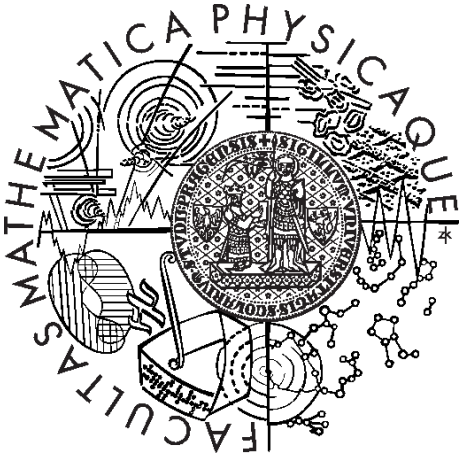
Charles University in Prague

Faculty of Mathematics and Physics

MASTER THESIS



Sara van de Moosdijk

Mining texts at the discourse level

Institute of Formal and Applied Linguistics

Supervisor of the master thesis:	RNDr. Pavel Pecina, Ph.D. Yannick Toussaint, Ph.D. Maxime Amblard, Ph.D.
Study programme:	Master of Computer Science
Specialization:	Mathematical Linguistics

Prague 2014

This research was performed as part of an internship under guidance of the ORPAILLEUR team at INRIA in Nancy (FR). I would like to thank my supervisors in France, Yannick Toussaint and Maxime Amblard, as well as my supervisor in Czech Republic, Pavel Pecina for their help and guidance. Furthermore I would like to thank Adrien Coulet for allowing me to use his implementation of the CloseByOne algorithm and helping me make changes to it. Finally, I thank my parents for all of their support during the past two years.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Beekbergen date 30 July, 2014 signature of the author

Název práce: Dobývání informací z textu na úrovni diskurzu

Autor: Sara van de Moosdijk

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Pavel Pecina, Ph.D., Yannick Toussaint, Ph.D., Maxime Amblard, Ph.D.

Abstrakt: Lingvistický diskurz se zabývá významem delších kusů textu, od vět po celé dokumenty, mohl by se však uplatnit i v úlohách získávání informací z textu, např. vyhledávání dokumentů či jejich sumarizace. Cílem této práce je uplatnění informací o stavbě diskurzu psaného textu pro potřeby získávání znalostí. Jedná se o prvnípokus, který se snaží skloubit tyto dva velice odlišné obory, a jeho ambicí je tak připravit základ pro tento způsob získávání znalostí. Náš postup spočívá v použití metod neřízeného strojového učení k analýze diskurzních vztahů a jejich následovném modelování pomocí vzorových struktur z formální konceptuální analýzy. Naši metodu jsme aplikovali na korpus lékařských článků z databáze PubMed. Tyto lékařské texty potom obohacujeme o koncepty z metathesauru UMLS, které jsou kombinovány s daty ze sémantické sítě UMLS, která fungují jako ontologie ve vzorových strukturách. Naše výsledky ukazují, že i přes vysokou úroveň šumu je naše metoda slibná a bylo by možné ji aplikovat i na jiné domény.

Klíčová slova: dobývání informací z textu, výstavba diskurzu, formální konceptuální analýza

Title: Mining texts at the discourse level

Author: Sara van de Moosdijk

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Pavel Pecina, Ph.D., Yannick Toussaint, Ph.D., Maxime Amblard, Ph.D.

Abstract: Linguistic discourse refers to the meaning of larger text segments, and could be very useful for guiding attempts at text mining such as document selection or summarization. The aim of this project is to apply discourse information to Knowledge Discovery in Databases. As far as we know, this is the first attempt at combining these two very different fields, so the goal is to create a basis for this type of knowledge extraction. We approach the problem by extracting discourse relations using unsupervised methods, and then model the data using pattern structures in Formal Concept Analysis. Our method is applied to a corpus of medical articles compiled from PubMed. This medical data can be further enhanced with concepts from the UMLS MetaThesaurus, which are combined with the UMLS Semantic Network to apply as an ontology in the pattern structures. The results show that despite having a large amount of noise, the method is promising and could be applied to domains other than the medical domain.

Keywords: text mining, discourse structure, formal concept analysis

Contents

1. Introduction

The aim of this project is to use pattern structures, which is an extension of Formal Concept Analysis, to mine discourse representation structures in medical text. We believe that applying these data mining techniques to discourse representations can aid experts in extracting new knowledge from medical documents, as well as improving document selection or summarization methods.

Text mining is the process of extracting useful information and knowledge from natural language text, and it combines several important fields of study including machine learning, statistics, pattern recognition, and information retrieval [Hotho et al., 2005]. It can be used to select certain documents from a collection, summarize them, perform clustering, or perform sentiment analysis, all based on the content of the text. Common uses of text mining can be found in varying fields, whether you look at marketing experts using sentiment analysis to determine the public opinion on a product [Melville et al., 2009], or medical experts attempting to discover new treatment options by combining links between substances, biological processes, and diseases found in separate articles [Swanson, 1990].

Most text mining applications view a document simply as a bag-of-words, from which key words can be extracted and used to guide the process of clustering, summarization, or other analysis. These methods can involve some degree of preprocessing in the form of filtering, lemmatization, stemming, part-of-speech tagging, or word sense disambiguation [Hotho et al., 2005]. Such preprocessing methods can provide extra information to guide and enhance the text mining process, but it still adheres to the bag-of-words point of view. To move beyond this viewpoint, one has to go beyond lexical, morphological, or semantic treatments of natural language into the domain of discourse.

Discourse is the study of meaning applied to phrases, sentences, or larger pieces of text. Instead of placing the focus on the meaning of an utterance, it attempts to model relationships between larger text units and how they affect each others' meaning. Looking at discourse of a natural language can generate new information which would not be directly extractable from the meaning of the individual utterances in the text. Example 1.0.1a shows a sentence from a medical article about hereditary hemorrhagic telangiectasia (HHT)[1]. This particular sentence lists some symptoms which were frequent in the population of their study, as well as some symptoms which were infrequent. If it were analyzed from a bag-of-words perspective, one could match the words of the sentence to a medical thesaurus, which could provide the additional information that epistaxis, pulmonary AVM, GI bleeding, and symptomatic liver VM are symptoms or medical procedures, and that HHT is a disease. This would lead to the conclusion that all of these terms are somehow related, but it is impossible to extract the type of the relation without taking discourse into account.

[1] `http://www.ncbi.nlm.nih.gov/pubmed/22991266`

Example 1.0.1
a. Recurrent epistaxis and pulmonary AVM were frequent in our study population, whereas HHT-related GI bleeding and symptomatic liver VM were infrequent, for all HHT genes.
b. Recurrent epistaxis and pulmonary AVM were frequent in our study population,
c. whereas HHT-related GI bleeding and symptomatic liver VM were infrequent, for all HHT genes.

When taking into account the discourse of example sentence 1.0.1a, one can arrive at the conclusion that the sentence can be split as shown in Example 1.0.1b and 1.0.1c, and that there is in fact contrast between the two parts. Part b lists the symptoms or medical procedures which are frequent, whereas part c lists the ones which are infrequent. Creating a complete discourse representation would contribute even more detailed information than just the fact that there is a contrast between the two lists, such as that both lists relate to the disease HTT, one positively and one negatively. This is relatively crucial information if one were to attempt summarizing a bunch of medical articles based on their content, since the summary would need to list the frequent symptoms and not the infrequent ones.

The relative complexity of discourse-annotated data, especially in combination with additional annotations from ontologies, means that one needs a data mining algorithm which can handle complex data and is suitable for text mining. One such data mining process is Formal Concept Analysis, which shows great promise for handling various text mining situations, including the use of ontologies and other annotations [Carpineto and Romano, 2005, Priss, 2005]. It is a mathematical theory which builds concept lattices based on sets of objects and descriptions. The inherent specialization/generalization structure of lattices means the method is naturally suitable for handling other hierarchical components like ontologies. Furthermore it can combine a variety of features, not necessarily of the same data type (e.g. some of your attributes can be numerical, others can be sets, and still others can be textual) through an extension called pattern structures. This feature should ensure that Formal Concept Analysis is flexible enough to handle textual data at a discourse level.

As far as we know, there has been no attempt at text mining which takes discourse into account, so far. Hence the aim of this project is to make a start in applying text mining to discourse data, using Formal Concept Analysis as a basis; to see if this approach to text mining shows promise, and to find the possible pitfalls. Our experiments use medical articles for textual data, since there are large sources available online such as PubMed[2], and the medical domain could benefit greatly from text mining approaches. Tools which could automatically summarize a collection of articles about a disease, or find new links between different articles leading to new knowledge, could certainly make life easier for both patients and professionals.

We start by describing discourse structures in more detail, as well as providing an explanation of Formal Concept Analysis and its features in Chapter 2. Chap-

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

ter 3 gives an overview of the data collection process, and Chapter 4 describes how the data is annotated with discourse features, including an evaluation of the annotation results. Finally the application of Formal Concept Analysis and pattern structures to the discourse-annotated data is shown in Chapter 5.

2. Background

Whenever one aims to combine two very different fields of research, in this case semantics of natural language and Knowledge Discovery in Databases (KDD), it is important to first understand each field separately. Hence this chapter aims to give a general description of each field, as well as providing references to other materials containing more detailed explanations. Section 2.1 describes the aim of formal semantics and how it can be applied mathematically, through the introduction of several different discourse theories. Section 2.2 gives a brief overview of KDD, as well as describing one important method for knowledge discovery: Formal Concept Analysis.

2.1 Discourse representation

There are many different sub-fields within linguistics, including phonology, morphology, syntax, semantics, and pragmatics to name a few. Phonologists study the sounds of language, including phonemes, syllables, rhythm, and even gestures. Morphology is the analysis of linguistic units such as parts of speech, intonation, and stress. Syntax is the study of sentence structure, which culminates in attempts to form rules which govern the structure of a particular language. Semantics on the other hand is the study of meaning, often with a focus on separate utterances. Finally, pragmatics studies language meaning in a larger context, taking into account world knowledge, environment, and other factors which could influence language use. Our purpose, to represent meaning in medical text, means we will mostly focus on the last two sub-fields of semantics and pragmatics.

The study of semantics can be applied to relations between symbols, relations between words in a sentence, or relations between phrases in a sentence. Once one moves beyond that, to a level where the study focuses on relations between full sentences or even whole texts, it will be referred to as discourse analysis. Example 2.1.1 shows a case of discourse using two sentences, taken from an article about Duchenne muscular dystrophy[1]. One element of discourse representation is the extraction of discourse relations, which describe the relationship between two text segments. For example, there is a contrast relation between the two sentences in the example: one refers to a situation without a specific treatment, whereas the other describes the situation with that treatment.

Example 2.1.1
Without such treatment the children would die between the ages of 14 - 18 years as a result of severe respiratory complications such as pneumonia. With their respiratory problems resolved, however, the patients could enjoy a life extended by a number of years, with cardiomyopathies then becoming the life-limiting factor.

[1] `http://www.ncbi.nlm.nih.gov/pubmed/23620648`

Discourse relations can also be found within the sentences. There is a causal relation between two segments of the first sentence from Example 2.1.1, because it states that the death of children without treatment is caused by severe repiratory complications. In Example 2.1.2 we use square brackets to show how the sentence would be split based on this causal relation. So far we have seen the contrast and the cause relation, but there are several more types which often occur in text, including relations which illustrate a temporal progression between text units.

Example 2.1.2
[Without such treatment the children would die between the ages of 14 - 18 years][as a result of severe respiratory complications such as pneumonia.]

Extracting discourse representations from natural language text is not easy. There are many problems which need to be resolved before the representation is complete. Example 2.1.1 shows one example of such a problem: anaphora resolution, where pronouns (or other references) need to be linked to their antecedent. In this case one needs to link *such treatment* with some treatment option mentioned in previous sentences of the article, and the word *their* in the second sentence needs to be linked with *the children* from the first sentence as its antecedent (similarly for *the patients* in the second sentence).

Representing the semantics of discourse brings along many other difficulties, including presuppositions, modal subordination, and donkey sentences (part of anaphora resolution). For a description of these phenomena, see [Amblard and Pogodalla, 2014]. There are many theories concerning the structure and mechanisms of semantics and discourse in natural language. Ideally we would like to extract a complete discourse structure which is more complete than only extracting the discourse relations. Therefore we will provide a brief overview of several of theories, focusing on the shift from static semantics to more dynamic representations, as detailed in a paper by [Amblard and Pogodalla, 2014].

2.1.1 Montague Semantics

When referring to 'static' semantics, we in fact mean Montague semantics. Richard Montague developed this approach based on the idea that natural languages can be treated with the same mechanisms as formal languages.

> "There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory." -Richard Montague [Montague, 1974c]

The main mechanism used by Montague for analyzing semantics is first-order logic, where a sentence or phrase can be represented as a logical formula. These formulas are built on the principle of compositionality, that is to say, the meaning of a complex expression can be built up from the meaning of its parts. Hence, one starts by assigning logical representations to words, and the rest of the formula is built based on the syntax of the sentence. Example 2.1.3 shows the logical

representations of a few sentences varying in complexity.

Example 2.1.3
a. John loves Mary: $love(john, mary)$
b. Every man eats: $\forall x\, man(x) \rightarrow eat(x)$
c. If John owns a donkey, he is rich: $(\exists x.donkey(x) \wedge owns(John, x)) \Rightarrow rich(John))$

Interpretation of these logical formulas is based on model-theoretic semantics and truth-conditional semantics. The first is a popular approach to semantics by Alfred Tarski, using models to represent 'worlds', and the second is mainly associated with Donald Davidson, equating semantics with truth conditions. In this approach, terms in logical formulas can be mapped to individuals (assuming that we have some universe or vocabulary of individuals), with different mappings resulting in different models. Propositions can be evaluated as being true or false, relative to the model. So if you consider the meaning of *John loves Mary*, with respect to a certain model, then it is true only if John and Mary are entities in the universe and if John does in fact love Mary.

Montague semantics can effectively handle quantification, definite articles, ambiguity, and the various parts of speech such as adjectives and adverbs. For more detailed reading about the inner workings of this theory, see the three fundamental papers written by Montague [Montague, 1974a,c,b]. However, the theory also has a few shortcomings, some of which can be illustrated through the use of donkey sentences like in Example 2.1.4.

Example 2.1.4
If John owns a donkey, he beats it.
a. $(\exists x.donkey(x) \wedge owns(John, x)) \Rightarrow beats(John, x)$
b. $(\forall x.(donkey(x) \wedge owns(John, x)) \Rightarrow beats(John, x))$

The logical formula in 2.1.4a is what you would expect from the sentence according to the compositionality principle, because its structure is similar to the sentence in 2.1.3c. However there are two problems with this formula, first that the last occurrence of x does not fall under the scope of the quantification, and second that we would normally expect a different quantification in the first place. Hence the second formula, 2.1.4b, is what we would actually like to see.

Problems with donkey sentences, as well as other issues encountered by Montague semantics, can be solved by moving on to a dynamic approach to semantics. The next few theories we describe all belong to this category, and eventually lead to Segmented Discourse Representation Theory [Asher and Lascarides, 2003].

2.1.2 Discourse Representation Theory and Dynamic Predicate Logic

Originally developed by Hans Kamp, Discourse Representation Theory (DRT) introduces the key concept of a context. This forms a dynamic element, because sentences are not only interpreted on the basis of the context, but they can also

change the context. In basic terms, the context keeps track of items (usually noun phrases) introduced in earlier sentences, so they become available for anaphora resolution in subsequent sentences. A detailed description of DRT, including discussions on recent developments and issues, is provided by [van Eijck and Kamp, 1997]. Example 2.1.5 is taken from this same paper, and shows how the theory applies to a basic example.

Example 2.1.5
A man entered. He smiled.

$x\ y$
$man\ x$
$entered\ x$
$y \doteq x$
$smiled\ y$

In terms of logical formulas, the discourse presented in Example 2.1.5 can be expressed as $\exists x(man(x) \wedge entered(x) \wedge smiled(x))$. However, in order to properly represent the addition of context, DRT introduces a new representation called Discourse Representation Structure (DRS), which can be viewed as the table shown in the example. The top-most box shows the elements currently in the context, accessible to subsequent sentences, and the bottom box shows the knowledge already built up from previous sentences. In this case the $\doteq$ indicates equality between the two reference markers. This model can be applied to new sentences, which have access to the variables x and y, but new sentences can also update the model by placing new variables in the context and adding restrictions/knowledge about the variables to the model. Example 2.1.6, from [Amblard and Pogodalla, 2014], shows how the DRSs of two sentences combine to form one DRS.

Example 2.1.6
A man walked in. Another man followed him.

x
$man\ x$
$walked\_in\ x$

$\cdot$

$y\ z$
$man\ y$
$y\neg \doteq ?$
$z \doteq ?$
$followed\ y, z$

$=$

$x\ y\ z$
$man\ x$
$walked\_in\ x$
$man\ y$
$y\neg \doteq x$
$z \doteq x$
$followed\ y, z$

DRT works just as well as Montague semantics for quantification, modal subordination, and other linguistic phenomena. Depending on the structure of the discourse, a DRS can contain another DRS, making one context available to consecutive sentences and another context unavailable. For details about how DRT handles different linguistic features, see [van Eijck and Kamp, 1997]. There is one downside to DRT, as pointed out by [Groenendijk and Stokhof, 1991] and summarized by [Amblard and Pogodalla, 2014], namely that it does not always adhere to the compositionality principle. Although this principle causes issues with donkey sentences in Montague semantics, it is possible to solve these problems without breaking the principle itself, as shown by Dynamic Predicate Logic (DPL).

DPL was developed by [Groenendijk and Stokhof, 1991], with the aim of establishing a discourse theory which is empirically equivalent to previous theories, without discarding the compositionality principle. They compare it to programming languages, in that it works like transitions between machine states (assignments of items to variables). Furthermore, it goes back to a representation in first-order predicate logic, like in Montague semantics.

Example 2.1.7
A man entered. He smiled.
$\{\langle g, h\rangle | h[x]g \wedge man(h(x)) \wedge entered(h(x)) \wedge smiled(h(x))\}$

Example 2.1.7 shows how the theory can be applied to simple discourse. The pair g and h are states (assignments) such that they form the interpretation of a program, where an input of state g can result in state h. The first condition $h[x]g$ means that the two states can differ at most in the assignment of variable x. Consider now the discourse in Example 2.1.7 and a universe where Mary, John, and Bill have all entered a room. Furthermore, consider a situation where state g is the input state before the above two sentences are seen, k is the state after the first sentence is seen, and h is the final output state after both sentences. Then we can see that $man(k(x))$ and $entered(k(x))$ must hold, meaning the state k (and by extension state h) can only assign John or Bill to variable x.

Dynamic Predicate Logic can deal with all the same linguistic phenomena that the previously described theories can handle, all without breaking the compositionality principle. However it does have some downfalls, one of which is the destructive assignment problem, which means that the last assignment of a variable in a program hides any previous assignments to that variable. This is a common problem in imperative programming languages, the paradigm which provided some of the inspiration for this theorem.

2.1.3 Segmented Discourse Representation Theory

So far we've seen simple examples of discourse which were all linear in structure, but this is not always the case. Therefore, Segmented Discourse Representation Theory (SDRT) aims to model the semantics of sentences within the structure of the discourse. It was developed by [Asher and Lascarides, 2003] as an extension of DRT, but it can be combined with other discourse representation theories [Asher and Pogodalla, 2011]. In order to define the structure of the discourse, SDRT relies on discourse relations which describe the relation between two text segments (or sentences in this case). Asher and Lascarides define two discourse relations for dealing with this particular example: Narration and Elaboration. A Narration relation between two sentences means there is a temporal progression from one sentence to the other, which can be viewed as a type of coordination, whereas an Elaboration relation means the second sentence adds more information to what was stated in the first sentence, viewed as a subordination. These relations are used within the framework of Example 2.1.8, taken from [Asher and Lascarides, 2003].

Example 2.1.8
a. π_1 *Max had a great evening last night.*
b. π_2 *He had a great meal.*
c. π_3 *He ate salmon.*
d. π_4 *He devoured lots of cheese.*
e. π_5 *He then won a dancing competition.*
f. $*\pi_6$ *It was a beautiful pink.*

When reading through the discourse in this example, the last sentence clearly feels out of place because *It* refers to the salmon which was introduced to the context three sentences earlier. In regular DRT this sentence would be accepted, but in SDRT this type of situation will be rejected for being ungrammatical. The relationship between discourse types is further illustrated in Figure 2.1.

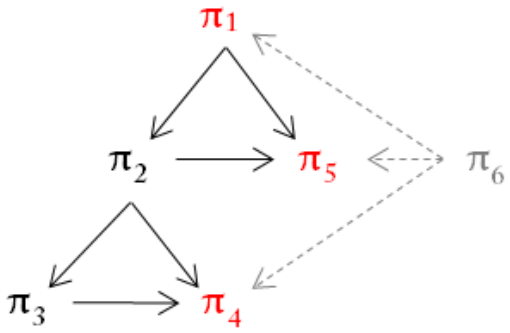


Figure 2.1: Discourse structure using discourse relations Narration and Elaboration

The figure shows how the sentences in the example discourse are connected by the two discourse relations, for example there is an Elaboration relation between *Max had a great evening last night* and *He had a great meal*, also represented as **Elaboration**(π_1, π_2). All of the downward facing arrows represent such an Elaboration, whereas all of the horizontal arrows represent a Narration relation, for example **Narration**(π_3, π_4) between the sentences *He ate salmon* and *He devoured lots of cheese*. Now, the Right Frontier Constraint (RFC) is the key principle used in SDRT to ensure that the addition of sentence π_6 is ungrammatical. RFC restricts the points in the structure where a new sentence can be attached, by only allowing new attachments to the last sentence and every sentence for which it is a subordinate. In this case it restricts the possible points where π_6 can be attached to sentences π_1, π_4, and π_5. None of these options will allow the anaphora resolution algorithm to find a sensible antecedent to *It* in *It was a beautiful pink*, which results in the sentence being correctly rejected from this discourse.

2.1.4 Discourse Relation Algebra

There is one more advancement to SDRT based on discourse relations, which was introduced by [Roze, 2011], who described a method for building inference rules for discourse relations. The aim is that these inference rules can be used to deduce

the complete set of relations of a text (i.e. the discourse closure). Essentially, it forms an algebra, where one can abstract over the known discourse relations to infer more relations until every possible relation has been found. It could be useful for merging discourse annotations done by different annotators, possibly referring to different discourse theories. Roze used a simple example to illustrate the basic idea, repeated here as Example 2.1.9.

Example 2.1.9
a. π_1 *It has rained a lot today.*
b. π_2 *So John cooked.*
c. π_3 *He made a pie.*

Consider the situation where an annotator marks the relations $Result(\pi_1, \pi_2)$ and $Elaboration(\pi_2, \pi_3)$. This would be a correct annotation, but it would not be entirely complete. It is possible to formulate an inference rule of the form $Result(\pi_1, \pi_2) \wedge Elaboration(\pi_2, \pi_3) \rightarrow Result(\pi_1, \pi_3)$, whose result could then be added to the annotation to make it complete. Roze builds several such rules and describes the process for doing so, although the set of rules is currently far from complete. Still, this method is a promising option for possibly completing future discourse-annotated corpora.

Automatically extracting discourse structures and sentence semantics based on SDRT and any of the other theories described so far is still a difficult task. However it is clear that applying such a structure to texts would be ideal for analyzing data and extracting even more information. Indeed one application which could benefit greatly from these theories is automatic summarization of (medical) articles. If these methods are applied to a large number of articles and texts, it becomes impossible to analyze manually, and they need to be linked to knowledge discovery methods for further analysis. Therefore the next section will provide a short summary of such methods.

2.2 Knowledge Discovery in Databases

Knowledge discovery in databases (KDD) is the process of extracting knowledge from a large set of data. Traditionally, data was evaluated manually by experts in a certain domain, but the amount of data being stored these days has far exceeded our analysis capabilities. From thousands of satellite images which need to be scrutinized for new celestial objects, to databases filled with individual customer purchases which need to be analyzed for new trends in spending, KDD is applicable to many different fields. It aims to develop tools and theories for automatically extracting knowledge from a huge database, which can then be evaluated by the human experts.

[Fayyad et al., 1996] describe KDD as a process which encompasses all the steps required to apply the actual extraction algorithms (also called data mining), including data preparation and evaluation. They use Figure 2.2 to illustrate the five basic steps, starting with initial, unstructured data usually stored in a database. The first step, **selection**, consists of deciding which data sources to

us, if there are multiple sources available, and possibly using a selection criterion to cut down on the number of data instances which will be analyzed. This results in sets of target data for use in the second step called **preprocessing**. It can consist of cleaning the data by removing noise, filling in missing data, or combining data from different sources, finally resulting in preprocessed data for the third step. **Transformation** refers to formatting the data so it can serve as input to a data mining algorithm. Sometimes this step requires data reduction or simplification of some kind for the algorithm to be applicable. The fourth step is the **data mining** itself, the application of some algorithm which attempts to extract patterns or other information from the data. Any machine learning algorithms can be used in this step, depending on the goal of the extraction. Common examples include classification or clustering algorithms. Patterns which are outputted by the algorithm need to then be **interpreted and evaluated** in the final step, usually by experts in the domain of interest, to extract the useful knowledge which can be gained from them. The final step often includes visualization of some kind, to make the evaluation easier for human experts.

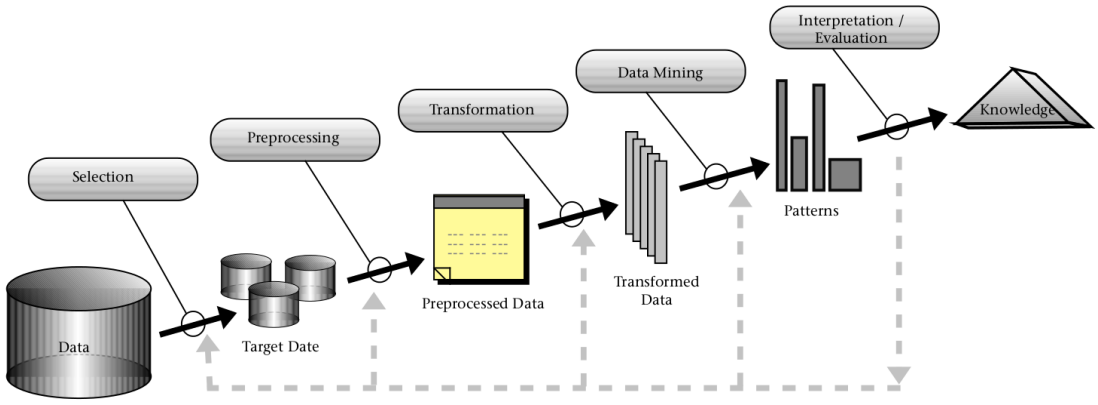


Figure 2.2: The five steps of knowledge discovery in databases (KDD)
Source: Fayyad et al. [1996]

Notice that the KDD process is a recursive one. It is normal to evaluate the results at each stage of the process, and to go back to an earlier stage for applying improvements. One interesting algorithm which can be applied during the fourth step of the KDD process is Formal Concept Analysis (FCA), which can handle complex data by mathematically defining concepts. Section 2.2.1 will provide a brief overview of FCA with examples, and in Section 2.2.2 we will outline an extension of FCA called pattern structures which allow us to apply the methods to complex data.

2.2.1 Formal Concept Analysis

Formal Concept Analysis (FCA) is a mathematical theory for the analysis of data which is based on the notion of formal concepts, forming a concept lattice. Formal concepts are defined as units which have an extent and an intent. The extent is a set of objects, the instances of the concept, whereas the intent is a set of

attributes, forming a description which is common to all the instances. A lattice structure can organize the concepts based on relations between the extents and intents, showing how some concepts generalize over others. The lattice also serves as a useful visualization tool which domain experts can use to extract knowledge. Here we will provide only the basic definitions of FCA; a detailed description with proofs can be found in [Ganter et al., 1997].

FCA starts with a formal context $\mathbb{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes, and I is a binary relation between the two sets. Hence, gIm means the object g has attribute m. A simple example of a formal context in the medical domain, is shown in Table 2.1. Here, the set of objects is $G = \{breastCancer, asthma, lungCancer\}$ and the set of attributes is $M = \{isCancer, requiresInhaler, causedBySmoking, foundInAdults\}$. A cross in the table indicates that gIm, whereas an empty cell shows that there is no relation between that particular object-attribute pair. This particular example is very simplified, since patients with asthma do not always require an inhaler, and smoking is only one of the many possible causes for both asthma and lung cancer, but it shows the basic idea.

	isCancer	requiresInhaler	causedBySmoking	foundInAdults
breastCancer	X			X
asthma		X	X	X
lungCancer	X		X	X

Table 2.1: Simple formal context

To create formal concepts from a formal context, we need derivation operators. There are two such operators, both represented by prime ($'$), one for a set of concepts and one for a set of attributes. Consider a set of objects $A \subseteq G$, then A' defines the set of attributes which are shared by all objects in A, defined as $A' = \{m \in M |\, gIm\, \forall g \in A\}$. Similarly, consider a set of attributes $B \subseteq M$, such that B' defines the set of objects that carry all the attributes in the set, formally defined as $B' = \{g \in G |\, gIm\, \forall m \in B\}$. Example 2.2.1 shows how the derivation operators can be applied to the formal context in Table 2.1. The first case shows that when you consider the set of objects containing only $asthma$, the derivation operator returns the set of all attributes which apply to that object. Of course the operator can be applied to a larger set, as shown in the second case where the set of objects contains both $breastCancer$ and $lungCancer$. In that case the operator returns the set of attributes which both of the objects share. The third case shows the derivation operator applied to a set of attributes, returning the set of objects to they apply. And finally the fourth case illustrates that the derivation operator can return an empty set, since there isn't a single object which both is a cancer and requires an inhaler.

Example 2.2.1
a. $\{asthma\}' = \{requiresInhaler, causedBySmoking, foundInAdults\}$
b. $\{breatCancer, lungCancer\}' = \{isCancer, foundInAdults\}$
c. $\{requiresInhaler\}' = \{asthma\}$
d. $\{isCancer, requiresInhaler\}' = \{\}$

Not all of the pairs of sets shown in Example 2.2.1 can form formal concepts. Similar to the examples above, a formal concept is a pair of sets (A, B) where $A \subseteq G$ and $B \subseteq M$. However one important restriction on formal concepts is that $A' = B$ and $B' = A$; when this restriction holds, you have a formal concept (A, B) where A is called the extent and B is called the intent. Based on the definition, it should be clear that the two sets shown in Example 2.2.1c do not form a formal concept, because $\{requiresInhaler\}' = \{asthma\}$ but $\{asthma\}' = \{requiresInhaler, causedBySmoking, foundInAdults\}$. Notice that the first two examples do indeed form formal concepts. The double prime ($''$) operator is a closure operator, and can therefore be used to find closed sets of concept extents and concept intents. It is illustrated using Example 2.2.1c below:

$$\{requiresInhaler\}' = \{asthma\}$$

$$\{asthma\}' = \{requiresInhaler, causedBySmoking, foundInAdults\}$$

$$\{requiresInhaler\}'' = \{requiresInhaler, causedBySmoking, foundInAdults\}$$

The complete set of formal concepts belonging to a formal context is denoted by $\mathfrak{B}(G, M, I)$, in contrast to the concept lattice, which is denoted by $\underline{\mathfrak{B}}(G, M, I)$. To build the concept lattice, one needs to define a partial order on formal concepts:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$$

which is equivalent to

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow B_2 \subseteq B_1$$

Hence the formal concept (A_1, B_1) is the sub-concept of (A_2, B_2), or reversely, the latter is the super-concept of the former. This partial order can be illustrated with an example from our medical formal context:

$$(\{asthma\}, \{requiresInhaler, causedBySmoking, foundInAdults\}) \leq (\{breastCancer, asthma, lungCancer\}, \{foundInAdults\})$$

By introducing this partial ordering, concepts can now be organized in a concept lattice. A complete lattice is a lattice where for any two concepts, the greatest lower bound (infimum) and the least upper bound (supremum) always exist. In the case of a join-semi-lattice, only the supremum is defined for any two elements, and in the case of a meet-semi-lattice, only the infimum is defined for any two elements. In the case of concepts, the infimum and supremum are based on the double prime closure operator:

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)'' \right)$$

$$\bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)'', \bigcap_{t \in T} B_t \right)$$

There are several tools available for performing FCA; in this case we used the Galicia[2] tool to build the concept lattice of our small medical example as shown in Figure 2.3. Each node of the lattice is a formal concept, and each edge shows a partial order relation between two concepts. Every node is labeled with its intent (I) and its extent (E). Notice that the top node has an extent which contains all of the objects in the context, whereas the bottom node has an intent which contains all of the attributes. The lattice demonstrates a generalization/specialization between concepts: consider concept 2 as an example, which has an intent of $\{foundInAdults, isCancer\}$ and an extent of $\{breastCancer, lungCancer\}$. Any concept it is linked to which takes its place above concept 2 in the lattice is a generalization, which is only concept 0 in this example. In contrast, any concept which concept 2 links to and which is placed lower in the concept lattice is a specialization, which here includes concepts 4 and 5.

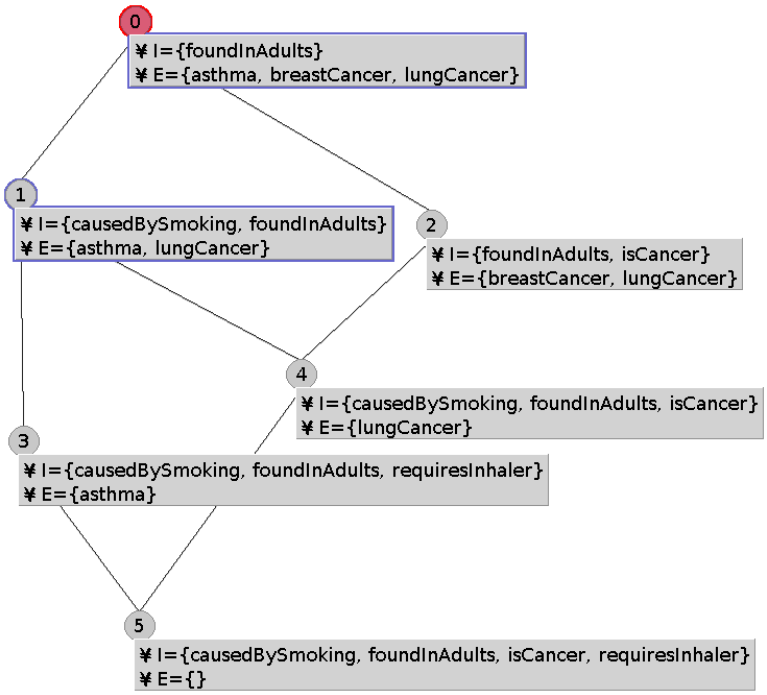


Figure 2.3: Simple lattice of our medical formal context

There are a few extensions to basic FCA, such as Relational Concept Analysis (RCA) and pattern structures. The former serves to model relationships between objects of FCA. Consider a situation where there are two formal contexts: one which features different hospitals and their facilities, and one which features patients with details about their symptoms. Using RCA it becomes possible to define a *mostSuitable* relation between the objects of both formal contexts (hospitals and patients) regarding which hospital has the most suitable facilities for a patient with certain symptoms. Pattern structures, on the other hand, allow us to apply FCA to complex data which cannot be modeled in a binary context. Since we aim to work with discourse in medical text, the pattern structures extension is most relevant in this case and will be described in more detail in section 2.2.2.

[2] `http://www.iro.umontreal.ca/~galicia/`

2.2.2 Pattern Structures

For many-valued data it is possible to apply a range of scaling procedures to transform data into a binary context, such as nominal, ordinal, or interordinal scaling. However these scaling methods often result in massive contexts, and limiting the size of the context would result in loss of information. Pattern structures avoid this build-up of attributes by working directly with the original complex data. They can be applied to multi-valued data [Kaytoue et al., 2011], data in the form of graphs [Kuznetsov, 1999], and indeed textual data [Coulet et al., 2013]. The key idea of pattern structures for handling complex data, is that one must be able to compare the data descriptions. More specifically, it must be possible to define a similarity operator which enables ordering the descriptions in the form of a semi-lattice. It is then possible to build a concept lattice directly from the complex data descriptions without any loss of information.

Let us assume a set of objects G and a set of their descriptions D (which we will call patterns). The set of patterns D varies from the set of attributes M which is used in basic FCA, in that the patterns are more complex (they can be sets, intervals, graphs, or other formats). Next, there has to be a meet-semi-lattice $(D, \sqcap)$ which provides an order for the patterns. In basic FCA, where M is a set of attributes, this order is defined by the set-intersection operator $\cap$. This means that when you have several sets of attributes, each describing one or more objects, these sets can be ordered in a meet-semi-lattice such that more specific elements of the lattice are subsets of the more general elements. For complex patterns where the intersection operator does not suffice, one must define a different similarity relation between the patterns, such that $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, where $c, d \subseteq D$ and $\sqcap$ is the similarity operator which will be used to describe the similarity between patterns according to a semi-lattice.

Given the definition of the similarity operators, based on the type of patterns which are being used, the whole pattern structure can be represented as $(G, (D, \sqcap), \delta)$. Comparing this to the original definition of a formal context in FCA, which was (G, M, I), it is clear that G still represents a set of objects like it did before, the set of attributes M has been replaced with a meet-semi-lattice of patterns $(D, \sqcap)$, and the binary relation I has been changed to a mapping from objects to their patterns: $\delta : G \rightarrow D$. To create formal concepts from pattern structures it is also necessary to define new derivation operators. Again there are two operators, both represented by the box symbol, one which applies to a set of objects, and one which applies to a pattern. The first operator returns the most general pattern which describes all of the objects in set A, whereas the second operator returns the set of objects which can be described by pattern d. Like in simple FCA, applying the derivation operator twice gives the closure operator $(.)^{\square\square}$.

$$A^{\square} = \bigsqcap_{g \in A} \delta(g)$$

$$d^{\square} = \{g \in G | d \in \delta(g)\}$$

As in basic FCA, the derivation operators can be used to form formal concept pairs (called pattern concepts in the case of pattern structures). A pattern

concept is of the form (A, d) where $A \subseteq G$, $d \in D$, $A^{\square} = d$, and $A = d^{\square}$. In this case, A is called the pattern extent while d is called the pattern intent. These pattern concepts can be organized into a concept lattice just as before. We will now illustrate the idea of pattern structures using an example from [Coulet et al., 2013], who also applies them to textual data in the medical domain. For more detailed theoretical information about pattern structures, see [Ganter and Kuznetsov, 2001].

In [Coulet et al., 2013], they compare documents describing medical drugs based on annotations from the National Cancer Institute (NCI) Thesaurus [3]. The annotations form the descriptions of each document, which is too complex for a binary context and therefore requires pattern structures to handle it within a FCA framework. The NCI Thesaurus is in fact an ontology, where the terms are organized in a tree-like structure of specialization/generalization. Furthermore terms from the thesaurus map to a semantic type from the Semantic Network of the Unified Medical Language System (UMLS) Metathesaurus[4], which is also an ontology in the form of a tree-like structure. An expert can choose categories from the Semantic Network according to the information (s)he is interested in. Then each document is scanned for terms which appear in the NCI Thesaurus and belong to one of the semantic categories chosen by the expert. In the example described in [Coulet et al., 2013] there are four semantic categories, but we will simplify their example to two semantic categories: Disease or Syndrome, and Molecular Function. Table 2.2 shows the (adapted) formal context. The two attribute columns each correspond to a semantic type chosen by an expert. Rows are documents describing a certain drug. Each cell shows the set of terms found in the document, which are described by the NCI Thesaurus and belong to the semantic category of the particular column. So the document describing Drug1 contains mentions of tuberculosis and bacterial infection, which are terms belonging to the thesaurus and correspond to the semantic type Disease or Syndrome.

	Disease of Syndrome	Molecular Function
Drug1	{Tuberculosis, Bacterial_Infection}	{Protein_Synthesis}
Drug2	{Bacterial_Infection}	{Protein_Synthesis}
Drug3	{Tuberculosis, Bacterial_Infection}	{}
Drug4	{Tuberculosis}	{Protein_Synthesis}
Drug5	{Tuberculosis, Bacterial_Infection}	{}

Table 2.2: Adapted formal context example in medical domain
Source: Adaptation from Coulet et al. [2013]

Each row from Table 2.2 forms a pattern describing the document in question. So the document which describes Drug2 has a description:

$$\{Bacterial\_Infection\}\{Protein\_Synthesis\}$$

[3] `http://ncit.nci.nih.gov/`
[4] `http://www.nlm.nih.gov/research/umls/`

Every document in the set of objects can be described with this pattern of two sets of terms. Therefore the set of objects G and the set of patterns D for this example are:

$$G = \{Drug1, Drug2, Drug3, Drug4, Drug5\}$$
$$D = \{\{Tuberculosis, Bacterial\_Infection\}\{Protein\_Synthesis\},$$
$$\{Bacterial\_Infection\}\{Protein\_Synthesis\},$$
$$\{Tuberculosis, Bacterial\_Infection\}\{\},$$
$$\{Tuberculosis\}\{Protein\_Synthesis\}\}$$

So to apply pattern structures, there needs to be a similarity operator, which is where the ontology structure comes in. Figure 2.4 shows a small part of the NCI Thesaurus ontology which is relevant to this example; again adapted from [Coulet et al., 2013]. The tree shows how the terms found in the text can be ordered, and it includes terms which were not found in the text but are present in the ontology (like Mycobacterial_Infection in this case). Semantic types of each term are shown for reference, but are not part of the ontology.

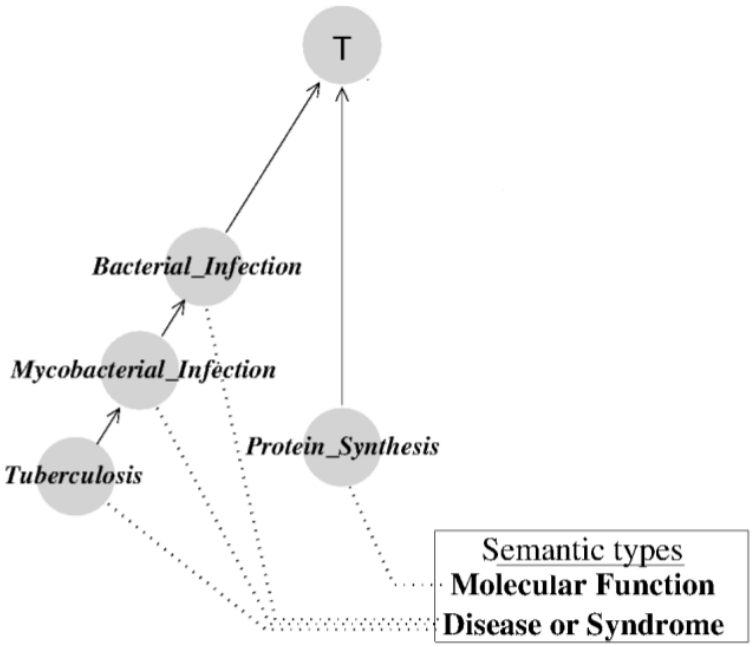


Figure 2.4: Small part of the NCI Thesaurus ontology
Source: Adaptation from Coulet et al. [2013]

The authors define the similarity operator as being the convex hull, which is the smallest convex set of the set of terms it is applied to. Convex set refers to the set which includes the initial terms and every term between them and their least common subsumer (the most specific term which subsumes all initial terms). Consider the set $\{Tuberculosis, Bacterial\_Infection\}$; in this case the least common subsumer is $Bacterial\_Infection$ because it is the most specific term which subsumes both terms. The top node T also subsumes both terms, but it is less specific than $Bacterial\_Infection$. Since the convex hull is the set

of initial terms, the least common subsumer, and everything in between it will be:

$$Conv(\{Tuberculosis, Bacterial\_Infection\}) =$$
$$\{Tuberculosis, Mycobacterial\_Infection, Bacterial\_Infection\}$$

In this case the set of patterns is small enough that it is possible to draw the whole meet-semi-lattice formed by the similarity operator. It is shown in Figure 2.5, and is also an adapted version of a similar image from [Coulet et al., 2013]. From this image it is possible to see every combination of patterns and how they would be ordered.

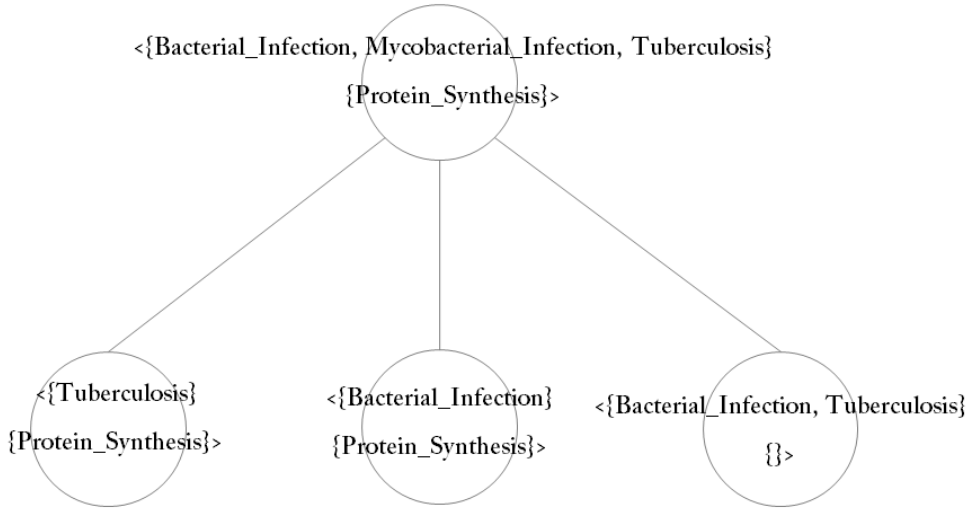


Figure 2.5: Meet-semi-lattice of all patterns from working example
Source: Adaptation from Coulet et al. [2013]

Before now, we have not described how to construct the actual formal (or pattern) concepts which form the lattice. There are several algorithms available for achieving this, but one of the more popular ones is the CloseByOne algorithm by [Kuznetsov, 1993], which is modified slightly for use in pattern structures and used in both [Kaytoue et al., 2011] and [Coulet et al., 2013]. The algorithm creates concepts from the bottom up, starting with concepts which have the smallest extents. Every time it generates a new concept, it expands upon that concept by adding one more object to the extent (determined by a linear order of the objects) and it then applies the closure operator to generate another closed concept. Intents are computed by intersecting the intent of the original concept with the pattern of the added object. Repeating this process recursively produces all of the closed concepts. To prevent generating identical concepts there is usually an auxiliary data structure storing existing concepts. However look-ups in such a data structure can be expensive, so there is also a canonicity test to determine if the concept is completely new or if it could have been generated before and is worth looking up. Consider A to be the extent of the old concept and C to be the extent of the new concept we want to generate; C is larger by one object g. If there exists another object h which appears before g in the linear order of objects, and which generates exactly the same set C when it is added to the set A, then it fails the canonicity test and the algorithm backtracks.

A few simple modifications are necessary to apply the CloseByOne algorithm on pattern structures. First, the original derivation operator needs to be replaced with the one defined for the pattern structure (shown in blue in the pseudo code). Second, one must also replace the intersection operator with the similarity operator which applies to the pattern structure (shown in red in the pseudo code). The pseudocode for this modified algorithm, as defined by [Coulet et al., 2013], is shown in Algorithm 1 and 2. Besides those two minor changes, the process is exactly as described above and has the same time complexity of $O(|G|^2|D||L|)$ where G is still the set of objects, D is the set of patterns, and L is the set of concepts.

Algorithm 1 CloseByOne Algorithm

$L = \varnothing$ ▷ L is the concept set.
for each $g \in G$ **do**
  $process(\{g\}, g, (g^{\square\square}, g^{\square}))$
end for

Algorithm 2 process(A, g, (C,D))

if $\{h|h \in C \setminus A \text{ and } h < g\} = \varnothing$ **then**
  $L = L \cup \{(C, D)\}$
  for each $f \in \{h|h \in G \setminus C \text{ and } g < h\}$ **do**
    $Z = C \cup \{f\}$
    $Y = D \sqcap \{f^{\square}\}$
    $X = Y^{\square}$
    $process(Z, f, (X, Y))$
  end for
end if

CloseByOne results in a list of closed concepts, which can be organized in a lattice structure like Fig 2.3; for an expert to evaluate and use in knowledge extraction. The ability of pattern structures to handle complex data, makes it a favorable choice for our aim of mining textual data by taking into account discourse structure. The only restriction on the type of data which pattern structures can handle is that it must be possible to define a similarity operator on the pattern descriptions for establishing an order.

3. Data collection

A successful combination of Kowledge Discovery in Databases (KDD) and Natural Language Processing (NLP) could be applied to helping medical experts discover new knowledge about rare diseases through the analysis of existing articles. For this reason we aim to use medical text in our experiments, although we do not limit ourselves to rare diseases only.

3.1 Types of text

PubMed[1] is a service which provides free access to a large database of scientific articles, case reports, and other texts concerning a large number of diseases and ailments. It is made accessible by the US National Library of Medicine[2], and contains mostly content in English, although there are a few articles present in other languages. Figure 3.1 shows a screenshot of a PubMed article about fibromuscular dysplasia, which is a typical example of what we use for this research. There is a tool available to download a large number of article abstracts automatically, but the full-text articles need to be accessed manually.

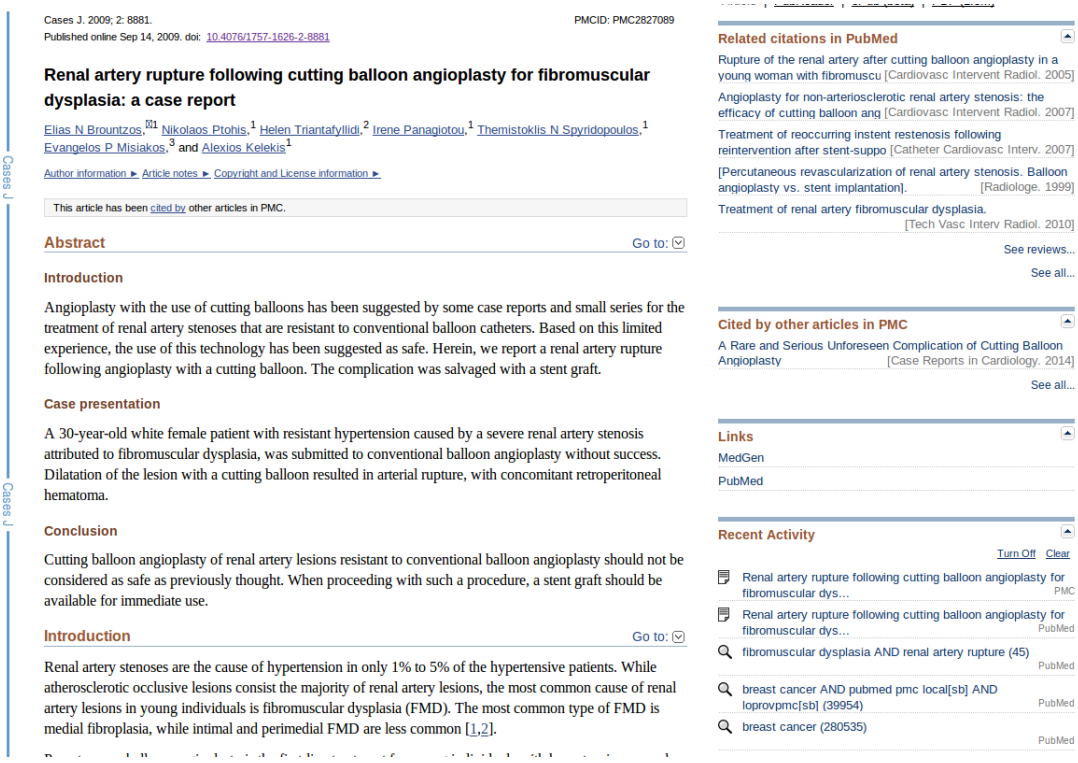


Figure 3.1: Screenshot example of a PubMed article
Source: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2827089/

[1] `www.ncbi.nlm.nih.gov/pubmed/`
[2] `www.nlm.nih.gov/`

For this reason we do a preliminary evaluation to determine how many discourse relations can be extracted from a collection of abstracts as opposed to a collection of full-text articles and case reports. The details of discourse relation extraction are presented in Chapter 4; but it is important to know that the more discourse relations are extracted, the more complete our representation of the discourse of an article is. Our first preliminary corpus consists of 162 abstracts from articles about hereditary hemorrhagic telangiectasia, which results in a total of 1474 sentences. The full-text corpus consists of 10 articles about the same disease, resulting in a total of 1692 sentences. As expected, the full-text articles do yield more discourse relations relative to the number of sentences, in fact the full-text corpus contains at least 7% more relations than the abstract corpus. So despite the extra manual work required in the absence of a simple download tool, we choose to build our corpus using full-text articles and case reports.

Most of the data on PubMed comes in the form of articles or case reports. Both types of text contain a lot of discourse information which can be extracted. Case reports usually report the medical experience of one or two patients, therefore yielding a lot of temporal relations as the text describes changes in their situation over a period of time. Medical articles usually describe a piece of research regarding specific treatments, the presence of certain genes, or the prevalence of a disease among a demographic group. These types of articles contain a lot of information about causal relationships between drugs, genes, demographic groups, and disease. Condition relations are also prevalent, concerning the types of situations which caused a patient's condition to improve or worsen. Other articles provide a historical account of how treatments and attitudes towards disease changed over time. Naturally this is another source of temporal relations, and also provides a lot of contrast relations between different situations in varying time periods.

Although there is a lot of discourse structure present in medical articles, there are also disadvantages due to its relative complexity. If one wants to apply any type of parsing, named entity recognition, stemming, or other preprocessing tool, it is usually necessary to find one specifically trained for medical data. The terminology is so different from standard news articles that most of the popular tools work very badly. Fortunately, although the terminology is complex, the sentence length is at an average 18.3 words for the entire corpus.

3.2 Building a corpus

We randomly choose 12, not necessarily rare, diseases from the PubMed database in order to extract 50 articles or case reports from each. Each search on PubMed is ordered by relevance to the query, and the list of diseases is shown below.

- polycystic ovary syndrome
- hereditary hemorrhagic telangiectasia
- breast cancer
- primordial dwarfism
- kawasaki disease
- myasthenia gravis

- lupus erythematosus
- renal failure
- synovial sarcoma
- septic arthritis
- fibromuscular dysplasia of arteries
- Duchenne muscular dystrophy

Although it is possible to download the articles in PDF format, we collect the HTML files for easier processing. There are many tools available for removing HTML tags, but the HTMLParser tool included in Python allows us to extract only the paragraphs from an article, leaving out other text such as the footnotes, and captions to figures, which can cause problems for the sentence tokenizer applied afterwards. This tokenization is performed with the Punkt sentence tokenizer provided by the Natural Language Toolkit (NLTK). Unfortunately the tokenizer handles references very poorly, especially references which appear just after the end of a sentence, like the two sentences in Example 3.2.1.

Example 3.2.1
Tracheostomy is effective in severe or emergent cases.7 Respiratory stimulants such as caffeine and doxapram, commonly used for apnea of prematurity and respiratory depression after anesthesia, could be a future treatment option in babies with achondroplasia, due to the stimulation of breathing on the medullary respiratory centers and carotid bodies; however, they have not been evaluated for use in this patient population.12

We attempted to remove all references from the articles, but because there are so many different referencing styles this only lead to different problems with tokenization and sentence legibility. Other tokenization tools did not fare much better, so we use the results as is and propose more specific solutions based on careful evaluation of the discourse relation extraction results in Section 4.5. Hence the resulting corpus is saved in XML format and contains a total of 600 articles.

3.3 External resources

As mentioned before, medical text requires specialized preprocessing tools since it contains so much domain-specific terminology. Fortunately there are a few tools available, including thesauri and a named entity recognizer, which we will briefly introduce here. We utilize these tools when building the pattern structures in Formal Concept Analysis, which will be described in detail in Section 5.3.

One of the most important sources for medical texts is the Unified Medical Language System (UMLS)[3] from the U.S. National Library of Medicine. It provides several tools, including a large MetaThesaurus which incorporates terms from several different medical thesauri, and a Semantic Network[4] which provides a categorization of the concepts from the MetaThesaurus. The MetaThesaurus is a large collection of medical terms, in fact the 2014AA release contains 2,973,458 concepts, and it provides additional information about each term including the

[3] `http://www.nlm.nih.gov/research/umls/`
[4] `http://semanticnetwork.nlm.nih.gov/`

variations in names for a term, the preferred name for the term, a unique concept ID called the CUI, relationships between different terms, short definitions of the term, and links to semantic types in the Semantic Network [National Library of Medicine, 2009]. In order to match medical text with terms found in the MetaThesaurus, there is the MetaMap tool[National Library of Medicine, 2013], which is essentially a named entity recognizer for medical terms. It provides several candidates in the form of MetaThesaurus terms for every phrase in a sentence, ranked according to a confidence level. Example 3.3.1b shows the set of concepts which MetaMap recognizes based on the sentence in 3.3.1a, when we choose the top candidate for each phrase.

Example 3.3.1
a. On the other hand, it has been shown that BMP9, a liver-specific BMP, is present at significant levels in both mouse and human plasma (13, 14), suggesting that it could act systematically on the endothelium where ALK1 is expressed.
b. {Hand, Show, BMP9 (GDF2 gene), BMP (Bone Morphogenetic Proteins), Present, Mouse, human plasma, Suggest, ACT, Endothelium}

Although the tool does generate quite a bit of noise, like recognizing the term *Hand* from the phrase *on the other hand*, it also recognizes a lot of terms correctly. Removing stop words from the sentence does not have any effect on the amount of noise which the tool produces. Furthermore, the tool can only process ASCII text, so some information contained in our corpus might be lost during the conversion. Despite these disadvantages, MetaMap works well for adding additional semantic information to textual data, especially when the terms it generates can be connected to semantic types in the Semantic Network.

Every MetaThesaurus term links to at least one semantic type in the Semantic Network. The network forms a tree-like structure of 133 semantic types, with 54 relationships between them. Some examples of major semantic types are organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas. There are several types of relationships between semantic types, such as *spatially_related_to* and *functionally_related_to*, but the most general relationship is the *is_a* relation. Figure 3.2 shows a small portion of the network using the *is_a*, taken from [National Library of Medicine, 2009]. The tree structure of the Semantic Network allows us to use it in the creation of pattern structures, which is further described in Section 5.3.

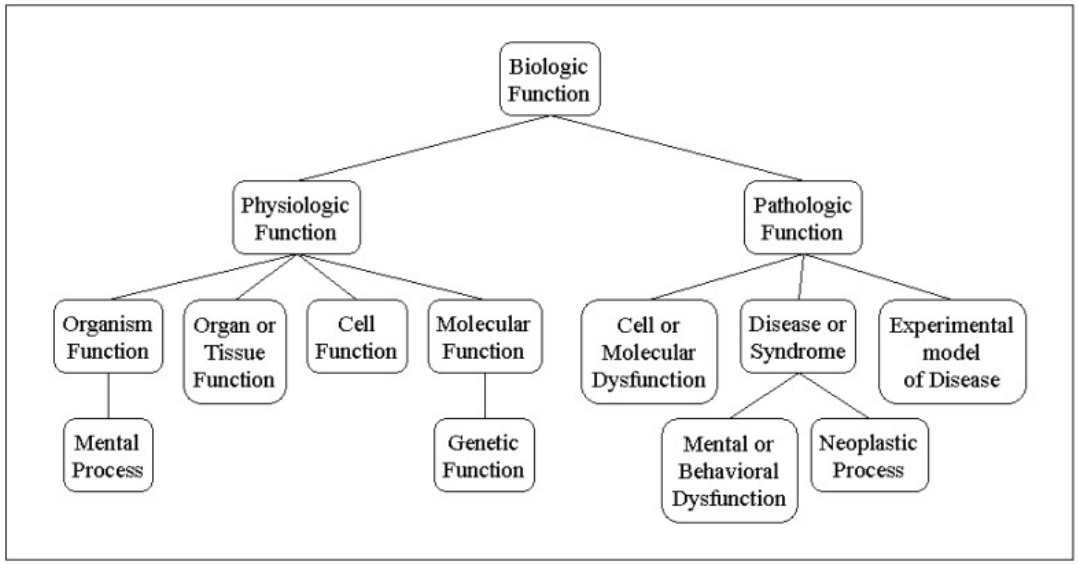


Figure 3.2: Small portion of the UMLS Semantic Network
Source: National Library of Medicine [2009]

4. Extracting discourse relations

There exist many well-developed theoretical representations of discourse in language (see Section 2.1), but extracting discourse structures from text in practice is still a challenge. Some research focuses on extracting discourse relations between or within elementary units of texts like sentences [Marcu and Echihabi, 2002, Sporleder and Lascarides, 2008, Sporleder, 2008], whereas others focus on generating a more complete discourse structure either at a sentence-level [Baldridge and Lascarides, 2005, Soricut and Marcu, 2003, Wellner et al., 2009] or at a document-level [Muller et al., 2012].

The latter task of extracting a complete discourse structure is more ambitious in that it aims to capture the discourse at several levels, resulting in tree or graph representations. However the methods employed so far rely heavily on the availability of annotated training data, and the range of corpora with discourse-related annotations currently available is very limited. Furthermore the different corpora which do feature discourse-related annotations are based on different discourse theories, so they cannot be used in combination without first finding methods to merge them. [Soricut and Marcu, 2003] relied on the RST Discourse Treebank [Carlson et al., 2002] which is based on Rhetorical Structure Theory (RST) and results in tree structures of discourse. [Baldridge and Lascarides, 2005] performed their own annotation on dialogues found in the Redwoods Treebank [Oepen et al., 2002], using Segmented Discourse Representation Theory (SDRT). Whereas [Muller et al., 2012] used ANNODIS [Afantenos et al., 2012], a French-language corpus also based on SDRT. Finally, [Wellner et al., 2009] performed experiments on the Discourse GraphBank [Wolf et al., 2004]. As the name suggests, this corpus contains discourse structures in the form of graphs instead of trees, which allows for more complicated features such as discourse elements with multiple parents, or cross-dependencies. Although the results achieved through supervised methods involving corpora is generally promising, none of the resources mentioned above are based on medical texts. Applying these resources to the medical articles extracted from PubMed would lead to high sparsity, and consequently give worse results.

There also exist unsupervised methods for extracting discourse from text, which focus on the sub-task of finding discourse representations in the form of discourse relations between texts segments, but allow us to circumvent the problems associated with the corpora. [Marcu and Echihabi, 2002] and [Sporleder and Lascarides, 2008] both focused on this approach, where they looked for discourse markers to indicate a relation between segments, and subsequently trained classifiers to recognize discourse relations even when the markers are not present. We use their methods to create our own corpus of discourse-annotated medical articles. Section 4.1 describes the discourse relations we considered, after which Section 4.2 lists the discourse markers and their patterns. An overview of the method along with our initial results can be seen in Section 4.3, which is evaluated in Section 4.4. Finally, Section 4.5 lists the improvements which were adopted and shows the final results.

4.1 Choosing discourse relations

Discourse relations (also called rhetorical relations) describe the connection between two text segments. Here, we consider two types of relations: intra-sentential relations occur between two parts of the same sentence, whereas inter-sentential relations occur between two whole sentences. [Marcu and Echihabi, 2002] illustrated the concept of relations with the following two examples:

Example 4.1.1
a. Such standards would preclude arms sales to states like Libya, which is also currently subject to a U.N. embargo.
b. But states like Rwanda before its present crisis would still be able to legally buy arms.

Example 4.1.2
a. South Africa can afford to forgo sales of guns and grenades
b. because it actually makes most of its profits from the sale of expensive, high-technology systems like laser-designated missiles, air-craft electronic warfare systems, tactical radios, anti-radiation bombs and battlefield mobility systems.

As readers, we can in most cases automatically infer a contrast relation between the two sentences in Example 4.1.1, and an explanation relation between the pair of sentence segments in Example 4.1.2. These particular examples feature very obvious markers: the word *but* in the first example, and the word *because* in the second example. However even when these markers are not present, we can determine discourse of text through semantic interpretation and our knowledge of the world. As Marcu and Echihabi pointed out, the sentence 4.1.1.a can be semantically represented as $cannot\_buy\_arms\_legally(libya)$, the next sentence can be represented as $can\_buy\_arms\_legally(rwanda)$, our background knowledge tells us that $is\_similar(libya, rwanda)$, and all of this leads to the conclusion that there is a contrast relation between the two sentences.

Unfortunately such a robust semantic interpreter does not yet exist, so both [Marcu and Echihabi, 2002] and [Sporleder and Lascarides, 2008] relied on the obvious markers in a text segment to determine the discourse relation. One major difference between their methods is the set of discourse relations they considered. Linguists do not agree on when it comes to the number of discourse relations or their definitions; each of the theories introduced in Section 2.1 has its own set of relations, some more detailed than others. Therefore, [Marcu and Echihabi, 2002] generalized the different theories based on the features they had in common, to create a small set of just four discourse relations: contrast, cause-explanation-evidence, condition, and elaboration.

In contrast, [Sporleder and Lascarides, 2008] chose a subset of relations defined by Segmented Discourse Representation Theory (SDRT): contrast, result, summary, continuation, and explanation. They chose the relations for which unambiguous markers are known, but which also appear in text without any markers, since the goal of the experiment was to use the former to classify the latter. Both sets of relations roughly overlap, and since [Marcu and Echihabi, 2002] took into account SDRT's relations in their generalization, we can convert

the relations and compare or combine the two methods.

4.2 Choosing discourse markers

Choosing the discourse markers which indicate discourse relations is another difficult task. Words can be used in so many different contexts that there is sure to be some noise in the data extracted using markers. [Marcu and Echihabi, 2002] chose some very common words, like *but* and *because*, based on evidence from previous research that these words result in contrast and cause-explanation relations respectively, the majority of the time. Patterns were built around these markers to determine where the text should be split into the two segments connected by the relation. Two examples of such patterns are shown in Example 4.2.1, where BOS and EOS stand for beginning-of-sentence and end-of-sentence respectively, and the two text segments are contained in square brackets.

Example 4.2.1
[BOS ... EOS] [But ... EOS]
[BOS ...] [because ... EOS]

Applying the second pattern to a simple example sentence shown in Example 4.2.2.a, results in the sentence being split into two text segments (indicated by square brackets) such that a cause-explanation relation holds between the two parts, shown in Example 4.2.2.b. In total Marcu and Echihabi listed 12 patterns, containing 8 distinct discourse markers.

Example 4.2.2
a. The apple is bruised because it fell from the tree.
b. [The apple is bruised] [because it fell from the tree.]

In comparison, [Sporleder and Lascarides, 2008] worked with a list of 55 discourse markers. They performed a corpus study to select only discourse markers which are unambiguous, meaning that the study showed the marker to indicate the same relation in each case. Sporleder and Lascarides also wrote detailed extraction patterns, but main goal of the patterns was to further disambiguate discourse markers and decrease the number of false positives in their data. Segmentation of the text was done afterwards by taking into account punctuation, the position of the marker, and linguistic background knowledge.

Although the Sporleder and Lascarides mentioned the importance of the patterns in ensuring the quality of the data, they unfortunately did not include their patterns in the paper. For this reason we start the analysis of our medical data using the discourse markers and patterns provided by Marcu and Echihabi, and consequently use their generalization of the set of discourse relations: contrast, cause-explanation-evidence, condition, and elaboration.

4.3 Extracting relations from medical text

All of the markers and patterns used by [Marcu and Echihabi, 2002] are listed in Table 4.1. Any words which are explicitly written are discourse markers, and the commas are important features of the patterns. In the case of complicated sentences with multiple segments separated by commas, we always consider the first comma which is encountered and stays true to the pattern. Square brackets indicate where the sentence(s) will be split into the two segments connected by the discourse relation.

CONTRAST
[BOS ... EOS] [BOS But ... EOS]
[BOS ...] [but ... EOS]
[BOS ...] [although ... EOS]
[BOS Although... ,] [... EOS]
CAUSE-EXPLANATION-EVIDENCE
[BOS ...] [because ... EOS]
[BOS Because ... ,] [... EOS]
[BOS ... EOS] [BOS Thus, ... EOS]
CONDITION
[BOS If... ,] [... EOS]
[BOS If...] [then ... EOS]
[BOS ...] [if ... EOS]
ELABORATION
[BOS ... EOS] [BOS... for example... EOS]
[BOS...] [which... ,]

Table 4.1: Initial set of patterns for relation extraction
Source: Marcu and Echihabi [2002]

Extraction is performed by going through every article in the XML corpus, sentence by sentence. Each individual sentence is checked against every intra-sentential pattern in Table 4.1. A sentence may only contain one discourse relation, since splitting a sentence on several relations requires finding an order or structure between them (often in the form of a tree). Afterwards, the current sentence and the next sentence adjacent to it are compared against every inter-sentential relation in the table. Once again, only one relation may hold between two adjacent sentences. This means that in total, a sentence can have maximum three relations: one with the previous sentence, one within the sentence itself, and one with the next sentence.

This method results in a total of 10,962 relations found in 81,505 sentences, with more detail shown in Table 4.2. In this table, the third column shows a ratio calculated by dividing the number of relations found by the number of sentences, and this multiplied by 100. The last column shows this very same ratio calculated with the results reported by [Marcu and Echihabi, 2002], to examine the differences between our results. In their experiments, Marcu and Echihabi used what they called a Raw corpus, composed of several different corpora provided by the Linguistic Data Consortium. Although they did not mention the specific

corpora which were included, nor the type of text these corpora contained, it is probably safe to assume that not all of it was based on medical texts.

The ratios are quite similar in most cases, the only exception being the contrast relation, which appears less frequently in our medical corpus than in the mixed corpus used by Marcu and Echihabi. One important feature of our corpus is that it consists only of medical articles and case reports, which in most cases aim to dispense information about a disease or a patient in a very straightforward way, leaving out any information deemed inessential. Although contrast relations can obviously be used to dispense such information, the very nature of contrast also makes it ideal for adding more excitement to a story or other text. Hence a possible explanation for the disparity between the number of contrast relations found in our respective corpora, is that Marcu and Echihabi's corpus possibly contained more text of the type which would use contrast relations for the excitement factor (like journalism). Similarly, a possible explanation for why our corpus contains more elaboration relations than Marcu and Echihabi's corpus, is that elaboration is essential for conveying information straightforwardly without repetition, and could therefore be more important in our collection of medical articles.

Relation type	# found	Ratio	Original ratio
Contrast	4072	5.00	9.43
Cause	1670	2.05	2.16
Condition	951	1.17	2.93
Elaboration	4269	5.24	4.46

Table 4.2: Initial results of extracting relations from medical text

4.4 Manual error evaluation

We perform manual evaluation to determine the quality of the extracted relations and the most common source of errors. The evaluator is a native speaker of English, although she does not have experience in annotating medical texts. A total of 200 relations are chosen randomly for evaluation, 50 for every relation type. Each of these is checked against five categories of errors, described below. Table 4.3 shows the results of the evaluation for each type of relation.

- **Error 1 - Wrong relation:** This error means the wrong relation type is assigned to two text segments, when in reality they are connected by a different relation type or not connected at all. It also includes cases where two relations are possible, but where the one which is extracted is not the main one.
- **Error 2 - Wrong span:** Relations belong to this error category when the type of relation is correct, but the splitting of the two text segments is incorrect.
- **Error 3 - Mistake in preprocessing:** This category contains erroneous relations where the error is caused by a mistake in the preprocessing of

the text, either during the HTML parsing step or the sentence tokenization step.

- **Error 4 - Sentence is too complicated:** Since we make a very important simplification of discourse structure, in restricting each sentence to a maximum of 3 relations (one intra-sentential, and two inter-sentential), some sentences have a structure which is too complicated to be analyzed with this simplification. Hence, erroneous relations which are the result of a sentence being too complicated for our method fall into this category.
- **Error 5 - Error is unclear:** This category includes any cases where the relation does not seem quite right, but cannot be placed in any of the previous categories.

Table 4.3 shows that the condition relation is the most erroneous out of the relation types, and the most common error types are error 4 (sentences are too complicated) and error 3 (earlier mistakes in preprocessing). In total, 40.5% of the extracted relations contain some sort of error.

Relation type	Error 1	Error 2	Error 3	Error 4	Error 5	Total
Contrast	1	2	6	7	0	16
Cause	1	2	6	6	2	17
Condition	7	7	5	5	2	26
Elaboration	1	5	7	9	0	22

Table 4.3: Error analysis of initial results

Below are some examples of erroneous relations and how they are categorized, to show the main sources of error.

Example 4.4.1
Error 1 - Wrong relation
a. [This can be rather worrisome] [because patients may receive an inappropriate treatment if pathologists or physicians make an incorrect diagnosis of it, particularly in cases of MSS occurring in an uncommon site.]
b. [A two-tailed Mann Whitney t-test was used to determine] [if significant differences existed.]

Example 4.4.1 shows two relations which are marked as being erroneous due to the wrong relation having been assigned. The first two text segments, shown in Example 4.4.1.a, were marked during the extraction process as having a cause relation. Although this relation is not incorrect, the first text segment gives very little useful information. There is another relation present within this sentence, namely a condition relation marked by the word *if*, which presents much more useful information than the relation type cause in this case. Hence the condition relation would be preferable over the cause relation in this case.

The second pair of text segments, shown in Example 4.4.1.b, were marked as having a condition relation during extraction. Although one could possibly argue against marking this as an erroneous relation, we do not consider these cases to truly represent condition because it is unclear in the two segments which one is

the condition and which one is the result. This is one of the reasons why Table 4.3 shows so many errors of the first type for the condition relation; most of those errors are caused by sentences which contain *to determine if*, *to assess if*, or *to see if*.

Example 4.4.2
Error 2 - Wrong span
a. [Acquired myasthenia gravis (MG) is an autoimmune disorder of the neuromuscular junction in] [which patients experience fluctuating skeletal muscle weakness that often affects selected muscle groups preferentially.]
b. [Although the interactions between PCOS,] [OSA, and the cardiometabolic consequences are complex, a recent study has shown improvement in cardiometabolic profile after the successful treatment of OSA (10).]

Example 4.4.2.a shows an intra-sentential relation marked as elaboration during automatic extraction. The type of the relation in this case is correct, however the splitting of the sentence into two parts is off by one word. The last word of the first segment, *in*, should be included in the second segment in order for this relation to be correct. Example 4.4.2.b shows a relation where the span is off by more than one word. It is marked as a contrast relation by the extraction process, which is again correct, however splitting the sentence on the first comma according to our pattern caused the mistake in spans. The split should have been made between "... consequences are complex," and "a recent study...". Some might argue that this example belongs to the fourth category of errors (i.e. sentences which are too complicated), however we consider it to be the second category because a list is quite a common occurrence within sentences.

Example 4.4.3
Error 3 - Mistakes in preprocessing
a. [Tracheostomy is effective in severe or emergent cases.7 Respiratory stimulants such as caffeine and doxapram, commonly used for apnea of prematurity and respiratory depression after anesthesia, could be a future treatment option in babies with achondroplasia, due to the stimulation of breathing on the medullary respiratory centers and carotid bodies; however, they have not been evaluated for use in this patient population.12 Our patient underwent three-dimensional computerized tomography (CT) of the cervicomedullary junction without sedation instead of MRI with sedation,] [because of faster image acquisition time with CT than MRI, and the risks associated with sedating an infant patient with SDB in order to acquire MRI images.]
b. [Transient or low-affinity interactions could appear CSA/CSB independent] [if interactions are fixed by cross-linking as in ChIP experiments (Schwertman et al.]

Most of the errors in the third category, mistakes in preprocessing, are caused by the sentence tokenization process. Example 4.4.3 shows two cases which are common in our data. In the first case, the tokenizer is unable to handle citations placed directly after the sentence boundary, resulting in a concatenation of several sentences. In the second case, the abbreviation *et al.* causes a problem. This problem was also seen with abbreviations such as *e.g.* or *i.e.*, resulting in

sentences cut off halfway.

Example 4.4.4
Error 4 - Sentence is too complicated
a. [The explanation of this phenomenon is not evident: the complex diploid tumors cannot be regarded as aneuploid, even if the complex karyotype shows evident aneuploidy () [although at a more sensitive level).]
b. [For example, whereas vascular-specific targeting of Alk1 recapitulates the phenotype of HHT2 [29], vascular-specific targeting of TGF RII,] [which is the major type II receptor for TGF on EC, has no vascular phenotype [18].]

Finally, Example 4.4.4 shows two erroneous relations where the error occurs because the sentence is too complicated for our method to work. The first sentence shows typical use of parentheses for creating almost another sentence within a sentence. Our extraction process recognized the relation within the parentheses correctly, but because the part within parentheses is only a side-note, the relation does not hold with the entirety of the rest of the sentence. The second sentence does not have any side-notes hidden between parentheses, but it does consist of many different parts separated by commas. These different parts could easily be split into two separate, simpler sentences. In these cases it is very unlikely that our extraction method can find one relation which hold between the two main parts, because there are just too many sub-parts to consider.

We will not look at the errors belonging to the fifth category, since there were only four such errors in the 200 examples. Furthermore, the errors belonging to the last category are so different that there is no common problem to look at solving.

4.5 Improvements in the relation extraction process

Based on manual evaluation of the errors encountered in our initial results, we implemented and tested several ideas for improving relation extraction. The improvements are listed based on the error category they belong to.

Error 1 - Wrong relation

- Clearly the condition relation suffers the most from cases where an incorrect relation is assigned. This is almost always caused by phrases such as *to determine if*, *to assess if*, or *to see if*. Hence an idea for decreasing the number of these errors is to check if the word preceding *if* in a sentence is a verb. Upon further inspection of the relation examples, we restrict the word appearing before *if* to past tense verbs, past participle verbs, and non-verbs. This will approve relations like *[In these analyses, the whole family was excluded] [if a proband had T2DM (n = 2 families and 7 people).]*, but reject the cases we mention above.

Error 2 - Wrong span

- Although it is again the condition relation which suffers most from wrong span errors, we could find no clear pattern to these errors. On the other hand, the elaboration relation also has a lot of errors involving spans, and there is a pattern which can be used to formulate an improvement. Most of the errors involve the discourse marker *which*, because the pattern defined by Marcu and Echihabi splits the relation directly before this marker, but it is often accompanied by prepositions or other auxiliary words, for example *in which*, *for which*, or *at which*. Hence a possible improvement is to check the word appearing before the marker *which*, and to split the sentence one word earlier if this word is a preposition or a subordinating conjunction.

Error 3 - Mistake in preprocessing

- A lot of mistakes in preprocessing are caused by the sentence tokenizer mistaking an abbreviation for the end of a sentence. For this reason we remove the punctuation marks from abbreviations which commonly appear in scientific (and medical) documents: Fig., et al., e.g., etc., and i.e.

- Another type of error which involves the sentence tokenizer is when citations appear after the end of a sentence. Quite a few of the articles in the corpus use this style of citation, so to prevent this from disturbing the tokenization process, all citations of this type are removed from the texts.

Error 4 - Sentence is too complicated

- Many sentences in medical texts contain additional information in parentheses. Sometimes the parentheses are a complete sentence within themselves, which causes errors during relation extraction, especially when the text in parentheses contains one of the discourse markers. To solve this problem, any information between parentheses is removed before relation extraction occurs, and replaced after the algorithm has decided the relation type and the splitting point.

- Many of the sentences encountered in our corpus could easily be split into several simpler sentences. These types of sentences are often characterized by many phrases separated by commas. We perform a simple evaluation on sentences which are marked as being too complicated, compared to sentences which are marked as being correctly extracted. In this evaluation we count the number of phrases separated by commas and the average length of these phrases. More than 90% of the correctly extracted sentences contain less than 5 phrases separated by commas. When looking at the phrases which are marked as being too complicated, about 90% of the phrases are larger than 50 characters. Although it is desirable to perform further evaluation to determine the best values at which to reject relations, time constraints push us to use these values as cut-off points.

Although the implementation of the above improvements is a good starting point, it is important to remember that there is another experiment from which we can draw information. [Sporleder and Lascarides, 2008] provide a large list of 50 discourse markers used in their experiments, so it is worthwhile to try finding

patterns for the most popular markers on their list. As explained in Section 4.1, Sporleder and Lascarides use a different set of discourse relations than Marcu and Echihabi, but there is some comparison. The set of discourse relations used so far in this project is a generalization made by Marcu and Echihabi based on several discourse theories, including SDRT, which is the theory used by Sporleder and Lascarides. The former authors provide a table in their paper, detailing how relation types from different theories are categorized in the four relation types they defined (and which we use here). From this table it is possible to see that the relations Result and Explanation in SDRT are part of our Cause relation, and the Contrast relation in SDRT is equal to our own Contrast relation.

To determine which of the markers belonging to the Result, Explanation, and Contrast relations are best to add to our algorithm, we count how often each marker appears in our corpus. The most common markers are *however* (Contrast), *whereas* (Contrast), *(in|by) contrast* (Contrast), and *consequently* (Results). Proper extraction patterns are needed to minimize errors, and since these patterns are not provided by Sporleder and Lascarides, we created new patterns based on a manual inspection of the occurrences of these markers in the corpus. This results in a new pattern table, shown in Table 4.4.

CONTRAST
[BOS ... EOS] [BOS But ... EOS]
[BOS ...] [but ... EOS]
[BOS ...] [although ... EOS]
[BOS Although... ,] [... EOS]
[BOS ... EOS] [BOS However ... EOS]
[BOS Whereas... ,] [... EOS]
[BOS ...] [whereas ... EOS]
[BOS (In|By) contrast ... ,] [... EOS]
[BOS ... EOS] [BOS (In|By) contrast, ... EOS]
CAUSE-EXPLANATION-EVIDENCE
[BOS ...] [because ... EOS]
[BOS Because ... ,] [... EOS]
[BOS ... EOS] [BOS Thus, ... EOS]
[BOS ... EOS] [BOS Consequently ... EOS]
[BOS ...] [(and)(,) consequently ... EOS]
CONDITION
[BOS If... ,] [... EOS]
[BOS If...] [then ... EOS]
[BOS ...] [if ... EOS]
ELABORATION
[BOS ... EOS] [BOS... for example... EOS]
[BOS...] [which... ,]

Table 4.4: Final set of patterns used in experiment

For the marker *however*, there is only one pattern, to check if it occurs are the start of a sentence. The occurrence of *however* within a sentence is too complicated to replicate correctly in one or two patterns, especially to decide where

to split the sentence and whether it is an inter-sentential or intra-sentential relation. In contrast, the marker *whereas* is relatively easy and has similar patterns to the marker *although*. Again, for the marker *(in|by) contrast* we only check the beginning of a sentence because its occurrence in the middle of sentences is too complicated. However there are two different situations for this marker: when it appears at the beginning of a sentence followed directly by a comma, it forms an inter-sentential relation with the previous sentence, otherwise it is intra-sentential. Finally, the marker *consequently* can appear at the beginning of a sentence to form an inter-sentential relation. When it appears in the middle of a sentence, the pattern will only be accepted if the word *and*, or a comma, or both, appear in front of the marker. This does not cover all cases of the use of *consequently*, but it does reject many cases where consequently does not indicate the main relation within a sentence (i.e. when it appears in one of the phrases found in a complicated sentence).

After running the algorithm again with all of the improvements and the new markers, and doing another manual evaluation of 200 relations, we found that the first improvement listed above, which tries to decrease the number of wrong relation errors for the condition relation, is ineffective. Therefore it was scrapped before running the algorithm one last time to arrive at the results presented in Table 4.5. It shows the number of relations of each type found in our text, which consisted of 82,667 sentences.

Relation type	# found	Ratio	Initial ratio	Original ratio
Contrast	6545	7.92	5.0	9.43
Cause	1726	2.08	2.05	2.16
Condition	793	0.96	1.17	2.93
Elaboration	4181	5.06	5.24	4.46

Table 4.5: Final results of relation extraction

Again, the ratio was calculated by dividing the number of relations found by the total number of sentences, multiplied by 100. It can be compared with the ratio of our initial results, repeated in the third column, and the ratio of the results achieved by [Marcu and Echihabi, 2002] in the last column. Our ratios have improved for contrast and cause, due to the extra markers taken from [Sporleder and Lascarides, 2008]. The ratios for condition and elaboration did decrease a little, however this is compensated by the improvement in error rate of more than 10%. Table 4.6 shows the final error analysis results.

Relation type	Error 1	Error 2	Error 3	Error 4	Error 5	Total
Contrast	3	2	3	1	0	9
Cause	1	3	0	7	1	12
Condition	6	5	2	7	0	20
Elaboration	3	1	2	7	1	14

Table 4.6: Error analysis of final results

The new error analysis results show that the number of errors caused by mistakes in preprocessing has decreased significantly, and the elaboration features

less errors in span now that the word before *which* is being examined for its type. There are still quite a few errors involving sentences which are too complicated, so this requires more experimentation to fix. Ideally it would involve methods for simplifying the sentence based on its content, but this would require the availability of the semantics which we want to extract in the first place. So instead, a good start to tackling the issue would be to further analyze which features make the sentences too complicated and how these features can be detected. Finally, notice that the condition relation is the most problematic relation in both the initial and final results. A possible reason for this is that the algorithm really only has one discourse marker (*if*) and three patterns involving this pattern to check for. If it is not possible to find more markers for the condition relation, or to improve the results found with the available patterns, it might be better to choose a different discourse relation for future experiments. Indeed, [Sporleder and Lascarides, 2008] chose not to use the condition relation, they featured the continuation and explanation relations instead.

4.6 Final analysis of discourse relation extraction

The ratios found in Table 4.5 show that the relation data is very sparse. Relying solely on explicit discourse markers in a text is clearly not sufficient for extracting the majority of the semantics. To create a truly complete structure of a sentence requires background information and semantic knowledge of words, however there are possibilities for extending the current extraction capabilities. One such possibility is Discourse Relation Algebra [Roze, 2011], introduced in Section 2.1.4. It provides inference rules which can be used to generate a complete set discourse relations from an incomplete set, hence it would be perfect for reducing the sparsity of our data. However the method is not entirely finished, only a few rules have been defined so far, but it is a good option for improving our data in the future.

We managed to decrease the error rate in relation extraction from 40% down to 30%, but this still carries quite a lot of noise through to the next step of this project. In the future it would be worthwhile to attempt decreasing the error rate even further. The sentence tokenizer is a source of quite some noise, so experimenting with different preprocessing tools is one option for improvement. Devising other improvements similar to the ones we've already introduced requires more manual evaluation of the relations to build even more complex patterns. One final possibility is to combine the unsupervised methods with some of the supervised methods. For now, the results are promising enough to move on to the data mining stage.

5. Representation using FCA

To apply Formal Concept Analysis to relational discourse data it is necessary to define a formal context as a starting point, with a set of objects and a set of attributes (or patterns). By applying the CloseByOne algorithm to the formal context, one can generate the closed concepts for forming a concept lattice. And in the case of complex data which needs to be represented using the formalism of pattern structures, it is essential to define a proper similarity operator.

There are many different possibilities for converting our discourse relation data into such a formal context format with appropriate similarity operators, and each method can result in a very different concept lattice. Some concept lattices could provide useful knowledge to an expert, whereas others might be less suitable. Hence in this chapter we discuss and compare different representations of formal contexts, possible similarity operators, as well as external resources which could add value to the data. First, some basic choices regarding FCA representations are discussed in Section 5.1, followed by some more complex representations using pattern structures in Section 5.2. Section 5.3 describes external resources which we use to add additional semantic value to our data and to define an order relation between patterns. Finally, in Section 5.4 we talk about the algorithm which is used to generate the closed pattern concepts and the final lattice.

5.1 Representing discourse relation data in FCA

We start by discussing the options of using the basic FCA theory when representing discourse relations, because it is possible to imagine representations where our data is fit into a simple binary context. To illustrate how this could be done, we first introduce three relations from our corpus in Example 5.1.1. These three relations will be used throughout this chapter to give a better idea of what each approach could look like. Each relation has a name, for example bc49r3, which means that this relation is the third relation from article bc49 (the 49th article about breast cancer) in our corpus. The relations are displayed with their XML discourse annotation as they are stored in the corpus, such that the tag surrounding the entire relation indicates its type (CONT for contrast, CAUS for cause, ELAB for elaboration, and COND for condition), and within this there are two more tags to indicate how the sentence(s) are separated into two parts (PART1 and PART2).

Example 5.1.1
bc49r3*: <CONT><PART1>Using TFSEARCH (32), a web-based program that searches for transcription factor binding sites, an Nkx-2.5 binding site was found to be present when the major A allele was present, </PART1><PART2>but not when the minor G allele was.</PART2></CONT>*
bc49r8*: <ELAB><PART1>Phosphorylated SMAD2 and SMAD3, in association with SMAD4, form a complex </PART1><PART2>which accumulates in the nucleus and acts as transcription factors to regulate target genes.</PART2>*

</ELAB>
***bc50r4**: <CONT><PART1>On analyzing the data in rs353639 polymorphism with logistic regression, we found increased significance of both the genotype (P = 0.017, OR = 4.29) as well as allele (P=0.025, OR=3.34) with clinical tumour size when compared with the results of univariate analysis (Table 6).</PART1> <PART2>However, no significant association of both the polymorphisms was seen with treatment response to NACT.</PART2></CONT>*

These particular relations were chosen because they come from two articles which both discuss the role of particular genes in breast cancer cases, so there is a higher chance that the relations have something in common and can provide additional knowledge when combined. Both articles contain between 10-20 relations, but in order to keep the examples simple we chose three at random. The first two in Example 5.1.1a and 5.1.1b come from article bc49, where the first one is a contrast relation and the second is an elaboration relation. The third relation comes from article bc50 in the corpus and also belongs to the category of contrast.

So one option for applying the relation data to a binary context is shown in Table 5.1 where objects are the documents and attributes are the specific relations found in those documents (in this case limited to the three relations we specified). This is the most naïve and direct way of fitting the data into the binary context format. A document is related only to the relations which occur in its content. In this particular example the context is quite small, however one can imagine that such a method would quickly lead to contexts which are enormous in size, since most relations are unique and belong only to one document. For this reason, this type of context also doesn't provide much useful information, because there will likely be little to no overlap between documents (unless they contain exactly the same sentence). Finally, it would be up to the expert to read and interpret every relation in the resulting concept lattice, so it does not truly make use of the additional information which discourse provides.

	r3	r4	r8
bc49	X		X
bc50		X	

Table 5.1: Primitive context for discourse relations

One of the first choices which needs to be made when formulating a formal context is the definition of the objects. In Table 5.1 each object is an entire document, but one could just as easily define the objects to be paragraphs, sentences, or even words. The choice of objects is completely dependent on what it is that one wants to compare. If the goal is to compare various medical articles and select the relevant ones according to some criteria, then articles are the obvious choice for objects. However if a researcher were more interested in the structure of relations, then indeed it would be possible to use the relations themselves as objects. Since the aim of this project is simply to create a basis for the combination of FCA with discourse structure, we could work with either articles or relations as objects. For now we will use the latter in our examples.

The next important consideration to make when building a formal context is the definition of the attributes, which we defined as being whole relations in Table 5.1. In basic FCA the attributes are restricted to boolean values which can be represented in a binary table. In the case where we have a set of articles as objects, this means it is possible to use relations, words, relation categories, or other entities as attributes. However each of those options is either too general (there are only four relation categories) or too specific (almost every separate relation is unique by at least one word). Fortunately pattern structures make it possible to define more complex patterns to use as attributes.

5.2 Applying pattern structures

In order to determine how pattern structures can be applied to discourse data, it is important to realize exactly what the components of a relation are. Our relations contain three components: the type of the relation, the text of the left part of the relation, and the text of the right part of the relation. Hence the most direct application of pattern structures to our discourse data would be to have three types of patterns corresponding to the three parts of our relations. This format is shown in Table 5.2, using the three relations we introduced before.

Document	Type	Left Part	Right Part
bc49r3	CONT	Using TFSEARCH (32), a web-based program that searches for transcription factor binding sites, an Nkx-2.5 binding site was found to be present when the major A allele was present,	but not when the minor G allele was.
bc49r8	ELAB	Phosphorylated SMAD2 and SMAD3, in association with SMAD4, form a complex	which accumulates in the nucleus and acts as transcription factors to regulate target genes.
bc50r4	CONT	On analyzing the data in rs353639 polymorphism with logistic regression, we found increased significance of both the genotype (P=0.017, OR=4.29) as well as allele (P=0.025, OR=3.34) with clinical tumour size when compared with the results of univariate analysis (Table 6).	However, no significant association of both the polymorphisms was seen with treatment response to NACT.

Table 5.2: Direct application of pattern structures to discourse relation data

In order to generate closed concepts it is necessary to define similarity operators in such a way that any two rows can be compared, which is where we encounter the first problem with this representation. There is no obvious order relation between the semantic types of discourse relations; there is no overlap and there is no specialization/generalization relation between them. Nor is there a logical clustering between the four categories of contrast, condition, elaboration, and cause. In the absence of a similarity operator which can be applied to the second column of Table 5.2, it becomes necessary to redefine the formal context. A solution to this problem is to separate the four semantic categories in the formal context, resulting in a total of eight pattern columns, two for each semantic type. Since our example only features two contrast relations and one elaboration relation, we show only four of those columns in Table 5.3.

Document	CONT Left	CONT Right	ELAB Left	ELAB Right
bc49r3	Using TFSEARCH (32), a web-based program that searches for transcription factor binding sites, an Nkx-2.5 binding site was found to be present when the major A allele was present,	but not when the minor G allele was.		
bc49r8			Phosphorylated SMAD2 and SMAD3, in association with SMAD4, form a complex	which accumulates in the nucleus and acts as transcription factors to regulate target genes.
bc50r4	On analyzing the data in rs353639 polymorphism with logistic regression, we found increased significance of both the genotype (P=0.017, OR=4.29) as well as allele (P=0.025, OR=3.34) with clinical tumour size when compared with the results of univariate analysis (Table 6).	However, no significant association of both the polymorphisms was seen with treatment response to NACT.		

Table 5.3: Pattern structure with separated semantic categories

By separating the semantic categories of discourse relations, we have created a situation which requires the definition of only one similarity operator to handle the comparison of the strings which form the parts of the relations. Statistical methods are one possibility for comparing two strings, such as the Levenshtein distance or the Jaccard index. The former is an edit distance, meaning it compares two strings by calculating how many changes are needed to turn one string into another. Changes which can be made include insertions, deletions, and substitutions. So as an example consider the two words *dog* and *logs*; the Levenshtein distance between these two words is 2, since changing one word into the other requires one substitution and one insertion. On the other hand, the Jaccard index is a metric for calculating the similarity of sets. When applied to strings, it views the strings as sets of tokens and proceeds by calculating the similarity between two sets of tokens. Similarity is defined as the size of the intersection of the two sets, divided by the size of the union of both sets. Although statistical similarity

measures have been applied to text mining before, they are not ideal for applying to our data. A quick calculation on our corpus shows that the average Levenshtein distance between sentences is above 150 changes. Another possibility is to abstract over the sentences by first applying part-of-speech tagging and then calculating similarity with a string metric, but this would cause loss of information. However, the string metric method in combination with part-of-speech tagging could be kept in mind for linguistic research, since it could provide information about how relations are structured.

Since discourse relations rely on the meaning of sentences, it would be logical to apply a similarity operator which is based on semantics. Ideally we would build a logical formula of the sentence based on one of the theories introduced in Section 2.1, such as DRT or SDRT. However, as mentioned before, we lack the semantic tools to properly build these formulas regarding both semantics and background knowledge. Attempting to create the formulas anyway with the limited tools available would result in a lot of noise in our data. Furthermore, such formulas would have to be compared to create an order relation between them, and this cannot be done through formal logic alone. They contain complex elements which require further background knowledge to understand and compare. Although building the complete set of formulas is not an option at this time, we can add some basic background knowledge to the text by using an ontology. This requires the application of external resources, like a medical thesaurus.

5.3 Adding information through external resources

In Chapter 2 we described how [Coulet et al., 2013] use the NCI Thesaurus to extract sets of concepts from documents. The tree structure of the ontology provides an order relation between the concepts, and the similarity operator was defined to be the convex hull. NCI Thesaurus is one of several ontologies incorporated in the Unified Medical Language System (UMLS) MetaThesaurus [National Library of Medicine, 2009]. We describe the UMLS MetaThesaurus and Semantic Network in Section 3.3. The elements of the UMLS MetaThesaurus are usually referred to as concepts, but we will call them terms to make it clear that UMLS concepts are not the same as formal concepts of FCA. To extract sets of UMLS terms from textual data, we use the MetaMap tool [National Library of Medicine, 2013] which was developed specifically for that purpose. Example 5.3.1 shows the human-readable output which MetaMap produces for the phrase *human plasma*. It finds a total of 4 MetaThesaurus terms which can correspond to the phrase at varying confidence levels. In our experiments we use the XML output for easier processing, but the results are the same.

Example 5.3.1
Phrase: "human plasma"
Meta Candidates (Total=4; Excluded=0; Pruned=0; Remaining=4)
1000 human plasma [Pharmacologic Substance]
861 Plasma [Body Substance]
861 Human (Homo sapiens) [Human]

861 Plasma, NOS (not otherwise specified) [Body Substance]
Meta Mapping (1000):
1000 human plasma [Pharmacologic Substance]

In order to create the set of terms which corresponds to the text of a relation, we select only the top candidate for each phrase identified by MetaMap. These sets then replace the textual data in the formal context, as shown in Table 5.4 for the three chosen relations. Although these sets do abstract over the text somewhat, they clearly provide much more information than if we were to apply a statistical string metric like Levenshtein distance. One disadvantage is that MetaMap does overgenerate quite heavily; the sets shown in Table 5.4 are all quite large. On top of that there is quite a bit of noise: the relation bc50r4 mentions a table in the sense of a collection of information, but MetaMap links it to the term *Tablefurniture*. It is possible to tweak the MetaMap results through the use of a confidence threshold, which would at least decrease the size of the sets, even though it would not prevent the occurrence of noise. For future work it could be worth spending time on an empirical evaluation to determine the best threshold for our corpus.

Document	CONT Left	CONT Right	ELAB Left	ELAB Right
bc49r3	{Useof; Basisconceptualentity; Programframeworkofgoals; searchEntityNameUse; BindingSites; Present; To; Present; Major; Alleles; Present}	{Negation; Alleles}	{}	{}
bc49r8	{}	{}	{SMAD2gene; SMAD3gene; Mentalassociation; SMAD4gene; Qualitativeform; Complex}	{CellNucleus; TRANSCRIPTIONFACTOR; CandidateDiseaseGene}
bc50r4	{Data; LogisticRegression; Present; StatisticalSignificance; Genotype; Pbloodgroupantibodies; Alleles; Pbloodgroupantibodies; Tumorsize; univariatestatistics; Tablefurniture}	{Mentalassociation; GeneticPolymorphism; therapeuticaspects; SLC13A5gene}	{}	{}

Table 5.4: Pattern structure with UMLS MetaThesaurus terms

With a collection of MetaThesaurus terms populating our formal context in place of textual data, it becomes much easier to define a similarity operator. In fact we can use the Semantic Network[1] as an ontology to order terms, similar to how [Coulet et al., 2013] use the structure of the NCI Thesaurus. Every MetaThesaurus term links to at least one semantic type in the Semantic Network. To define an order relation we use only the most general *is_a* relation between terms. For simplicity we link every term with only one semantic type in the network, namely with the type which MetaMap lists first for that term. By linking the terms to the Semantic Network, we have a tree-like structure which allows us to define the similarity operation to be the convex hull.

[1] `http://semanticnetwork.nlm.nih.gov/`

5.4 Generating closed pattern concepts

Having defined a formal context in the form of Table 5.4, it is possible to apply the CloseByOne algorithm to generate the closed pattern concepts. In fact, notice that our formal context is similar to the formal context in Table 2.2 which shows an adapted example from [Coulet et al., 2013]. Furthermore we define the same similarity operator, namely the convex hull. We showed an example of this similarity operator in Section 2.2.2, but we will now formally define it. The convex hull $Conv(\{c_1, c_2\})$ of the set of two terms c_1 and c_2 will be a set of terms $\{x_1, x_2, ..., x_n\}$ such that:

$$x_i \leq lcs(\{c_1, c_2\})$$

$$(x_i \geq c_1 \text{ and } x_i \wedge c_1 \equiv c_1) or (x_i \geq c_2 \text{ and } x_i \wedge c_2 \equiv c_2)$$

$$x_i \not\equiv \top$$

where $\top$ refers to the top term of the ontology, and *lcs* finds the least common subsumer of two terms (the most specific term which subsumes both smaller terms). Furthermore, the convex hull can be recursively applied to a set of terms $C_p = \{c_1, c_2, ..., c_p\}$:

$$\forall p \in \mathbb{N}, Conv(C_p) = Conv(\{Conv(C_{p-1}), c_p\})$$

Since we use the same similarity operator as [Coulet et al., 2013], we can use the modified CloseByOne algorithm defined by the same authors which we repeated in Algorithms 1 and 2. The authors fortunately provide a java implementation of the algorithm[2]. It reads the ontology of the NCI Thesaurus from an OWL format. We replace this ontology with our own, which contains the complete structure of the Semantic Network as well as all 12,416 terms which MetaMap found in our corpus. It was converted to OWL format compatible with the implementation of the CloseByOne algorithm using the Protégé tool [Tudorache et al., 2013]. There is some discussion about whether or not the OWL format is suitable for modeling the Semantic Network, due to ambiguities in notation and a few other issues [Kashyap and Borgida, 2003], but a simple representation of the network using only basic relationships and terms can be represented in OWL format without problems.

Running the algorithm on the formal context in Table 5.4 results in 8 closed concepts. Figure 5.1 shows the structure of the resulting concept lattice. Since the table which shows the formal context leaves out the four columns relating to the condition and cause relations, we have also left out those categories in the lattice image. But keep in mind that this method can work with all four types of relations at once; every intent would have 8 sets of terms in total, to account for the left and right text segments of all four relation types. Notice that the concept lattice is a complete lattice; in fact it is a power set. This will be the case with any relations compared through our pattern structure model, because the ontology is a single tree structure, meaning that there is a path to the top starting from any term. If we want to prevent this, we would need to choose a different similarity operator.

[2] `https://github.com/coulet/OntologyPatternIcfca`

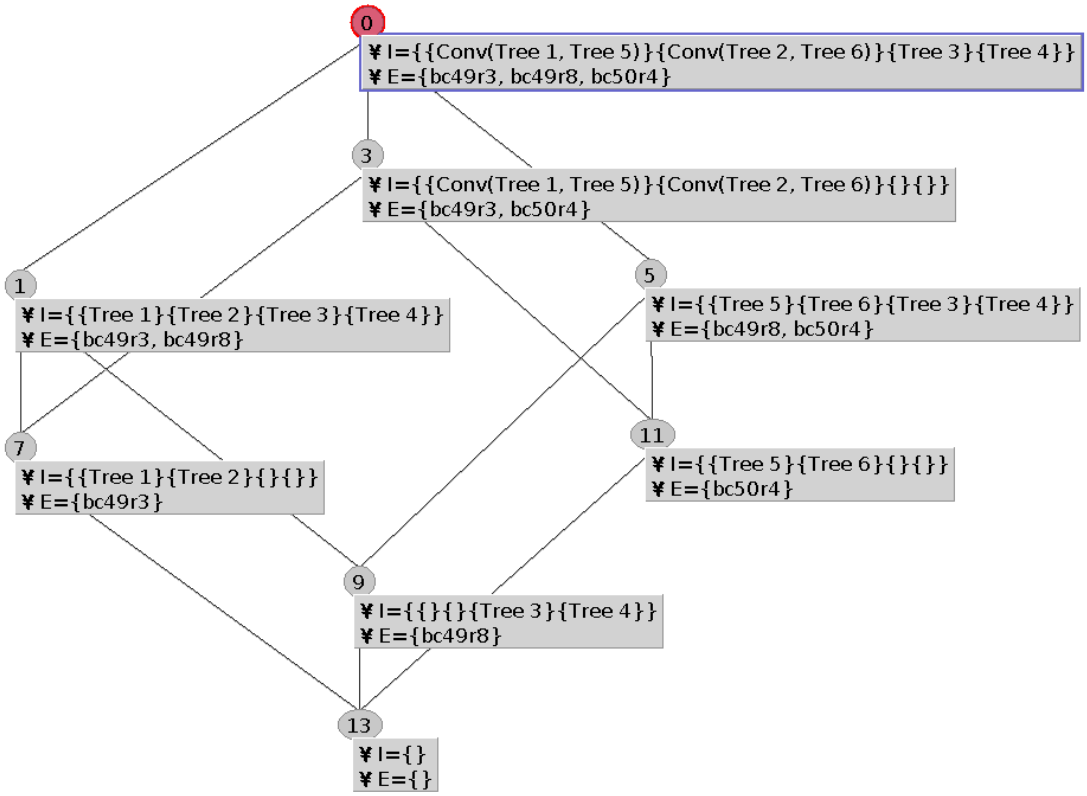


Figure 5.1: Concept lattice calculated from the formal context in Table 5.4

The full intents of the concepts are too long to present in the figure, therefore they have been given names corresponding to their representations shown below. However we will show one example of a full intent to illustrate how the lattice should be interpreted. Example 5.4.1 shows the full intent of concept 3 in the lattice. The intent consists of four sets contained in curly brackets, where the first two sets respectively represent the left and right text segments of all contrast relations in the extent of the concept, and the last two sets represent the left and right text segments of all elaboration relations in the extent of the concept. Since concept 3 in the lattice has an extent with relations bc49r3 and bc50r4, which are both contrast relations, the last two sets of its intent are empty. Example 5.4.1 shows that the first two intents are both large sets containing terms from the UMLS MetaThesaurus and the Semantic Network.

Let us consider the very first set in Example 5.4.1. It represents the left text segments of both contrast relations bc49r3 and bc50r4, which we converted into a set of terms using MetaMap, the results of which are shown in the first column of Table 5.4. To combine the two relations into a formal concept, the algorithm computes the convex hull of the two sets of terms. Again in this example we encounter the problem that MetaMap over-generates to produce very large sets of terms as pattern descriptions, which result in equally large convex hulls. This convex hull contains all of the terms from both sets, the least common subsumer of all of those terms, and every term in between, which results in the first set in Example 5.4.1. Similarly, the second set shown in the example is the convex hull of the original sets of terms corresponding to the right text segments of contrast

relations bc49r3 and bc50r4 (see the second column in Table 5.4).

Example 5.4.1
{*Activity, Alleles, Amino_Acid,_Peptide,_or_Protein, Anatomical_Structure, Basisconceptualentity, BindingSites, Biologically_Active_Substance, Chemical, Chemical_Viewed_Functionally, Chemical_Viewed_Structurally, Conceptual_Entity, Data, Entity, Event, Fully_Formed_Anatomical_Structure, Functional_Concept, Gene_or_Genome, Genotype, Idea_or_Concept, Intellectual_Product, LogisticRegression, Major, Manufactured_Object, Occupational_Activity, Organic_Chemical, Organism_Attribute, Pbloodgroupantibodies, Physical_Object, Present, Programframeworkofgoals, Qualitative_Concept, Quantitative_Concept, Receptor, Research_Activity, Spatial_Concept, StatisticalSignificance, Substance, Tablefurniture, To, Tumorsize, Useof, searchEntityNameUse, univariatestatistics*}
{*Alleles, Anatomical_Structure, Biologic_Function, Conceptual_Entity, Entity, Event, Fully_Formed_Anatomical_Structure, Functional_Concept, Gene_or_Genome, GeneticPolymorphism, Genetic_Function, Idea_or_Concept, Mental_Process, Mentalassociation, Molecular_Function, Natural_Phenomenon_or_Process, Negation, Organism_Function, Phenomenon_or_Process, Physical_Object, Physiologic_Function, SLC13A5gene, therapeuticaspects*}
{}
{}

Since the ontology is structured as a tree, every set of terms can also be represented by a tree structure. Such a representation is much more intuitive than looking at a large set of terms. Therefore we will repeat the same example of concept 3 from the lattice, only now we will illustrate it with the tree structures. Each tree shows a convex hull which is the entire intent of a concept. It is important to understand that the colored terms (blue for bc49r3 and purple for bc50r4) are the original MetaThesaurus terms which describe the left textual segments of the relations. All of the other uncolored terms are simple part of the hierarchy computed by taking the convex hull of the original set of terms. The representations do not show the top element of the ontology, so the two most general semantic types below the top are Entity and Event. So the first two representations shown below are that of Tree 1, which is the left part of the contrast relation bc49r3, and Tree 5, which is the left part of the contrast relation bc50r4.

Tree 1 corresponds to the following text: *Using TFSEARCH (32), a web-based program that searches for transcription factor binding sites, an Nkx-2.5 binding site was found to be present when the major A allele was present,*

Tree1

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Functional_Concept
            Basisconceptualentity
            Use
         Qualitative_Concept
            Major
            To
         Quantitative_Concept
            Present
      Intellectual_Product
         searchEntityNameUse
      Programframeworkofgoals
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               Alleles
      Substance
         Chemical
            Chemically_Viewed_Functionally
               Biologically_Active_Substance
                  Receptor
                     BindingSites
```

Tree 5 corresponds to the following text: *On analyzing the data in rs353639 polymorphism with logistic regression, we found increased significance of both the genotype (P=0.017, OR=4.29) as well as allele (P=0.025, OR=3.34) with clinical tumour size when compared with the results of univariate analysis (Table 6).*

Tree 5

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Data
         Quantitative_Concept
            Present
            StatisticalSignificance
         Spatial_Concept
            Tumorsize
   Organism_Attribute
      Genotype
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               Alleles
      Manufactured_Object
         Tablefurniture
      Substance
         Chemical
            Chemical_Viewed_Structurally
               Organic_Chemical
```

```
                    Amino_Acid,_Peptide,_or_Protein
                       PBloodgroupantibodies
Event
   Activity
      Occuptional_Activity
         Research_Activity
            LogisticRegression
            univariatestatistics
```

The third representation is the convex hull formed from the first two representations $Conv(Tree1, Tree5)$, which is equivalent to the first two sets of terms shown in Example 5.4.1. When applying the convex hull operator to trees, we consider both trees to form a set of all the terms they contain. So the convex hull contains all of the original terms describing the textual segments, colored blue and purple. In this case, the least common subsumer is the top element because it is the only element which subsumes all of the colored terms (its is not shown in the representation, but it connects the current top elements Entity and Event). The black terms are all of the terms between the top element and each colored term. Hence all of the terms together is the result of taking the convex hull of the above two trees representations.

This representation shows the convex hull of the two previous convex hulls.

Conv(Tree1, Tree5)

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Data
         Functional_Concept
            Basisconceptualentity
            Use
         Qualitative_Concept
            Major
            To
         Quantitative_Concept
            Present
            StatisticalSignificance
         Spatial_Concept
            Tumorsize
      Intellectual_Product
         searchEntityNameUse
      Programframeworkofgoals
   Organism_Attribute
      Genotype
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               Alleles
      Manufactured_Object
         Tablefurniture
      Substance
         Chemical
            Chemically_Viewed_Functionally
               Biologically_Active_Substance
                  Receptor
                     BindingSites
```

```
            Chemical_Viewed_Structurally
               Organic_Chemical
                  Amino_Acid,_Peptide,_or_Protein
                     PBloodgroupantibodies
Event
   Activity
      Occuptional_Activity
         Research_Activity
            LogisticRegression
            univariatestatistics
```

It is important to explain one more aspect of how the lattice should be read. Normal concept lattices, like the one shown in Figure 2.3, have a large extent at the top element with a minimum intent, and a minimum extent at the bottom concept with a very large intent. Hence the further up the lattice a concept is placed, the more general its intent description is. In Figure 2.3, the top concept has extent $\{asthma, breastCancer, lungCancer\}$ and intent $\{foundInAdults\}$. Since every disease in that particular formal context has the attribute *foundInAdults*, this is the least specific description of a disease. At first glance, our lattice in Figure 5.1 looks completely different, with the top element having both large extent and large intent whereas the bottom element is empty on both. However it does adhere to the same specialization/generalization structure as the simpler concept lattice. When examining the final lattice, one must keep in mind that in this case a larger intent is a more general description than a smaller intent due to our similarity operator. It should be clear that a small convex hull gives a much more specific description than a large convex hull.

The concept lattice built for the three relations we chose to use as an example shows a clear representation of the possible relationships and combinations. It forms a promising visualization tool. Unfortunately we were unable to run the algorithm on the full corpus. The large number of terms which MetaMap assigns to every text segment means the algorithm takes about two hours to run for just one article. This time varies a lot depending on the complexity of the relations as well as the number of relations which the article contains. An even bigger problem is that it cannot handle the bigger articles, or indeed more than one complete article, because the algorithm will result in out of memory errors. It ran on a machine with an Intel® Core™i7-3540M processor and 16GB memory. So for future experiments it would be crucial to run the algorithm on a cluster in order to process the bigger inputs.

However it would also be a good idea to look into decreasing the number of MetaMap terms associated with a text segment, since this would greatly speed up the CloseByOne algorithm. This could be done through simplification of the sentences, but that might lead to loss of information, so a better method would be to empirically decide the best MetaMap threshold for producing useful results on a smaller scale. Limiting the size of the ontology could also speed up the algorithm during its calculation of convex hulls, but since the UMLS Semantic Network is already quite small this probably would not have a great impact in our case.

Despite being unable to run the experiments using our full corpus of 600

articles, we believe the results show great promise. It is difficult for us to interpret the lattices from a medical perspective, since we are not medical experts, so we cannot tell if the convex hull allows for the extraction of new medical knowledge. However, once one understands how to read the lattice in Figure 5.1, it becomes easy to see how the different relations (and by extension the different documents) interact. The intents of the formal concepts can be difficult to interpret in the form of sets, but by transforming the visualization into the form of trees, the intents become easy to understand and interpret at a glance. Such tree structures provide important hierarchical semantic knowledge, which can be very meaningful for experts who know the domain well.

In conclusion, we have shown that combining FCA with discourse structures and additional ontology information is possible and promising. It would be worthwhile to have the data and the method examined by a medical expert to determine how this method can best be applied to document summarization or document selection on medical articles. We encountered several problems during the process of generating the lattices, but with more research we believe these problems can be mostly resolved and the lattices can become a visualization tool for experts to use.

6. Conclusion

As far as we know, this is the first attempt at combining Formal Concept Analysis with discourse-annotated data. We believe that we have shown the merit of performing more research on these types of 'deep' text mining methods, since discourse can provide a lot of additional information which an analysis focusing on smaller unit of text cannot provide. In the medical field especially, there is a wealth of articles from which one can extract causal, temporal, and other discourse relations. Advancements in the fields of semantics, and in applying background knowledge to text, would allow for an even more robust extraction of discourse in the form of logical formulas.

In order to perform this research we have created a corpus of 600 full-text medical articles, containing case reports, treatment and drug evaluations, demographic studies, and historical accounts of disease treatment, with automatic annotation of discourse relations. In the future we hope that the annotations can be expanded, either through manual annotation, through the use of a discourse algebra with inference rules, or through improvements in our automatic extraction process. We were able to add additional semantic information to the medical text through the use of a medical thesaurus, as well as defining an order relation on those segments through an ontology. The resulting lattice structure provides a clear picture of the possible combinations of discourse relations. And combining the lattice with visualizations of the intents in the form of tree structures makes the whole output a lot easier to interpret. In our case we focused on the medical domain with a corpus of medical articles, but it might be interesting to try applying this in different domains, since discourse occurs in every type of text. Either way, it would be beneficial to eventually include an expert on the domain of choice, to give input on the modeling process.

Of course we did encounter problems throughout the process of this research. Many improvements are possible in both the linguistic side of this project, regarding the extraction of discourse elements, as well as the data mining side, with the application of external resources and the modeling of our data using pattern structures.

The biggest problem regarding the extraction of discourse structure is that we would ideally like to have a complete structure for every document, according to one of the theories described in Section 2.1. However this requires a very good semantic interpreter as well as background knowledge about the world, both of which are still difficult to create and model. Any advances in the practical side of discourse extraction could greatly benefit the information modeled by our method. The idea of extracting discourse relations between two text segments is much simpler than extracting an entire tree or graph structure, but it does provide a start to working with discourse in textual data. Due to a lack of annotated training data, especially in the medical domain, the unsupervised methods currently work equally as well as supervised methods. By combining the methods and key words used in two different works, we extract a total of 13,245 relations. However, considering that these relations were extracted from

a total of 82,667 sentences, it becomes clear that the annotations resulting from our experiments are still very sparse. Improving the density of annotations could be done by adding more key words to the current set used for extraction, or by possibly combining the unsupervised method with a supervised option. There are also some promising developments in building a discourse relation algebra, which could be used to complete the annotations [Roze, 2011].

Another problem is the degree of noise, regarding relations which are categorized incorrectly, which a random evaluation of 200 relations showed to be about a 30% error rate. Although we did manage to decrease the error rate from the original 40% through simple adjustments to the patterns and the algorithm in general, it would be worthwhile to invest more time in decreasing the error rate. A key problem is that the sentence tokenizer produces quite a bit of noise, which perpetuates into noisy discourse relation extraction. So trying out different processing tools could help improve the results and prevent a lot of noise from being forwarded into the rest of the process.

The process of applying pattern structures to the data gained from discourse relation extraction also has room for improvement. One issue is the large number of terms which MetaMap assigns to a piece of text, resulting in very large descriptions and concept intents which are difficult to interpret. It also causes the CloseByOne algorithm to have a very long running time and requires a lot of memory. This could be solved by investing some time in an empirical evaluation of the threshold value used in MetaMap, to determine the best value for our type of data. There might also be options for simplifying the textual data before running it through MetaMap, such as removing all data between parentheses, which is usually not the core information of a sentence. Despite these issues, applying the UMLS MetaThesaurus ontology to the data adds useful information about the semantics of the data, as opposed to the limited information which statistical methods can provide. However as mentioned before, the statistical methods could be used to apply this method to the linguistic domain, for modeling the structures of discourse relations. In that case the set of objects would again be the relations, and the patterns could involve string metrics based on part-of-speech tags to see if there is any new knowledge to gain about the linguistic features of relations.

Currently the set of objects in our formal context consists of individual relations, so another interesting extension to this process would be to apply it to an object set consisting of whole documents. This would add an extra level on top of the current model, since one would need to group the current patterns into sets. A document contains multiple relations, each of which we can describe as a pattern. It would be necessary to define another similarity operator to compare the sets of patterns corresponding to documents. However once that is defined, it would be possible to create a concept lattice for comparing articles instead of individual relations. In this project we show just one way to model discourse-annotated data in a formal context, but there are many possibilities for other models. Involving a domain expert in the modeling process could produce further improvements.

Since each step of our experiments experiences some problems, there is quite a build-up of noise by the end of the process. The sentence tokenizer introduces

noise, which causes more errors in the extraction of discourse relations, on top of which MetaMap adds a little more noise. It is essential to reduce the amount of noise passed forward by the process in order for this method to effectively create a platform for experts to discover new knowledge from text. However we have shown that the process creates an informative lattice despite the problems, and there are many options for improving the modeling process at different stages. We believe that performing data mining on complex linguistic structures is a very promising field.

Bibliography

Stergos D Afantenos, Nicholas Asher, Farah Benamara, Myriam Bras, Cécile Fabre, Mai Ho-Dac, Anne Le Draoulec, Philippe Muller, Marie-Paule Péry-Woodley, Laurent Prévot, et al. An empirical resource for discovering cognitive principles of discourse organisation: the ANNODIS corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2727–2734, Istanbul, Turkey, 2012. European Language Resources Association (ELRA).

Maxime Amblard and Sylvain Pogodalla. Modeling the Dynamic Effects of Discourse: Principles and Frameworks. In Manuel Rebuschi, Martine Batt, Gerhard Heinzmann, Franck Lihoreau, Michel Musiol, and Alain Trognon, editors, *Dialogue, Rationality, and Formalism*, volume 3 of *Logic, Argumentation & Reasoning*. Springer, 2014. URL `http://hal.inria.fr/hal-00737765`.

Nicholas Asher and Alex Lascarides. *Logics of conversation*. Cambridge University Press, 2003.

Nicholas Asher and Sylvain Pogodalla. SDRT and Continuation Semantics. In Takashi Onada, Daisuke Bekki, and Eric McCready, editors, *New Frontiers in Artificial Intelligence*, volume 6797 of *Lecture Notes in Computer Science*, pages 3–15. Springer Berlin Heidelberg, 2011. URL `http://dx.doi.org/10.1007/978-3-642-25655-4_2`.

Jason Baldridge and Alex Lascarides. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 96–103. Association for Computational Linguistics, 2005.

Lynn Carlson, Mary Ellen Okurowski, Daniel Marcu, Linguistic Data Consortium, et al. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania, 2002.

Claudio Carpineto and Giovanni Romano. Using concept lattices for text retrieval and mining. In Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors, *Formal Concept Analysis*, volume 3626 of *Lecture Notes in Computer Science*, pages 161–179. Springer Berlin Heidelberg, 2005. URL `http://dx.doi.org/10.1007/11528784_9`.

Adrien Coulet, Florent Domenach, Mehdi Kaytoue, and Amedeo Napoli. Using Pattern Structures for Analyzing Ontology-Based Annotations of Biomedical Data. In *Formal Concept Analysis*, volume 7880 of *Lecture Notes in Computer Science*, pages 76–91. Springer Berlin Heidelberg, 2013.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

Bernhard Ganter and Sergei O Kuznetsov. Pattern structures and their projections. In HarryS. Delugach and Gerd Stumme, editors, *Conceptual Structures: Broadening the Base*, volume 2120 of *Lecture Notes in Computer Science*,

pages 129–142. Springer Berlin Heidelberg, 2001. URL http://dx.doi.org/10.1007/3-540-44583-8_10.

Bernhard Ganter, Rudolf Wille, and Cornelia Franzke. *Formal concept analysis: mathematical foundations*. Springer-Verlag New York, Inc., 1997.

Jeroen Groenendijk and Martin Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100, 1991. URL http://dx.doi.org/10.1007/BF00628304.

Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A Brief Survey of Text Mining. In *Ldv Forum*, volume 20, pages 19–62, 2005.

Vipul Kashyap and Alex Borgida. Representing the UMLS® semantic network using OWL. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web-ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 2003. URL http://dx.doi.org/10.1007/978-3-540-39718-2_1.

Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sébastien Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989 – 2001, 2011. URL http://dx.doi.org/10.1016/j.ins.2010.07.007.

Sergei O Kuznetsov. A fast algorithm for computing all intersections of objects in a finite semi-lattice. *Automatic documentation and Mathematical linguistics*, 27(5):11–21, 1993.

Sergei O Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *Principles of Data Mining and Knowledge Discovery*, volume 1704 of *Lecture Notes in Computer Science*, pages 384–391. Springer Berlin Heidelberg, 1999. URL http://dx.doi.org/10.1007/978-3-540-48247-5_47.

Daniel Marcu and Abdessamad Echihabi. An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 368–375. Association for Computational Linguistics, 2002. URL http://dx.doi.org/10.3115/1073083.1073145.

Prem Melville, Wojciech Gryc, and Richard D. Lawrence. Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1275–1284. ACM, 2009.

Richard Montague. English as a Formal Language. In Richmond H. Thomason, editor, *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, 1974a.

Richard Montague. The Proper Treatment of Quantification in Ordinary English. In Richmond H. Thomason, editor, *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, 1974b.

Richard Montague. Universal Grammar. In Richmond H. Thomason, editor, *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, 1974c.

Philippe Muller, Stergos Afantenos, Pascal Denis, Nicholas Asher, et al. Constrained decoding for text-level discourse parsing. In *COLING-24th International Conference on Computational Linguistics*, 2012.

National Library of Medicine. *UMLS® Reference Manual*. Bethesda (MD): National Library of Medicine (US), 2009. URL `http://www.ncbi.nlm.nih.gov/books/NBK9676/`.

National Library of Medicine. MetaMap - A Tool for Recognizing UMLS Concepts in Text, 2013. URL `http://metamap.nlm.nih.gov/`. Accessed in May 2014.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper Manning. LinGO Redwoods. A Rich and Dynamic Treebank for HPSG. In *In Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, 2002.

Uta Priss. Linguistic applications of formal concept analysis. In Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors, *Formal Concept Analysis*, volume 3626 of *Lecture Notes in Computer Science*, pages 149–160. Springer Berlin Heidelberg, 2005.

Charlotte Roze. Towards a Discourse Relation Algebra for Comparing Discourse Structures. In *Proceedings of Constraints In Discourse (CID 2011)*, pages 1–7, 2011. URL `http://hal.archives-ouvertes.fr/hal-00655825`.

Radu Soricut and Daniel Marcu. Sentence Level Discourse Parsing Using Syntactic and Lexical Information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 149–156. Association for Computational Linguistics, 2003. URL `http://dx.doi.org/10.3115/1073445.1073475`.

Caroline Sporleder. Lexical Models to Identify Unmarked Discourse Relations: Does WordNet help? *Journal for Language Technology and Computational Linguistics*, 24:20–32, 12 2008.

Caroline Sporleder and Alex Lascarides. Using automatically labelled examples to classify rhetorical relations: an assessment. *Natural Language Engineering*, 14:369–416, 7 2008. URL `http://journals.cambridge.org/article_S1351324906004451`.

Don R Swanson. Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association*, 78(1):29, 1990.

Tania Tudorache, Csongor Nyulas, Natalya F Noy, and Mark A Musen. Webprotégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, 4(1):89–99, 2013.

Jan van Eijck and Hans Kamp. Representing Discourse in Context. In *Handbook of Logic and Language*, pages 179–237. Elsevier, Amsterdam, 1997.

Ben Wellner, James Pustejovsky, Catherine Havasi, Anna Rumshisky, and Roser Sauri. Classification of discourse coherence relations: An exploratory study using multiple knowledge sources. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 117–125. Association for Computational Linguistics, 2009.

Florian Wolf, Edward Gibson, Amy Fisher, and Meredith Knight. Discourse Graphbank. *Linguistic Data Consortium*, 2004.

List of Tables

List of Abbreviations

BOS	beginning of sentence
CAUS	cause
COND	condition
CONT	contrast
DPL	Dynamic Predicate Logic
DRS	Discourse Representation Structure
DRT	Discourse Representation Theory
ELAB	elaboration
EOS	end of sentence
FCA	Formal Concept Analysis
HHT	hereditary hemorrhagic telangiectasia
KDD	Knowledge Discovery in Databases
NCI	National Cancer Institute
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OWL	Web Ontology Language
RCA	Relational Concept Analysis
RFC	Right Frontier Constraint
RST	Rhetorical Structure Theory
SDRT	Segmented Discourse Representation Theory
UMLS	Unified Medical Language System

Attachments

Intents of closed pattern concepts

The following representations are the remaining convex hull representations for the intents of concepts which contain only one relation in the extent. Intents of concepts with larger extents can be built by taking the convex hulls of these smaller representations. All of the colored concepts are the original MetaThesaurus terms which describe part of the text of the relation. Every other entity in the representations are semantic types which generalizes over the terms. None of the representations feature the top entity of the ontology, so the most general types are Entity and Event.

Tree 2 corresponds to the following text: *but not when the minor G allele was.*

Tree 2

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Functional_Concept
            Negation
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               Alleles
```

Tree 3 corresponds to the following text: *Phosphorylated SMAD2 and SMAD3, in association with SMAD4, form a complex*

Tree3

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Qualitative_Concept
            Complex
            Qualitativeform
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               SMAD2gene
               SMAD3gene
               SMAD4gene
Event
   Phenomenon_or_Process
      Natural_Phenomenon_or_Process
         Biologic_Function
            Physiologic_Function
               Organism_Function
                  Mental_Process
                     Mentalassociation
```

Tree 4 corresponds to the following text: *which accumulates in the nucleus and acts as transcription factors to regulate target genes.*

Tree 4

```
Entity
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Cell_Component
               CellNucleus
            Gene_or_Genome
               CandidateDiseaseGene
      Substance
         Chemical
            Chemical_Viewed_Structurally
               Organic_Chemical
                  Amino_Acid,_Peptide,_or_Protein
                     TRANSCRIPTIONFACTOR
```

Tree 6 corresponds to the following text: *However, no significant association of both the polymorphisms was seen with treatment response to NACT.*

Tree 6

```
Entity
   Conceptual_Entity
      Idea_or_Concept
         Functional_Concept
            therapeuticaspects
   Physical_Object
      Anatomical_Structure
         Fully_Formed_Anatomical_Structure
            Gene_or_Genome
               SLC13A5gene
Event
   Phenomenon_or_Process
      Natural_Phenomenon_or_Process
         Biologic_Function
            Physiologic_Function
               Molecular_Function
                  Genetic_Function
                     GeneticPolymorphism
               Organism_Function
                  Mental_Process
                     Mentalassociation
```