association or the association occurs frequently, but not all of the time.

- *L3*: the default category. No explicit expression that either there is uncertainty or speculation towards the associations or that the association does not occur all of the time.

Source

The source of the knowledge expressed by the causal association is encoded as:

- *current*: the association makes an assertion that can be attributed to the current study. This is the default category, and is assigned in the absence of explicit lexical or contextual clues, although explicit clues such as *the present study* may be encountered.

- *other*: the association is attributed to a previous study. Explicit clues are usually present either as citations, or by using words such as *previously* and *recent studies*.

Knowledge type

The *Knowledge Type* (*KT*) captures the general information about the content of the causal association, classifying it into five categories:

- *analysis*: inferences, interpretations, speculations or other types of cognitive analysis, always accompanied by lexical clues, typical examples of which include *suggest*, *indicate*, *therefore* and *conclude*.

- *fact*: events that describe general facts and well-established knowledge, and sometimes accompanied by lexical clues such as *known*.

- *investigation*: enquiries or investigations, which have either already been conducted or are planned for the future, typically accompanied by lexical clues like *examined*, *investigated* and *studied*.

- *observation*: direct observations, sometimes represented by lexical clues like *found*, *observed* and *report*, etc.

- *other*: the default category, assigned to associations that either do not fit into one of the above categories, do not express complete information, or whose *KT* is unclear or is unassignable from the context.

B.3 Other items

This section provides further directions that are not entirely relevant to the annotation procedure, such as actions to be taken in case of mistyped words, grammatical errors, and concerns.

B.3.1 Spelling or grammatical errors

Ignore any spelling mistakes or grammatical errors that you encounter whilst reading the texts. Take, for instance, example (B.1), where the word *through* has been mistyped as *throug*. As these are published articles, they must be taken as is, and must not suffer any alteration. Editing the text is forbidden and has been disabled for your convenience.

(B.1) It is hypothesized that the FimS/FimR system regulates expression of each fimbrial gene **throug** a unique mechanism.

B.3.2 Points for discussion

If at any point during your annotation you come across something that you wish to discuss with other annotators or task creator, please mark-up the relevant part of text and use the `NOTES` area of your annotation dialogue to describe your concern. Figure B.2 shows the location of the `NOTES` area in the annotation dialogue.

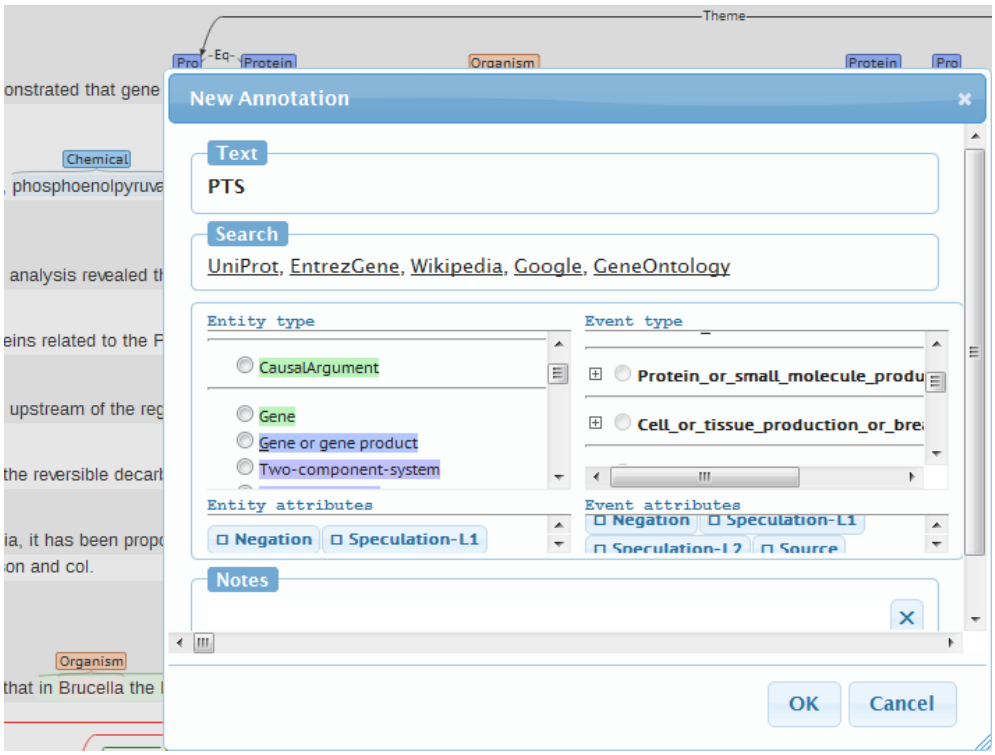


Figure B.2: NOTES area in the annotation dialogue

Appendix C

List of part-of-speech and syntactic category tags

The items included in Table C.1 represent the base forms to create syntactic categories. This is performed by adding a suffix which indicates whether a constituent is a saturated phrase (expressed by "P") or an unsaturated constituent ("X").

Tag	Description
ADJ	Adjective
ADV	Adverb
CONJ	Coordination conjunction
COOD	Part of coordination
C	Complementiser
D	Determiner
N	Noun
P	Preposition
PN	Punctuation
PRT	Particle
S	Sentence
SC	Subordination conjunction
V	Verb

Table C.1: Syntactic category tags.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Table C.2: Penn Treebank part-of-speech tags.