Learning Information Extraction Rules with Sequential Patterns: a Quantitative and Qualitative Evaluation

Laura Handojo

Erasmus Mundus Master Program
in Language and Communication Technologies

June 18, 2012

Université de Lorraine
UFR - Mathématiques et Informatique
Supervisor: Yannick Toussaint
Academic year: 2011-2012

Rijksuniversiteit Groningen
Faculty of Arts
Supervisor: Dr. Gosse Bouma

Abstract

This thesis describes an automatic approach to learning information extration rules, or more precisely relation extraction rules. It uses sequential pattern mining to learn patterns from research abstracts on a rare disease called *fibromuscular dysplasia*. Information extraction can be useful in this field for creating review articles on rare diseases, thereby making information on them more accessible.

Preprocessing includes lemmatization, POS tagging and semantic tagging. Different settings with regard to the semantic tags as well as what to use as input sequences are tried out. The patterns are post-processed in order to obtain those which relate entities in a treatment-relation. After a manual analysis, the patterns are filtered and gap constraints are implemented to reduce noise. Evaluated on a test set, the best F_1-measure reaches 50%.

Contents

1 Introduction

Recently, information extraction (IE) has been increasingly applied to the domains of biology, biomedicine and medicine. In these domains, the growing number of articles makes processing with human eyes unfeasible and time-consuming, while the information in them can be crucial to further research or treatment of patients. Therefore, automatic extraction of certain information is an important application in this domain.

IE is the task of identifying and extracting information from raw text. One subtask of IE is named entity recognition (NER), which is the identification of references to specific things. Another subtask of IE is the identification of relations between the extracted named entities. A third subtask of IE involves co-reference resolution, i.e. the identification of sets of expressions referring to the same entity (Humphreys et al., 2000). In order to develop an IE system, we have to define which named entities and relations we want to extract, i.e. a template, and we have to develop IE extraction rules for the template (Kim et al., 2007).

There are basically three ways to develop IE rules: hand-construct them, use Machine Learning (ML), or data-mine them. While hand construction takes a lot of time and needs to be adapted to every new domain, ML based methods are automatic. However, they require training material and their output is supposed to be hard to understand. On the other hand, data mining techniques are also automatic and their output is easy to understand.

My work has been applied to research articles on rare diseases, specifically on one rare disease called *fibromuscular dysplasia*. Since a rare disease affects less than 0.05% of the population, interest in them is not widespread. However, advancing research on them is crucial since for many of them, the treatment or even the cause are not known yet. Therefore, an automatic method of learning IE rules will be very useful here. My aim has been to extract the different ways with which a given disease is treated. I have assumed that the required named entities are already recognized.

In order to learn rules for relation extraction, I have made use of sequential pattern mining, i.e. a data mining technique. I have implemented constraints in a postprocessing step and have included a very simple anaphora resolution method. Finally, I have analyzed the quality of the resulting patterns and have come up with ways to further improve the patterns.

In chapter 2 of this thesis, I will define information extraction and its subtasks in more detail and point out issues for them. Furthermore, I will describe different methods for IE, which have been presented in the literature. In chapter 3, I will describe the data that I have used as well as the algorithm for sequential pattern mining, preprocessing and postprocessing steps, and a simple method for anaphora resolution. In chapter 4, I will give a manual analysis of learnt sequential patterns and their performance

as relation extraction rules. Finally, in chapter 5, I will discuss some shortcomings and possible improvements of my work and give a conclusion.

2 Information Extraction - State of the Art

In this section, I will define what information extraction is and how it can be useful. Different subtasks of IE and issues that need to be dealt with are presented. Finally, I will outline various approaches to IE.

2.1 Definition

Information extraction refers to the automatic extraction of structured information, such as named entities or relationships between named entities, from unstructured text and its placement into a template. IE is useful when a large number of natural language text exists so that processing with human eyes is too time-consuming, while it is possible to predefine the information that is supposed to be extracted (Humphreys et al., 2000).

Therefore, IE is especially useful for extracting information from scientific research papers, e.g. in the domain of biomedicine. As an example, the MEDLINE database (searchable via PubMed[1]), which is a database for citations of biomedical literature, currently contains about 19.6 million citations and has been growing at an average rate of 550,000 new ones per year since 1995 (U.S. National Library of Medicine, 2012). Therefore, keeping up with new publications can be very time-consuming, while it is important in order to prevent overlaps in research and to promote further research, for example on diagnosis, prevention and treatment of diseases (Cohen and Hersh, 2005).

Information extraction consists of several subtasks: named entity recognition (NER), relation extraction and co-reference resolution.

Named Entity Recognition: NER is the identification of all names referring to a certain type of object in a text or text collection. For example, one could extract all the names referring to persons or locations, or in the biomedical domain all the names referring to genes or diseases (Humphreys et al., 2000). In example 1, the words in blue are named entities of the type *disease* and *syndrome*[2], respectively.

(1) *Fibro-muscular dysplasia (FMD) is a rare but well documented disease with multiple arterial aneurysms.*

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

[2] The term *syndrome* refers to the co-occurrence of clinical signs, symptoms and other characteristics of a disease. A clinical sign is something that is observed by a doctor, while a symptom is experienced by the patient.

The idea behind NER is that it enables identification of relationships between named entities, e.g. between a certain person and a certain location, by knowing that those types are related to each other and that those particular names are instances of those types (Cohen and Hersh, 2005). Approaches to NER can be manually encoded, statistically learnt, Machine Learning (ML) based or text mined. I will describe selected approaches in section 2.3.

Relation extraction: Relation extraction refers to the task of identifying relations between named entities. For instance, if it is known that named entities of type *person* are related to named entities of type *location* in a living-relation, it can be extracted that a certain person lives in a certain city (Humphreys et al., 2000). In the biomedical domain, if it is known that genes cause diseases, it can be extracted that a certain gene causes a certain disease. In example 2, the words in blue are named entities of the type *syndrome* and *therapeutic procedure*, respectively, which are related in a treatment-relation.

(2) *The aneurysm was treated by embolization.*

Approaches to relation extraction can be manually encoded, ML based or text mined. I will present various approaches in section 2.3.

Co-reference resolution: Another less discussed subtask of IE is co-reference resolution, also called anaphora resolution. This is the identification of different names, definite noun phrases (NPs) or pronouns, called *anaphora*, referring to the same entity as an earlier name or NP, called the *antecedent* (Humphreys et al., 2000). This can be illustrated by the abstract in figure 1.

1 A case of fibromuscular dysplasia (FMD) with intra- and extracranial multiple aneurysms is reported. A 42-year-
2 old woman was admitted to Kagawa Central Hospital with severe headache and stiffness of the neck. CT scan
3 showed subarachnoid hemorrhage predominantly in the left side of the basal cisterns and hydrocephalus.
4 Angiography at admission revealed marked stenosis and dilatation of the extracranial major arteries and
5 multiple aneurysms in the lt. PCA, lt. ICA, bil. VA, and the lt. renal artery. String-of-beads appearance was also
6 seen in the branches of the lt. external carotid artery. During the operation, the PCA aneurysm which has been
7 diagnosed as the ruptured one, was found to arise from the posterior communicating artery itself. It was thus a
8 so-called true posterior communicating aneurysm. The aneurysm was trapped by clipping the artery on both
9 the ICA and the PCA sides. The giant aneurysm of the lt. ICA was successfully treated by lt. STA-MCA
10 anastomosis and ligation of the lt. ICA. [...]

Figure 1: Example abstract

For example, *the PCA aneurym* (figure 1, line 6) refers to one of the *multiple aneurysms in the lt. PCA, lt. ICA, bil. VA, and the lt. renal artery* (line 5). Next, *it* in line 7 refers to *the PCA aneurysm*, and so does *The aneurysm* in line 8.

The motivation for doing co-reference resolution is to be able to find more instances of a relation. For instance, if the IE system extracts a relation between *aneurysm* (figure 1, line 8) and *clipping the artery* (line 8), this might not be informative enough since this treatment method might be something which can only be used on certain aneurysms. Furthermore, for pronouns, as long as they are not recognized as a named entity of the given type, they cannot be extracted in a relation with another named entity. Even if they are extracted, without their antecedent they are not informative, either. Therefore, this shows the importance of anaphora resolution. Approaches to anaphora resolution are usually manually encoded or ML based, some of which I will describe in section 2.3.

Development of an IE system: In order to develop an IE system, one has to develop a template, i.e. specify which named entities should be recognized and which relations between them should be extracted. Furthermore, one has to develop rules which will be used in the process of NER and relation extraction (Kim et al., 2007). Based on the method, the development of IE rules might include preprocessing steps, such as POS tagging, lemmatization, syntactic or semantic parsing or others (Nédellec, 2004), and postprocessing steps, such as the implementation of constraints based on specific features or manual selection of rules (Kim et al., 2007).

Evaluation of IE: Finally, the IE system should be evaluated. Evaluation is usually done for each of the subtasks separately. Usually, a gold standard annotation, i.e. a (human) annotation which is considered as perfect, is taken and the system's output is compared to it. Two metrics called *precision* and *recall* are calculated. As shown below, precision is the percentage of the system's output that is correct, meaning that incorrect output decreases the precision (Humphreys et al., 2000).

$$precision = \frac{|\text{correct extracted answers}|}{|\text{all extracted answers}|}$$

Recall is the percentage of the correct gold standard annotations which occur in the system's output, meaning that missing output decreases the recall (Humphreys et al., 2000).

$$recall = \frac{|\text{correct extracted answers}|}{|\text{all correct answers in the gold standard}|}$$

Often, the F_1-measure, which is the harmonic mean between precision and recall, is calculated as well.

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

While state-of-the-art NER systems can achieve between 75 and 85% F_1-measure, the performance for relation extraction tasks is mixed, depending on the relation to be extracted and the data that is used (Cohen and Hersh, 2005).

GATE: There are some existing IE tools, for example GATE (General Architecture for Text Engineering)[3]. GATE is a graphical environment for developing language engineering components and running them in combination with or without existing components. Existing components are known as CREOLE (a Collection of REusable Objects for Language Engineering). Among others, they include resources for preprocessing text, such as tokenization, POS tagging or syntactic parsing.

They also include an IE system called ANNIE (A Nearly-New IE system). It consists of a tokenizer, a sentence splitter, a POS tagger, a gazetteer used for NER, an orthomatcher and a coreferencer, both used for co-reference resolution. Custom rules, e.g. for relation extraction, can be implemented via a finite state transducer, which uses JAPE (Java Annotations Pattern Engine) grammars as input. A JAPE grammar consists of patterns over text strings or previously created annotations, and annotations to be created (Cunningham et al., 2002).

Furthermore, CREOLE provides resources for processing biomedical text, e.g. GENIA for tokenization, POS tagging, shallow parsing and NER, or MetaMap for semantic tagging using the UMLS Metathesaurus (Cunningham et al., 2011).

Finally, GATE allows the user to create manual annotations, thereby creating a gold standard annotation. Then, with GATE's AnnotationDiff tool, automatically created annotations can be evaluated against the gold standard, using precision, recall and F-measure (Cunningham et al., 2002).

To sum up, IE is the automatic extraction of information from unstructured text. Its subtasks include named entity recogntion, relation extraction and co-reference resolution. An IE system is developed by defining a template and extraction rules, and it is evaluated using precision and recall. There are existing tools for it, such as GATE.

2.2 Different Issues of Information Extraction

IE systems need to handle several issues, among them variation and ambiguity of named entities or relations, performance and evaluation issues.

Issues for NER: Firstly, NER has to deal with the fact that the same named entity may be expressed through orthographic, morphological, syn-

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

tactic or semantic variations of its name. For example, *arterial fibromuscular dysplasia* may be expressed as:

(3) *arterial fibro-muscular dysplasia* (orthographic variation)

(4) *arterial FMD* (abbreviation)

(5) *fibromuscular dysplasia of the arteries* (syntactic variation)

(6) *intimal fibroplasia* (semantic variation - hyponym)

Furthermore, names can be ambiguous, i.e. one name can refer to more than one named entity. For example, *embolization* can refer to a therapeutic procedure as in example (2), or it can refer to a syndrome as in example 7.

(7) *This case report describes a patient with digital embolization from brachial artery fibromuscular dysplasia.*

Variations in names can decrease the recall of NER, while ambiguity of names can decrease its precision (Nenadić et al., 2002). In existing methods, variations are for example handled by using similarity metrics (Mooney and Bunescu, 2005), a normalization step (Schneider et al., 2009) or by using transformation rules (Cherfi et al., 2003).

Issues for relation extraction: Relation extraction needs to deal with the fact that the same relation may be expressed through different verbs in active or passive, through different nominalizations or even through relative clauses (Kim et al., 2007). For instance, a treatment-relation between a syndrome A and a therapeutic procedure B may be expressed as:

- `treat A with B` vs. `A be treated with B` (active vs. passive)
- `undergo B for A` (different verb)
- `B as treatment of A` vs. `management of A with B` (different nominalizations)
- `A which be treated with B` (relative clause)

If one of those patterns is included in the IE rules, but another one or more are lacking, this will decrease recall, as described in Cellier et al. (2010). Here again, transformation rules may be useful (Kim et al., 2007).

Furthermore, for relation extraction, coordination and negation have to be treated (Bunescu and Mooney, 2005; Cherfi et al., 2003). Examples of this are:

(8) *Patients with fibromuscular dysplasia (FMD) and hypertension are frequently treated with percutaneous transluminal renal angioplasty (PTRA).*

(9) *Twelve (52.2%) of the patients were taking no antihypertensive medications at 6 months and were classified as cured.*

Missing one or even more named entities belonging to a coordinated structure in a relation would decrease the recall, while extracting a relation from a negated sentence would decrease the precision.

Performance and evaluation issues: Another problem is that IE systems usually perform with high precision, but with significantly lower recall (Nahm and Mooney, 2004). Therefore, methods for improving recall are needed. One approach is to do NER and relation extraction within the same step. This is based on the idea that named entities can help to identify relations, through their presence as well as type, and identified relations can help to recognize previously missed named entities (Kate and Mooney, 2010). Another approach is to use back-off techniques, such as simple surface patterns, to improve recall whenever the primary method fails (Schneider et al., 2009). This, however, introduces the risk of hurting the precision.

However, evaluation using recall is not always possible. This is mostly due to the fact that gold standard annotations do not always exist. As a solution, Kim et al. (2007) for example, approximate recall by assuming that each relevant sentence contains only one relation. Based on this, they use the number of relevant sentences as the denominator in the calculation of recall. If an approximation is not possible, or also in addition to the use of precision and recall, a manual evaluation by a domain expert can be done (e.g. Cherfi et al. (2003)).

Since the number of IE rules produced by a learning algorithm is usually huge, the process has to be constrained in some way (e.g. Cellier et al. (2010)) or the rules have to be ranked in an intelligent way (e.g. Cherfi et al. (2003)), so that the number of rules which the domain expert has to evaluate is manageable.

To sum up, IE has to deal with variation and ambiguity of named entities and relations, and with phenomena such as coordination and negation. Furthermore, evaluation is not always possible, has to be approximated or be done manually.

2.3 Methods for Developing Information Extraction Systems

There are basically three approaches to developing IE rules: manual, ML based and data mining ones. I will describe selected methods for each of the approaches.

2.3.1 Hand-constructed Rules

Previously, many IE systems have made use of hand-constructed rules. While they usually give a good precision, their recall is much lower. Furthermore, the construction of manual rules is very time-consuming and needs to be adapted to every new domain (Nédellec, 2004).

NER & relation extraction: Ono et al. (2001) are an example of doing NER and relation extraction with hand-constructed rules. They want to extract protein-protein interactions. They recognize protein names based on pattern matching, using a manually constructed lexicon. Similarly, they use pattern matching rules for relation extraction. The rules make use of protein names, prepositions and a set of keywords which indicate interaction. An example pattern is the following, where A and B express protein names:

```
A interact with B
```

Ono et al. (2001) handle coordinated sentences by finding them with pattern matching rules and then splitting them up into sub-sentences. Different from other approaches, they also try to handle negated sentences by including negated patterns into their rules. An example is:

```
A not interact with B
```

Relation extraction & anaphora resolution: Pustejovsky et al. (2002) describe a method for relation extraction and anaphora resolution. They want to extract inhibit-relations between biological entities. With the help of semantic automata and corpus analysis, they create extraction rules based on predicate-argument structures for the verb in question. Their anaphora resolution method is based on syntactic information, such as person and number, semantic type and simple string comparisons. I have adopted a similar anaphora resolution method in my work.

Co-reference resolution: Castaño et al. (2002) describe a method for co-reference resolution in biomedical texts. They limit resolution to pronouns in the third person and definite NPs tagged with a semantic type from the biomedical domain. For each candidate anaphora-antecedent pair, a salience measure is calculated based on agreement of person and number. Furthermore for definite NPs, string similarity and matching semantic types are taken into account. Unlike other approaches, a semantic type is taken into account for pronouns, as well: it is "coerced" on them, based on semantic types occurring frequently with the verb in the given sentence.

Tested on 46 MEDLINE abstracts, Castaño et al. (2002) report a precision of 77% and a recall of 72%. This gives an F_1-measure of 74%.

2.3.2 Statistical and ML Based Learning Methods

Statistical methods are less time-consuming than manually coded rules and usually give a higher recall, but their precision is usually worse (Nédellec, 2004).

NER: Nenadić et al. (2002) are an example of using a statistical method for NER. Their overall system uses automatic term recognition and clustering in order to populate an ontology and perform IE-like tasks based on the ontology. Terms can be thought of as named entities, even though unlike named entities they are not further distinguished into classes. Terms are recognized based on two statistical measures: C- and NC-values. The C-value for a candidate term is calculated based on different frequency counts over a given text collection, such as frequency of the candidate or frequency as a substring in other candidates. The NC-value is an improvement of the C-value since it also takes surrounding words of the candidate term into account. Nenadić et al. (2002) deal with variation in terms by normalizing them to a single form before performing statistics on them.

Machine Learning based techniques automatically learn rules and are therefore less time-consuming, as well. Unlike hand-coded methods, they can easily be adapted to new domains (Kim et al., 2007). As with statistical methods, their recall is usually higher than that of manual approaches, but unlike statistical methods, their precision is also comparable to that of manual approaches (Nédellec, 2004).

However, supervised ML techniques need a correctly annotated text collection to learn from (Kim et al., 2007), i.e. all the named entities and relations between them have to be annotated as such, which can be quite time-consuming. They also need a set of features on which to base the learning (Cellier et al., 2010). Furthermore, according to Plantevit et al. (2009), their output is not easily understandable by humans and therefore not as useful whenever the validation of rules and possibly adaptation by a domain expert is desired.

NER & relation extraction: Kate and Mooney (2010) propose a method for doing NER and relation extraction at the same time using so called "card-pyramid parsing". As already mentioned in section 2.2, performing the two tasks in the same step is based on the idea that while recognized named entities help in the identification of relations, identified relations may also support recognition of otherwise missed named entities. Therefore, this is a method to improve recall and maybe also precision. Each sentence is represented as a card pyramid, with the named entities at the very bottom and relations (including a type *NR* for *no relation*) linking them, up until the top node of the pyramid. An example of this can be seen in figure 2. The

card-pyramid parsing amounts to labelling the nodes with specific named entity and relation types. The grammar used for parsing is learnt using Support Vector Machines. Features for named entity learning include word subsequences of the candidate entity and its POS tags, surrounding words and others. Features for relation learning include existing named entities.

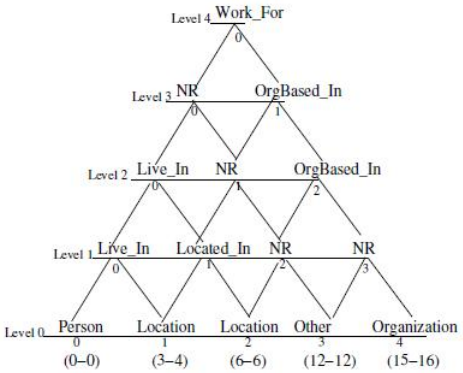

Figure 2: Example of a card pyramid by Kate and Mooney (2010)

Relation extraction: Craven and Kumlien (1999) compare three ML based approaches for extracting relations between proteins and other entities. Two use statistical ML, or more specifically Naive Bayes classifiers, and the other uses relational learning. In the first approach, they hand-label sentences as positive or negative instances of a given relation and then use a Naive Bayes classifier with a bag of words (BOW) representation[4].

In the second approach, they exploit article references for instances of the desired relation from a database, since the preparation of the training data is quite time-consuming. They make the simplifying assumption that each sentence in an abstract containing the relation is a positive example of it. Sentences in abstracts not containing the relation are taken as negative examples. Again they use Naive Bayes classification with a BOW representation. The second approach gives a better precision with a similar recall than the first approach.

Unlike the other approaches, the third makes use of linguistic information, namely phrase types (e.g. NP), order of phrases, whether a phrase contains another phrase, grammatical functions of phrases (e.g. subject, object) and co-occurrence of phrases in the same clause. The relational learning algorithm then learns rules which cover positive but not negative instances of the given relation. Compared to the other approaches, the pre-

[4]this means that each sentence is represented as a BOW, i.e. assuming that the order of the words does not matter and that the occurrence of a word does not depend on other words in the sentence (Craven and Kumlien, 1999)

cision is much higher at a similar level of recall. Therefore, Craven and Kumlien (1999) show the benefit of using linguistic information over not using it.

Similarly to Craven and Kumlien (1999), Kim et al. (2007) use a collection of sentences which are labelled as relevant or irrelevant with respect to a given relation. They want to extract relations between proteins and diseases, functions or structure. They use Inductive Logic Programming (ILP), with features such as POS tags, named entities and grammatical functions. ILP then tries to find rules which cover all relevant instances, but no irrelevant instances for a given relation from the labelled sentences. Before the thus developed IE rules are applied to a test set of texts, a domain expert selects relevant rules, eliminating about half of them. However, they do not give details on what basis the expert makes his decisions. I have also done a manual selection of rules, but will describe this process in more detail.

Schneider et al. (2009) combine manual work and ML in their approach to extracting interactions between proteins. Firstly, they use dependency parsing on all sentences and then extract syntactic paths which link two proteins. Since not all syntactic paths actually express an interaction, they manually classify paths into relevant and irrelevant paths. Then, for relevant paths, they use ML to learn words which frequently occur inside them, so called "transparent words" (Schneider et al., 2009). A transparent word is usually a noun which does not add to the meaning, as in the following example by Schneider et al. (2009), where the transparent word is marked in blue:

(10) *A activates groups of B* vs. *A activates B*

Finally, they manually develop surface patterns, such as:

```
A interacts with B
```

The resulting IE system is used as following: whenever a syntactic path can be applied to a test sentence, this is used for extraction. If no path exists and the sentence contains any transparent words, those are cut out of the sentence. Then the system checks again if a syntactic path exists. If not, the surface patterns can be used as a back-off technique. Both the transparent words and the surface patterns are used in order to improve the recall of the system.

Anaphora resolution: Yang et al. (2004) describe an ML based approach to co-reference resolution in biomedical literature. They want to link an anaphora to a "coreferential cluster" instead of a single antecedent. This is based on the idea that the closest antecedent to an anaphora sometimes

lacks information that would make linking them to each other possible, while an earlier antecedent has this information. Features for learning include type (e.g. definite/indefinite NP or pronoun) of the anaphora and of a reference NP from the cluster, agreement of number, gender and semantic type between the anaphora and both a reference NP and the cluster and string similarity. The learning is done with the C5.0 algorithm, i.e. decision trees. By looking at the decision trees, Yang et al. (2004) establish that the string similarity features are most important.

Based on this, I assume that a very simple anaphora resolution method, which is based mostly on string similarity, might already be quite successful.

2.3.3 Data and Text Mining Methods

To overcome the disadvantages of manual and ML based techniques, research on the use of data and text mining techniques for IE has been done. Similarly to ML based methods, they are easily adaptable to new domains. On the other hand, unlike ML based techniques, they do not need annotated texts for learning (Plantevit et al., 2009). Therefore, the data preparation for them is less time-consuming than for ML based methods. Furthermore, Plantevit et al. (2009) claim that the output of data/text mining is understandable for human experts of the domain, while the output of ML is not.

Data and text mining aim at identifying patterns from structured data (such as a database, in the case of data mining) and from unstructured data (i.e. text, in the case of text mining) (Delgado et al., 2002). Therefore, in the form of patterns, they try to generate new information, while IE wants to extract information that is already encoded in the text and therefore not new (Hearst, 1999). Mining techniques are frequent itemset mining, association rule mining, sequential pattern mining and sequential rule mining. I will shortly define important concepts for them in the context of text mining.

Important concepts: An *itemset* is a set of keywords which characterize a document[5]. In this way, itemsets can be used to detect the topic of a document. The number of documents which contain the itemset, normalized by the size of the entire document collection, is called the *support* of the itemset (Agrawal et al., 1993). Given the dataset in table 1 where d stands for document and i for item, itemsets are (i1), (i2), (i3), (i1, i2), (i1, i3), (i2, i3), (i1, i2, i3).

An itemset is called *large* or *frequent* if it appears in more than a fixed threshold, called the *minimum support*, of the documents (Agrawal et al.,

[5]for the sake of simplicity, I will keep referring to documents in the following paragraphs, but an itemset might also characterize a smaller unit, such as a paragraph or a sentence

	i1	i2	i3
d1	x		x
d2		x	x
d3	x	x	x

Table 1: Example dataset for itemsets

1993). A frequent itemset is called *closed* if there is no bigger itemset containing it while having the same support (Cherfi et al., 2003). Closure is based on the idea that a closed itemset will give the maximum information possible, as opposed to incomplete information coming from a non-closed itemset. Closed frequent itemsets with a support of $\frac{2}{3}$ in table 1 are (i3), (i1, i3), (i2, i3).

An *association rule* of the form $A \Rightarrow C$ consists of an itemset on the left-hand side of the rule, called antecedent, and an itemset on the right-hand side, called the consequent[6]. It expresses that whenever the antecedent occurs in a document, the consequent will appear with a given probability. The support of an association rule is defined as the number of documents which contain both the antecedent and consequent, normalized by the size of the document collection (Agrawal et al., 1993). The probability that the consequent will occur, given the antecedent, is called *confidence* of the rule. It is defined as the support of the association rule divided by the support of the antecedent alone. An association rule is called *valid* if its support and confidence are greater than their fixed thresholds (Cherfi et al., 2003). Association rules with a support of $\frac{2}{3}$ and a confidence of 1 in table 1 are (i1) $\Rightarrow$ (i3) and (i2) $\Rightarrow$ (i3).

A *sequential pattern* is similar to an itemset, but unlike in an itemset, the items are ordered. Therefore, not only mere occurrence in the document, but also their order in it matters. However, the items do not have to occur consecutively in order to form a sequential pattern. As with itemsets, the support of a sequential pattern is defined as the number of documents in which it occurs, normalized by the size of the document collection. A *closed frequent sequential pattern* is defined analogously to a closed frequent itemset (Agrawal and Srikant, 1995). Given the dataset in table 2 where s stands for input sequence and i for item, sequential patterns with a support of $\frac{2}{3}$ are <(i1) (i3)> and <(i3) (i4)>.

Furthermore, a sequential pattern can also consist of ordered itemsets instead of ordered items (Agrawal and Srikant, 1995). Many algorithms implement constraints into the mining process, e.g. syntactic ones or limits to the gap which exists between items in the document (Plantevit et al.,

[6]both of the itemsets might also be of size one, i.e. be a single item

s1	<(i1) (i2) (i3)>
s2	<(i1) (i3) (i4)>
s3	<(i3) (i2) (i4)>

Table 2: Example dataset for sequential patterns

2009).

Similarly to association rules consisting of itemsets, a *sequential rule* consists of sequential patterns. The rule expresses that if the sequential pattern in the antecedent occurs in the document, the sequential pattern in the consequent will occur with a given probability. Support and confidence of a sequential rule are defined analogously to those of association rules (Plantevit et al., 2009). Sequential rules with a support of $\frac{2}{3}$ and a confidence of $\frac{2}{3}$ in table 2 are (i1) $\Rightarrow$ (i3) and (i3) $\Rightarrow$ (i4).

There are two research directions combining text mining and IE: methods that use text mining to automatically learn IE rules (e.g. Cellier et al. (2010); Plantevit et al. (2009)), and methods that integrate IE and text mining with each other (e.g. Nahm and Mooney (2004)).

Within the first direction, sequential pattern mining, sequential rule mining, frequent itemset mining and association rule mining or combinations of them are used.

NER: Plantevit et al. (2009) are an example of the use of text mining for NER. They compare sequential pattern and sequential rule mining with a third approach, a combination of sequential pattern and itemset mining, which they call left-sequence-right (LSR) pattern mining. They want to recognize gene names. Firstly, they detect that sequential patterns for NER give a good recall but low precision and that sequential rules give a good precision but low recall. Therefore, they propose LSR patterns, which are sequential patterns with itemsets on their left and right sides. They implement two constraints: all patterns must contain a named entity and there must be no gaps between the items of the sequential pattern. The resulting patterns are used for NER as following: whenever the confidence of a sequential pattern[7] is not high enough, its context is checked for the presence of a minimum of words from the itemsets. In this way, precision can be improved, while also having a good recall through the use of sequential patterns.

[7]they define the confidence of a pattern to be its support divided by the support of a pattern with all words except for the given named entity

Relation extraction: Cellier et al. (2010) use recursive sequential pattern mining together with some linguistic constraints in order to build IE rules for relation extraction. They assume that the data is already annotated with recognized named entities. The linguistic constraints used are that each pattern must include two named entities and that it must include either a verb or a noun. I have adopted similar constraints in my method.

In this way, Cellier et al. (2010) avoid dealing with anaphora and with relations between more than two entities. They use recursiveness, i.e. they repeat the pattern mining on the output of the process, in order to reduce the number of patterns. This is useful since the mining process usually produces a huge, unmanageable number of patterns. The reduced set, on the other hand, can be validated by a human expert. In their experiment, roughly 35% of the candidate patterns are validated by the expert. Then the validated rules are tested on 200 random sentences each from three text collections (BioCreative, GeneTag, AIMed). The highest precision reaches 93% and the highest recall 84%, both on the AIMed sentences.

Usefulness of itemsets and association rules for IE: Cherfi et al. (2003) use frequent itemsets and association rules for creating rules in the domain of molecular biology. They propose a number of quality measures which can be used to sort the rules according to their interestingness and then have a domain expert judge the rules based on whether the words in the antecedent can be related to those in the consequent.

However, the rules are not used for IE. In fact, the form of appearance of the rules indicates that association rules are not useful for IE with the given kind of data. This is mostly due to the fact that the items are not ordered in any way in the rule, the disadvantage of which can be illustrated by example 11.

(11) *The following patient with multiple visceral aneurysms first had coil embolization of bilateral renal artery aneurysms and then operative excision of her remaining splenic artery aneurysms to minimize the potential morbidity of a larger operation.*

If a rule exists that associates *coil embolization* with *aneurysms* (yielding a treatment-relation, all marked in blue) but the order of occurrence is not known, it is possible that *multiple visceral aneurysms* or *splenic artery aneurysms* are extracted instead of the correct *bilateral renal artery aneurysms*. This shows that itemset and association rule mining might be useful for information discovery, but not for information extraction from scientific abstracts.

Anaphora resolution: To my knowledge, there are no data or text mining approaches to co-reference resolution.

As I said, other researchers do not use text mining in order to develop IE rules, but rather implement IE in a different way and then use text mining to improve the system.

Relation extraction: An example of this are Nahm and Mooney (2004). They want to extract job requirements from a collection of computer-science job postings. Firstly, they learn and apply IE rules by Boosted Wrapper Induction[8]. From the output, they mine association rules, such as "if requirement A has value B, then requirement C will have value D". In order to handle variations of named entities, they do the association rule mining based on soft matching of words using edit distance or BOW measures. However, the actual application of the resulting association rules is subject to the presence of the consequent from the rule or a similar string in the text. In this way, they can improve the recall of their method while preventing false positives.

Mooney and Bunescu (2005), using the same technique as Nahm and Mooney (2004), compare two approaches to mining association rules: on the one hand, mining them from a database constructed by IE, and on the other hand, mining them from a manually built database. Since the accuracy of rules is similar in the two approaches, they conclude that IE is a useful way to build a database which can then be mined for new knowledge.

Summary: To sum up, in this section I have defined information extraction with its subtasks named entity recognition, relation extraction and coreference resolution. I have described issues for them, such as variations in names or relations, difficulties for evaluation or the need to improve performance. Furthermore, I have described different manual, ML based and data mining approaches to the subtasks. Manual approaches are mostly based on pattern matching or matching of linguistic features, such as person and number. Machine learning methods use features such as words, POS tags and grammatical function. Finally, data and text mining techniques use mainly sequence mining with different constraints.

3 Sequential Pattern Learning for Information Extraction

In the following section, I will describe collection and characteristics of the data used for learning and testing. I will also give details on the IE relation that I aimed to extract. Furthermore, I will describe what tools, algorithms and scripts have been used. Based on the related literature, I have decided

[8] a wrapper is a contextual pattern, and wrapper induction is the automatic learning of wrappers (Nahm and Mooney, 2004)

to use sequential pattern mining in order to learn IE rules. I will also make use of preprocessing and postprocessing steps.

3.1 Description of Data and Information Extraction Task

Domain: As I said in section 2.1, information extraction has been increasingly applied to scientific research articles in the domain of biomedicine. For research articles on rare diseases, it is even more important that there exists an automatic method of learning IE rules, and that as a result there exists an automatic way of extracting information. Since a rare disease affects less than 0.05% of the population (European Commission, 2012), a doctor with a patient with a rare disease might have never come across it before. Therefore, finding the correct diagnosis and treatment might be difficult and could be supported by a quick search for relevant information with the possibility to go back to the original research articles. Furthermore, it is important to promote research on rare diseases since for many of them, the treatment or cause are not known yet.

Therefore, my work has been applied to research articles on rare diseases. I have used abstracts of research articles. Firstly, abstracts are usually freely available while the full text often is not. Secondly, all of the important information from the article is presented in a compact way in the abstract, which will make IE easier or at least will make the processing less time-consuming.

I have used abstracts on a particular rare disease called *fibromuscular dysplasia*. The choice of disease is based on the availability of a domain expert. However, my approach could be applied to any other disease as well. Fibromuscular dysplasia (FMD) refers to a group of arterial diseases, which mostly affect the renal and carotid arteries (i.e. the arteries of the kidney and of the head and neck, respectively). Depending on the affected arteries, clinical signs and symptoms include stenosis, "string of beads" appearance, hypertension and aneurysms. Treatment methods include anti-hypertensive medications, percutaneous angioplasty and reconstructive surgery. The cause of FMD is not known yet (Plouin et al., 2007).

IE task: My aim was to extract the different ways with which the disease is treated. Therefore, diseases, clinical signs and symptoms (i.e. anything that can be treated) have been identified as named entities of one type. On the other hand, treatment methods, including procedures, devices and drugs (i.e. anything that can be used for treating) have been identified as named entities of another type. Then I have aimed to extract a treatment-relation between them.

Even though Plouin et al. (2007) already include main treatment methods in their review article, other research articles mention further possibilities, such as thrombolytic therapy, orthotopic autotransplantation or

nephrectomy. Therefore, extraction of treatment-relations will bring forth previously missed treatment methods of a clinical sign or of course newly adopted ones. Then, if the IE method is able to perform well, it can be used to extract information needed in order to create review articles on other rare diseases (i.e. for diseases where no review article exists yet).

Collection of data: The abstracts have been collected from PubMed[9], a database of citations for biomedical literature. The query used to search for abstracts was "`fibromuscular dysplasia[majr] AND arteries[majr]`"[10]. Abstracts were manually selected according to whether there is at least one instance of a treatment-relation. If there is more than one instance of a treatment-relation present and it is not possible to disambiguate which entities are linked to each other, the abstract has been excluded. Selected abstracts were saved as xml-files. In this way, PubMed meta information can be kept, while at the same time the abstract's text can be used on its own for processing.

In total, I have collected 95 abstracts. They contain 686 sentences and 14,310 tokens (including punctuation). This makes an average of 7 sentences per abstract and an average of 21 tokens per sentence.

Next, I have divided the data into three sets: one for training, one for development, and one for testing, with around 75%, 15% and 10% of the data respectively. The training set contains 70 abstracts with an average of 7 sentences per abstract and an average of 21 tokens per sentence. The development set contains 15 abstracts with averaged 8 sentences per abstract and averaged 21 tokens per sentence. The testing set contains 10 abstracts with an average of 7 sentences per abstract and an average of 22 tokens per sentence. This shows that the distribution of long and short abstracts is similar across the three sets, as well as the distribution of long and short sentences.

Characteristics of the data: With the help of the example abstracts in figures 3, 4 and 5, characteristics of the data will be illustrated.

Most abstracts have the following implicit structure: describing one or more patients' disease and possibly symptoms (figure 3, line 1), describing clinical signs with or without the diagnostic procedures to find them (figure 3, line 1), describing treatment methods for the clinical signs (figure 3, line 3). Additionally, some of them start with a more general description of the disease and its syndromes and end with follow-up procedures for the given patient (figure 4, lines 4-5) and/or a more general statement of how the given clinical signs should be treated (figure 3, lines 5-6).

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

[10] the query was automatically created by using the link to PubMed from the Orphanet homepage (http://www.orpha.net/consor/cgi-bin/index.php?lng=EN) on fibromuscular dysplasia

An 18-year-old woman with renovascular hypertension had stenosis due to fibromuscular dysplasia (FMD) of the bilateral renal arteries, and this did not respond to conventional balloon angioplasty owing to marked elastic recoil. Implantation of Palmaz-Schatz stents resulted in successful dilation of the stenosis and remission of hypertension. Transstenotic pressure gradients were measured by use of a pressure-monitoring guidewire. It is suggested that stenting is a very useful adjunct to balloon angioplasty in the treatment of renal artery stenosis due to FMD.

Figure 3: Example abstract

Patients with fibromuscular dysplasia (FMD) and hypertension are frequently treated with percutaneous transluminal renal angioplasty (PTRA). Because the goal of this procedure is the cure of hypertension, we reviewed the outcomes of 23 consecutive patients undergoing this procedure to determine factors associated with cure. Twelve (52.2%) of the patients were taking no antihypertensive medications at 6 months and were classified as cured. Using logistic regression, we found three variables to be independently associated with cure: level of systolic blood pressure before intervention (P = 0.02), duration of hypertension (P = 0.03), and age (P = 0.03). Younger patients with milder hypertension of a shorter duration were most likely to be cured. Analysis of the regression equation predicts that some patients with an extremely low chance of cure might be managed with a trial of medical therapy, because FMD is unlikely to progress to renal failure.

Figure 4: Example abstract

Fibromuscular dysplasia of the aorta is an exceedingly rare disease with 26 cases reported to date in Medline. We present a case of stenosis of the infrarenal abdominal aorta in a 49-year-old woman with a history of intermittent claudication. The patient underwent aortic endarterectomy, and subsequent anatomopathologic examination of the specimen revealed fibromuscular dysplasia. The possible causes of aortic stenosis in this case, its angiographic findings, and the alternatives of treatment are discussed.

Figure 5: Example abstract

Characteristics of the treatment-relation: The treatment-relation is usually expressed within one sentence (figure 3, lines 1-2 and 3-4), but this is not always the case. It may occur over two sentences or even with sentences in between (figure 5, lines 2-3).

The named entities (referring to diseases/syndromes and treatment methods) do not always occur in the same order: e.g *stenosis* is followed by *balloon angioplasty* in figure 3 (lines 1-2), but *stenosis* and *hypertension* are preceded by *implantation of Palmaz-Schatz stents* in figure 3 (lines 3-4). The order depends, for example, on the way the relation is expressed (e.g. verb in active vs. verb in passive), or on pragmatic factors such as special emphasis on one of the named entities.

Verbs where the semantic agent is the doctor, such as *treat*, usually occur in passive voice (figure 4, line 1), while verbs where the semantic agent is the patient, such as *undergo*, usually occur in active voice (figure 4, line 3). Apart from verbs, the treatment-relation may be expressed as a nominalization, e.g. as *treatment* (figure 3, line 5).

Rarely, there is a negation occurring with the treatment-relation. In some cases, this means that the treatment method was not used (figure 4, line 4). In other cases, it means that it was used but was not successful (figure 3, line 2). Conjunctions frequently appear, expressing that one disease/syndrome is treated with more than one treatment method, one treatment method is used for more than one disease/syndrome (figure 4, line 1) or that pairs of diseases/syndromes and treatments are used.

While diseases/syndromes are straightforwardly referred to by nouns, treatment methods may be expressed as nouns as well as verbs (e.g. *resection* vs. *be resected*).

Both the disease/syndrome and the treatment method may be referred to by an anaphor. On average, there is about one anaphor referring to a disease/syndrome or treatment method within an abstract, and only about half of them actually participate in a treatment-relation. This corresponds to the observation of Liang and Lin (2005) that anaphora do not occur frequently in biomedical texts. According to Castaño et al. (2002), most anaphora in biomedical literature are expressed through NPs or pronouns. In my data set, however, most of them are expressed by NPs. An example of this is *this procedure* in figure 4 (line 2 and 3). Of all tokens in my data set, less than 1% are pronouns. Those mostly refer to persons (e.g. figure 5, line 2) and only very rarely to diseases/syndromes or treatment methods.

To sum up, the data used for training, development and testing consists of 95 abstracts on fibromuscular dysplasia, each of which contains at least one treatment-relation. Characteristics of the abstracts are that most of them share the same content structure. Characteristics of the treatment-relation are that it mostly occurs within one sentence and that its entities may be linked by different verbs and nouns in different orders. Anaphora rarely occur within the treatment-relation.

3.2 Sequential Pattern Mining Algorithm BIDE+

I have used sequential pattern mining in order to learn IE rules for relation extraction, following Cellier et al. (2010). On the one hand, this is based on the fact that sequential patterns take order into account. This is important since language does so, as well. On the other hand, the choice of sequential patterns is based on the observation of Plantevit et al. (2009) that IE with sequential patterns yields a higher recall than with sequential rules. This is relevant since, according to Nahm and Mooney (2004), recall is usually much lower than precision for IE.

I have decided to use the algorithm BIDE+ for sequential pattern mining (Wang and Han, 2004). According to them, it has several advantages. Firstly, it mines closed frequent sequential patterns, where closure leads to a complete but more compact result. Secondly, BIDE+ is faster and uses less memory than other sequential pattern mining algorithms, both closed and non-closed ones.

Wang and Han (2004) describe the BIDE algorithm for mining sequential patterns consisting of items. The BIDE+ algorithm can be used for sequential patterns consisting of both items and itemsets. It works essentially in the same way as BIDE.

In a nutshell, BIDE first creates frequent sequential patterns of size one from the data. Treating each of them as a prefix, it uses pseudo projection to create a database of projected sequences for each of the prefixes. This means, given an input sequence and a prefix, a projected sequence is the remaining part of the input sequence after having removed the first occurrence of the prefix. For example, in figure 6 given prefix `C`, the projected sequences are `AABC`, `B`, `ABC` and `A`. Pseudo projection means that the database is not actually created, but rather the projected sequences are referred to with pointers.

Sequence identifier	Sequence
1	C A A B C
2	A B C B
3	C A B C
4	A B B C A

Figure 6: Example sequence database by Wang and Han (2004)

Then for each prefix and its pseudo projected database, locally frequent items are computed. For the prefix `C` and the support threshold $\frac{2}{4}$, locally frequent items in figure 6 are `A`, `B` and `C`. Furthermore, it is checked in each direction (i.e. before, within and after the prefix) if there is any item which always occurs with the given prefix. If not, the prefix is output as a closed frequent sequential pattern. In figure 6, none of the items always occurs with the prefix `C`. Therefore, `C` is a closed frequent sequential pattern.

By combining it with its locally frequent items, new prefixes are formed. For each new prefix, it is checked whether it can already be removed as non-closed before the previous steps are performed on it as well.

The BIDE+ algorithm is implemented in the Sequential Pattern Mining Framework (SPMF)[11], which is a data mining framework. SPMF contains not only implementations of algorithms for sequential pattern mining, but also for frequent itemset mining, association rule mining and sequential rule mining. I have used the SPMF in my work.

3.3 Learning Sequential Patterns for Treatment-Relation Extraction

This subsection describes my approach to the extraction of the treatment-relation, including preprocessing, sequential pattern mining and postprocessing.

3.3.1 Preprocessing

Firstly, I have used GATE to do tokenization, sentence splitting, lemmatization, POS tagging and shallow parsing. For the first two tasks, I have used GATE Unicode Tokenizer and ANNIE Sentence Splitter. The latter three tasks were performed by the GENIA tagger. Since it is specifically developed for parsing biomedical texts, its performance on them is much better than that of the ANNIE POS tagger. For example, the ANNIE POS tagger frequently mistags biomedical adjectives as nouns, e.g. *fibromuscular*. Since I have used POS tags for further preprocessing and for postprocessing, a good performance is important. The shallow parsing module of the GENIA tagger recognizes chunks of phrases. I have made use of this for anaphora resolution.

NER was implemented with GATE as well, or more precisely the MetaMap Annotator. MetaMap uses the UMLS Metathesaurus to map tokens to biomedical concepts. Referring to their Semantic Groups[12], I have taken the following MetaMap types for diseases and syndromes: Acquired Abnormality, Anatomical Abnormality, Cell or Molecular Dysfunction, Congenital Abnormality, Disease or Syndrome, Experimental Model of Disease, Finding, Injury or Poisoning, Mental or Behavioral Dysfunction, Neoplastic Process, Pathologic Function, Sign or Symptom. From now on, I will refer to this group of types as "DISY" for "DIsease or SYndrome". For treatment methods, I have taken the followings MetaMap types: Antibiotic, Biomedical or Dental Material, Clinical Drug, Drug Delivery Device, Medical Device,

[11] http://www.philippe-fournier-viger.com/spmf/

[12] full list: http://metamap.nlm.nih.gov/SemGroups_2011.txt; shortened list in appendix A

Pharmacologic Substance, Therapeutic or Preventive Procedure. I will refer to this group of types as "TRME" for "TReatment MEthod" from now on.

By way of my own JAPE grammars and Perl scripts, I have transformed nouns into semantic types. E.g. example 2 will look this way:

(12) *The dsyn was treated by topp*[13]

The use of semantic types is based on the idea that there will be less variation and thus more opportunity to learn patterns. Independently from each other, I have tried out four different type transformation methods:

1. **AllTypes:** every noun which has a MetaMap type is transformed into it
2. **RelevantTypes:** only nouns which have a type from the DISY or TRME group are transformed into it
3. **OnlyTwo:** nouns which have a type from the DISY or TRME group are transformed into "DISY" or "TRME", respectively
4. **Two&Others:** nouns which have a type from the DISY or TRME group are transformed into "DISY" or "TRME", respectively, while nouns with other MetaMap types are transformed into that type

I have expected that with methods OnlyTwo and Two&Others, more patterns might be learnt since there is even less variation. On the other hand, patterns learnt with methods AllTypes or RelevantTypes might be more precise. The four methods have been tested in the experiment phase.

All tokens which have not been transformed into a semantic type, were then transformed into their lemmata. Again this is supposed to eliminate variation. Finally, I filtered the tokens based on their POS tags and on a stop word list. In this way, I kept nouns, verbs (except for modals, *have* and *be*), adjectives, adverbs and infrequent prepositions. E.g. example 12 will then look this way:

(13) *dsyn treat topp*

Stop wording is based on the idea that even though certain words, such as determiners, frequently appear, their presence or absence in a pattern used for relation extraction is not helpful. However, this has not been tested. Therefore, it is not clear if they are merely redundant or if they could make the performance worse.

Finally, there are four methods for transforming the text into input sequences for sequential pattern mining, which have been tried out independently from each other:

[13] mappings from abbreviations to full types can be found here: http://metamap.nlm.nih.gov/SemanticTypeMappings_2011AA.txt (full list) or shortened in appendix B

1. **OneSent:** one sequence consists of one sentence
2. **TwoSent:** one sequence consists of two sentences, where each second sentence is also taken as a first sentence in the next sequence
3. **Paragraph:** one sequence consists of one paragraph
4. **Paragraph&Sent** one sequence consists of one paragraph, where each sentence is an itemset[14]

All of them can be combined with all of the input sequence transformation methods. I expected that with method TwoSent more patterns might be learnt, while patterns learnt with OneSent might be more precise. With the method Paragraph I have wanted to test the idea whether the bigger context around the relation might help to identify it since the abstracts are structured similarly, e.g. with a diagnostic procedure coming before the treatment. Finally, with the method Paragraph&Sent I have wanted to exploit the bigger context while also trying to exploit sentence boundaries. I will get back to the input sequence methods in section 4.

In the next part, I will give some details on the input and output of the sequential pattern mining process.

3.3.2 Sequential Pattern Mining

Based on a short preliminary analysis of sequential patterns for each input sequence method[15], I start with a more thorough analysis for input sequence method OneSent. Therefore, I will give details on the sequential pattern mining process for OneSent combined with all type transformation methods in table 3. As I said before, the training set consists of 70 abstracts, and the algorithm BIDE+ implemented in SPMF has been used.

For all four combinations, the number of input sequences is 483, as this is the number of sentences in the training set. Depending on the type transformation strategy, i.e. depending on whether all MetaMap types are used or only types from the DISY and TRME group, they have a different number of items (i.e. semantic types and lemmata). This is due to the fact that a MetaMap type annotation often spans more than one word. This means that the more MetaMap types are used, the less lemmata remain. The number of unique items is different for each input since they not only have more or less lemmata, but also more or less semantic types (DISY and TRME as opposed to all types).

As the support threshold for all inputs, I chose 1%, which means that a pattern has to occur in at least four input sequences.

[14] this means that the sentences are ordered, while the tokens within each sentence are considered as unordered

[15] this will be described in detail in section 4

	OneSent AllTypes	OneSent RelevantTypes	OneSent OnlyTwo	OneSent Two&Others
Input				
No. sequences	483	483	483	483
Total no. items	4710	5235	5235	4710
Unique no. items	965	1312	1299	951
Support	0.01	0.01	0.01	0.01
Output				
No. patterns	1194	1124	1233	1319
Total no. items	2636	2391	2981	3313
Unique no. items	228	297	286	217
Average seq. length	2.2	2.1	2.4	2.5
Max. seq. length	5	5	6	6

Table 3: Details for input sequence method OneSent combined with all type transformation methods

After sequential pattern mining, every input results in a different number of sequential patterns. As suspected, the input types containing "DISY" and "TRME" result in more patterns. Of the two, type transformation method OnlyTwo&Others gives even more patterns than OnlyTwo. Therefore, the less variation there is due to the use of semantic types, the more patterns were found with the given data and support threshold.

An example pattern which has been learnt with method OneSent + OnlyTwo is the following:

<(patient) (TRME) (DISY)>[16]

Further patterns as well as their evaluation will be discussed in section 4.

Based on which type transformation method performs best together with input sequence method OneSent, the other input sequence methods have been combined only with that one. As I will show in section 4, this is type transformation method OnlyTwo. Therefore, I will give details for OnlyTwo combined with all the input sequence transformation methods in table 4.

The two combinations which use sentences have 483 input sequences again. The other two have 89 input sequences, as this is the number of paragraphs in the training set. Apart from method TwoSent, they have the same total number of items, and they all have the same number of

[16] as in section 2.3.3, the arrow brackets indicate the boundaries of a sequential pattern and the round brackets indicate the boundaries of an itemset

	OnlyTwo OneSent	OnlyTwo TwoSent	OnlyTwo Paragraph	OnlyTwo Paragraph&Sent
Input				
No. sequences	483	483	89	89
Total no. items	5235	9468	5235	5235
Unique no. items	1299	1299	1299	1299
Support	0.01	0.02	0.1	0.1
Output				
No. patterns	1233	1625	109	4778
Total no. items	2981	4657	624	20680
Unique no. items	286	197	14	91
Average seq. length	2.4	2.9	5.7	4.3
Max. seq. length	6	6	10	9

Table 4: Details for type transformation method OnlyTwo combined with all input sequence methods

unique items, as they all use the same type transformation method. The combination using TwoSent has almost twice as many total items[17].

As support thresholds, I chose 2% for method TwoSent (i.e. 8 sequences), and 10% for methods Paragraph and Paragraph&Sent (i.e. 8 sequences).

As I suspected, the combination using TwoSent results in more patterns than OneSent. The combination using the method Paragraph results in a much lower number of patterns and a very low number of unique items. This means that the patterns learnt from Paragraph contain many similar items in different orders. The combination with method Paragraph&Sent behaves similarly to Paragraph, with a higher number of patterns and a not much higher number of unique items. Again, patterns and results for this will be discussed in section 4.

3.3.3 Postprocessing with the Use of Constraints

Obviously, not all patterns that have been mined are to do with a treatment-relation. Examples are the two following patterns:

- `<(DISY) (month)>`
- `<(patient) (treat) (TRME)>`

In the first pattern, "DISY" occurs but "TRME" does not. In the second pattern, the verb *treat* and "TRME" occur but "DISY" does not. If one of

[17] the total number of items is not exactly twice as much because the first sentence of each abstract occurs only once, while the others occur twice

the entities that is supposed to be extracted does not occur in the pattern, it cannot be extracted.

Therefore, the patterns need to be constrained. Similarly to Cellier et al. (2010), I have decided that each pattern has to contain two named entities, where one belongs to the DISY group and one to the TRME group. Differently from them, however, patterns may also contain more than one named entity of each type. This is important since, as I mentioned in section 3.1, a relation may exist between more than one disease/syndrome and more than one treatment method. Furthermore, as in Cellier et al. (2010), each pattern has to contain a verb or a noun. This is based on the observation that the relation is mainly expressed through verbs and nouns, such as *treat* and *treatment*. The results of postprocessing with these constraints are shown in table 5.

	OneSent AllTypes	OneSent RelevantTypes	OneSent OnlyTwo	OneSent Two&Others
output before/after				
no. patterns	1194 / 26	1124 / 18	1233 / 33	1319 / 56
total no. items	2636 / 81	2391 / 57	2981 / 115	3313 / 195
unique no. items	228 / 10	297 / 11	286 / 15	217 / 15
average seq. length	2.2 / 3.1	2.1 / 3.2	2.4 / 3.5	2.5 / 3.5
max. seq. length	5 / 4	5 / 5	6 / 5	6 / 4
	OnlyTwo OneSent	OnlyTwo TwoSent	OnlyTwo Paragraph	OnlyTwo Paragraph&Sent
output before/after				
no. patterns	1233 / 33	1625 / 112	109 / 45	4778 / 1375
total no. items	2981 / 115	4657 / 424	624 / 302	20680 / 6943
unique no. items	286 / 15	197 / 26	14 / 11	91 / 35
average seq. length	2.4 / 3.5	2.9 / 3.8	5.7 / 6.7	4.3 / 5
max. seq. length	6 / 5	6 / 5	10 / 10	9 / 8

Table 5: Patterns before and after postprocessing

As before postprocessing, combinations containing "DISY" and "TRME" result in more patterns after postprocessing, as opposed to combinations which contain all the MetaMap types. Of the two, method OneSent + Two&Others has more patterns again than method OneSent + OnlyTwo. However, they have the same number of unique items. This means that they contain similar patterns where method OneSent + Two&Others will have additional patterns with different orders of the items. For method OnlyTwo + TwoSent, the number of patterns after postprocessing is much higher than the others and also has more unique items. Therefore, this will contain patterns which are not contained in the other sets. The method OnlyTwo + Paragraph&Sent has an even higher number of patterns after

postprocessing and a higher number of unique items, compared to the other methods.

To sum up, preprocessing mainly includes POS tagging and semantic typing, stop wording based on the POS tags, and then transformation of the text into different input sequences combined with different type transformation methods. Sequential patterns learnt in this way have been discussed by way of their quantity as well as total and unique number of items in them, while the actual patterns and a numerical evaluation will be discussed in section 4. Finally, postprocessed patterns include at least one "DISY", at least one "TRME" and at least one verb or noun.

3.4 Simple Method for Anaphora Resolution

Before I applied the sequential patterns to extract instances of the treatment-relation, I performed a simple anaphora resolution method. It is based on that of Pustejovsky et al. (2002).

Similar to Castaño et al. (2002), I restricted it to MetaMap types from the DISY and TRME groups. Furthermore, I restricted it to NP anaphora since, as I mentioned in section 3.1, almost none of the few occurring pronouns refers to a sign or treatment.

Firstly, I have made temporary annotations, marking NPs from the DISY or TRME group as "candidate antecedent" or "candidate anaphora". Candidate antecedents are NPs where the noun is preceded by the determiners *a*, *an* or which do not contain a determiner. Candidate anaphora are NPs where the noun is preceded by a determiner which is not *a* or *an*. However, in order to simplify the method, I have excluded nouns preceded by *both* and *neither* since those usually require more than one antecedent (as opposed to an antecedent in plural).

Next, I have used a Perl script to compare the string of the candidate anaphora to the strings of the candidate antecedents, starting from the candidate antecedent closest to the candidate anaphora going until the beginning of the given abstract. This is based on the observation that many of the anaphora are contained in their antecedent. An example of this is:

(14) *a subsequent brachial artery aneurysm ... the aneurysm*

For candidate anaphora which cannot be matched to an antecedent in this way, I have checked for matching semantic type and matching number. To reduce false positives, I have computed the string distance using Levenshtein distance. If the semantic types are an exact match, the string distance has to be less or equal to the length of the candidate anaphora string. If the semantic types match within their group (i.e. DISY or TRME), the string distance has to be less or equal to $\frac{3}{4}$ of the length of the candidate anaphora string[18]. An example of a pair which can be matched in this way is:

[18] this has been determined heuristically

(15) *thromboemboli ... the radial artery embolism.*

Matched pairs have then been annotated using a JAPE grammar.

The performance of this anaphora resolution method has been evaluated on the test set. In order to do so, the test set has been manually annotated with correct anaphora-antecedent pairs. In total, there are 15 pairs. 14 of those refer to entities from the DISY group and only one of them refers to an entity from the TRME group.

With the above described method, 17 pairs are identified. Out of them, 11 are correct and 6 are false positives. This gives a recall of 73%, precision of 65% and F_1-measure of 69%.

Of the missing pairs, three of them are absent because either the anaphora or antecedent has not been marked as a candidate in the first step. Of the false positives, three come from the string comparison part and the other three from the semantic type matching part. However, of the six false positives, none of them participates in a treatment-relation. Therefore, in this data, the medium precision of anaphora resolution will not influence the precision of relation extraction.

I will show in the next section that an anaphora resolution method with the given performance can already improve relation extraction.

To sum up, in this section I have described collection, division and characteristics of the data as well as which relation I aimed to extract. I have shortly described the algorithm BIDE+ used for sequential pattern mining. Then I have described my approach to extraction of the treatment-relation, and compared different sets of mined patterns based on numerical characteristics before and after postprocessing. Finally, I have described how I am resolving NP anaphora which have a type from the DISY or TRME group.

4 Evaluation

In this section, I will give a qualitative and quantitative evaluation of the sequential patterns described previously. Qualitative evaluation will be based on a manual analysis of patterns. For this, I have looked at the sequences from which the given pattern has been derived, i.e. at sequences from the training set. Quantitative evaluation will be based on precision, recall and F-measure computed on the test set.

For this purpose, I have created a reference set in which the treatment-relation relating a disease/syndrome with a treatment method is annotated. Furthermore, NP anaphora referring to diseases/syndromes and treatment methods are annotated with their antecedent. In total, there exist 16 treatment-relations in the test set. Of those, one relation contains more than one disease/syndrome, and two contain more than one treatment method. Six of the diseases/syndromes in a treatment-relation are expressed through

an anaphora. None of the treatment methods in a treatment-relation is expressed through an anaphora.

Evaluation against the reference set is done as follows: for a treatment-relation with one occurrence of "DISY" and "TRME" each, it is counted as 1 correct answer if the entities are connected to each other. For a treatment-relation with multiple occurrences of one or both of the entities, it is counted as 1 correct answer if all the entities are connected to each other. For example, if there are two "DISY" and one "TRME", it is counted as 1 correct answer if both "DISY" are connected to "TRME" and it is counted as 0.5 correct answer if only one of them is connected to it. Furthermore, if there is an anaphora present and the correct antecedent is present as well, this is counted as 1 correct answer. If the anaphora is present without its antecedent, this is counted as half a correct answer (based on the idea that an anaphora is still better to extract than completely missing the given relation).

In order to have a comparison for the quantitative evaluation, I have created a baseline result. For this, a JAPE grammar extracts a treatment-relation whenever a disease/syndrome and treatment method occur in the same sentence, in whatever order. Evaluated on the test set, this method gives a precision of 22%, a recall of 83% and an F_1-measure of 34%. I will refer to the F_1-measure as the baseline.

4.1 Preliminary Analysis of Sequential Patterns

OneSent vs. TwoSent patterns: Based on the observation that the treatment-relation mostly occurs either within one sentence or over two sentences, I have started with comparing patterns learnt with method OneSent to patterns learnt with TwoSent, both after postprocessing. For this, I have looked not only at the patterns itself, but also at the sequences in the training set from which the given pattern was learnt.

I have distinguished the sequences into relevant and irrelevant occurrences of the pattern. I define a relevant occurrence to be one where a correct treatment-relation can be extracted with the given pattern. Irrelevant occurrences are then defined as those where the entities extracted with the pattern are not related in a treatment-relation. Based on this, I have tried to find characteristics which distinguish relevant and irrelevant occurrences of a pattern.

For patterns from TwoSent, it can be seen that most relevant occurrences have the entities within one sentence, while many irrelevant occurrences have them over two sentences. For example, for pattern <`(TRME) (artery) (DISY)`> (pattern 2 in appendix F), this is true of sequence 176 and 206, respectively: sequence 176 has the entities in one sentence and is relevant, while sequence 206 has the entities over two sentences and is irrelevant[19].

[19]In each of the following sequences, the items of the pattern are marked in blue. For

TwoSent - 176: we describe a 57 - year-old woman who , on multiple occasions , presented with progressive gastrointestinal symptoms and eventually underwent surgical revascularization for celiac and superior mesenteric artery stenosis of uncertain etiology . her postoperative course was complicated by bowel ischemia , multiple organ failure , and death .

TwoSent - 206: angiographic control at the end of operation demonstrated a good reconstructive result without any changes in the right common iliac artery and the aorta . histopathological examination of the removed material showed fibromuscular dysplasia of the media .

For a sample of 10 patterns (two frequent, one medium and two rare patterns containing verbs, and the same for nouns), 73% of the relevant occurrences have the entities within one sentence, and 53% of the irrelevant occurrences have the entities over two sentences.

The pattern <`(TRME) (artery) (DISY)`> has 4 relevant occurrences and 36 irrelevant occurrences for the TwoSent input, where all the relevant occurrences have the entities within one sentence and 61% of the irrelevant occurrences have them in two sentences. On the other hand, the OneSent input has 2 relevant occurrences for the same pattern (pattern 3 in appendix C) and 9 irrelevant occurrences. An example of a relevant occurrence is sequence 176 and an example of an irrelevant occurrence is sequence 343.

OneSent - 176: we describe a 57 - year-old woman who , on multiple occasions , presented with progressive gastrointestinal symptoms and eventually underwent surgical revascularization for celiac and superior mesenteric artery stenosis of uncertain etiology .

OneSent - 343: this case report describes a patient with digital embolization from brachial artery fibromuscular dysplasia .

For both OneSent and TwoSent sequences, most of the irrelevant occurrences that have the entities within one sentence are irrelevant due to one of the entities being mistagged, i.e. being tagged as a treatment method when it is a syndrome or the other way around (e.g. sequence 343 for the OneSent input).

In sum, for the pattern <`(TRME) (artery) (DISY)`>, the coverage of relevant occurrences of the OneSent pattern is the same as for the TwoSent pattern[20]. At the same time, the accuracy of the OneSent pattern is better than that of the TwoSent pattern. Therefore, I assume that input method OneSent will perform better than method TwoSent.

relevant occurrences, this corresponds to the entities related in a treatment-relation, while for irrelevant occurrences it does not correspond to the correct entities. Furthermore, irrelevant occurrences are marked with a #

[20]as the 4 relevant occurrences from the TwoSent input come from the same two sentences as in the OneSent pattern

Paragraph and Paragraph&Sent: A preliminary analysis of the patterns learnt with Paragraph and Paragraph&Sent shows that the patterns mostly contain multiple occurrences of "DISY" and "TRME" and not many verbs or other nouns. An example of this is the following:

```
<(DISY) (undergo) (TRME) (TRME) (DISY)>
```

However, obviously not every occurrence participates in a treatment-relation since the occurrences are distributed over a whole paragraph. Therefore, in order to use those patterns, we would need a way of knowing which are the relevant occurrences. I will get back to this in section 4.4.

Quantitative evaluation of OneSent: Based on this preliminary analysis, I have continued to analyze patterns created with input method OneSent more closely. In order to decide with which type transformation method I should analyze it, I have evaluated OneSent combined with all type transformation methods on the test set. For all the patterns, all the items of the pattern have to occur within one sentence with an unlimited number of tokens in between. This gives the results in table 6.

	Precicion	Recall	F_1-measure
OneSent, AllTypes	21%	34%	26%
OneSent, RelevantTypes	21%	28%	24%
OneSent, OnlyTwo	**20%**	**44%**	**27%**
OneSent, Two&Others	16%	44%	24%

Table 6: Quantitative evaluation of OneSent patterns

Based on these results, I have decided to give a more detailed analysis for method OneSent + OnlyTwo in section 4.3. Based on the analysis, I will present ideas how to filter the patterns so that their precision will increase, as right now none of the results reaches the baseline result. Then I will apply these ideas to all the type transformation methods again since at least OnlyTwo and AllTypes have very similar results.

4.2 Impact of Anaphora Resolution on the Test Results

In this section, I will evaluate the impact of anaphora resolution on the extraction of the treatment-relation. By way of example, I have compared extraction with and without anaphora resolution with method OneSent + OnlyTwo.

Table 7 shows that both precision and recall are increased when anaphora resolution is performed. Therefore, all the following experiments have been performed with anaphora resolution, without explicitly stating it every time.

	Precicion	Recall	F_1-measure
without anaphora resolution	20%	44%	27%
with anaphora resolution	**22%**	**50%**	**31%**

Table 7: Impact of anaphora resolution for OneSent + OnlyTwo

4.3 Detailed Analysis of Sequential Patterns Created Using Most Relevant Method

For a detailed analysis of sequential patterns created with method OneSent + OnlyTwo, I have looked at all the sequences that were used in order to build the patterns. As I said in table 5, there are 33 patterns after postprocessing. They can be found in appendix C.

7 of them contain a verb and 26 contain a noun. There are five different verbs, with *treat* being used three times. There are five different nouns, with *patient*, *artery* and *case* being used multiple times. 5 of the noun patterns and one verb pattern contain two occurrences of "DISY", while 3 noun patterns and no verb pattern contain two occurrences of "TRME".

Again, I have distinguished the sequences for each pattern into relevant and irrelevant occurrences as described above and tried to find similarities and differences between them based on syntactic or semantic information.

Verb patterns: I have started with the analysis of the verb patterns. The five occurring verbs are *treat*, *undergo*, *describe*, *report* and *cause*. I have looked at the meaning of the verb, its voice, and the syntactic or semantic roles that the entities play with regard to the verb.

Firstly, by way of its meaning the verb *treat* is obviously able to relate two entities in a treatment-relation. The verb *undergo* is able to relate two entities in a treatment-relation as well, but could also be used in different ways (e.g. relating a disease/syndrome with a diagnostic procedure). Then, *describe* and *report* are both verbs of speech. Since they have a general meaning, i.e. not related to the biomedical domain, we have to check if and when they describe a treatment-relation. Finally, semantically the verb *cause* could describe a treatment-relation if "DISY" was the agent[21] and "TRME" was the theme[22].

The verb *treat* always occurs either as a main verb in passive voice or as a past participle. In all the relevant occurrences of its patterns, the "DISY"

[21] the semantic role *agent* expresses the doer of the action (here, the thing that causes something)

[22] the semantic role *theme* expresses something that undergoes an action while not changing in its course (here, the thing that is caused)

semantically expresses the patient[23] of *treat*, while "TRME" expresses its manner or instrument. The same semantic roles apply to the relevant occurrences of the patterns with *undergo*, *describe* and *report*, with all three of the verbs being in active voice in the sequences. For the pattern with *cause*, none of the sequences are relevant since in none of them "DISY" is the agent and "TRME" is the theme, as stipulated above. This analysis shows that for future work inclusion of semantic roles into the preprocessing and then into the sequential pattern mining might be very useful.

Syntactically, in the *treat* patterns "DISY" is usually expressed as an *of*-prepositional object, while "TRME" is usually expressed as the by-agent in a passive sentence or as a *with*-prepositional object. The verb *undergo* has a rigid syntactic structure: "TRME" is expressed as its direct object in all sequences and "DISY" is expressed as a prepositional object in all relevant occurrences. An example is sequence 176.

> OneSent - 176: we describe a 57 - year-old woman who , on multiple occasions , presented with progressive gastrointestinal symptoms and eventually underwent surgical revascularization for celiac and superior mesenteric artery stenosis of uncertain etiology .

In almost all relevant cases of *describe* and *report*, "DISY" is expressed as a prepositional object referring to the direct object of the verb (e.g. *case*, *management*) and "TRME" is expressed as a prepositional object referring to the direct object as well or to a past participle, such as *treated*. Examples are sequences 83 and 212, respectively.

> OneSent - 83: this case report describes management of a left renal artery aneurysm with covered stents .
> OneSent - 212: we report a case of symptomatic cervical carotid artery stenosis associated with fibromuscular dysplasia (fmd) successfully treated by percutaneous transluminal angioplasty (pta) .

As I said before, there are no relevant occurrences with *cause*, an example of which is sequence 270.

> # OneSent - 270: multiple medical regimens failed , including an angiotensin-converting enzyme inhibitor , which caused acute renal failure .

While there are syntactic similarities between relevant sequences of a given pattern, as described above, it depends on the support threshold

[23] the semantic role *patient* expresses something that undergoes an action while changing in its course

whether they can be exploited. Some of them will be too infrequent to be learnt.

The verb *treat* is the only one which occurs in more than one pattern and therefore the only one with different orders of items in the different patterns. For *treat*, different orders of "TRME" and "DISY" are mostly due to different emphasis.

- <`(treat) (TRME) (DISY)`> (pattern 5 in appendix C)
- <`(DISY) (DISY) (treat) (TRME)`> (pattern 29 in appendix C)

For example, for pattern 5 the emphasis seems to be on the treatment method, while for pattern 29 the emphasis seems to be on the diseases/ syndromes. However, there is also one pattern containing *treat* which has only irrelevant sequences:

<`(treat) (DISY) (TRME)`> (pattern 15 in appendix C)

Two of the irrelevant occurrences are due to mistagging, while the other two do not extract the correct disease/syndrome. Instead, the correct disease/syndrome occurs after "TRME" (as in pattern 5). This clearly shows that the order of items in the pattern matters and that using sequential pattern mining is a good choice, as opposed to itemset mining.

Finally, I have taken a closer look at pattern 29, which is the only verb pattern with multiple occurrences of a named entity. I have found out that both occurrences of "DISY" should only be extracted if they are connected by a conjunction such as *and*/*or*. On the contrary, if the first occurrence is connected to the second by a preposition or anything else, only the first occurrence should be extracted. Examples of this are the sequences 263 and 212, respectively.

> OneSent - 263: patients with fibromuscular dysplasia (fmd) and hypertension are frequently treated with percutaneous transluminal renal angioplasty (ptra) .
> # OneSent - 212: we report a case of symptomatic cervical carotid artery stenosis associated with fibromuscular dysplasia (fmd) successfully treated by percutaneous transluminal angioplasty (pta) .

While the relevant occurrences of the verb patterns have some things in common, as described above, this is not really the case for the irrelevant occurrences. On the one hand, the entities in them can be semantically and syntactically expressed as they are in the relevant occurrences, but do not relate the entities as they occur in the pattern. On the other hand, they can also be expressed differently from relevant occurrences. One thing the irrelevant occurrences have in common is that usually the gap between items, i.e. the number of words occurring between the items, is bigger than for relevant occurrences.

Noun patterns: There are noun patterns with the following nouns: *treatment*, *patient*, *artery* with or without modifiers, *case* and *type*.

As *treat*, the noun *treatment* is obviously able to relate two entities in a treatment-relation. Syntactically, "TRME" is usually the subject, *treatment* the direct object and "DISY" a prepositional object referring to *treatment*. An example is sequence 100.

> OneSent - 100: current percutaneous angioplasty is the preferred treatment for symptomatic carotid fmd , but no randomized controlled trials comparing this methodology with surgery is available .

On the other hand, the noun *type* does not have anything to do with the biomedical domain. A look at the sequences from which the pattern was learnt shows that none of them is relevant to the treatment-relation. This is due to "TRME" being related to another disease/syndrome than the one in the pattern. An example is sequence 302.

> # OneSent - 302: for uncontrolled hypertension , nephrectomy was performed and histopathology of the renal artery showed intimal fibroplasia , an uncommon type of fibromuscular dysplasia .

The other three nouns do have something to do with the biomedical domain, but as with the verbs of speech, it is not obvious in what way they can or cannot be used to extract a treatment-relation. Semantically, a patient is always explicitly or implicitly present in a sentence with a treatment-relation, but of course the word also frequently occurs in non-relevant sentences. The same holds for the word *case* and as long as we are talking about fibromuscular dysplasia, for the word *artery*.

Each of the three nouns occurs with more than one order of the noun, "DISY" and "TRME" in the pattern. When analyzing the sequences for each pattern, I have found out that most of them are not relevant to the treatment-relation. There are only three patterns for which most of the sequences are relevant:

- <`(case) (DISY) (TRME)`> (pattern 7 in appendix C)
- <`(artery) (DISY) (TRME)`> (pattern 12 in appendix C)
- <`(TRME) (DISY) (internal) (carotid) (artery)`> (pattern 16 in appendix C)

Examples are the sequences 212, 451 and 14, respectively.

OneSent - 212: we report a case of symptomatic cervical carotid artery stenosis associated with fibromuscular dysplasia (fmd) successfully treated by percutaneous transluminal angioplasty (pta) .

OneSent - 451: fibromuscular dysplasia (fmd) of the renal arteries is classically associated with secondary hypertension in younger individuals , which may be treatable and even curable by percutaneous transluminal renal angioplasty .

OneSent - 14: angioplasty may be used as an alternative to open arteriotomy and graduated dilatation in treating stenosis due to symptomatic fibromuscular dysplasia in the immediate extracranial part of the internal carotid artery .

Intuitionally, we can say that the disease/syndrome belongs to the patient/artery/case, while the treatment method is something which is done to them because of the disease/syndrome. Therefore, "DISY" occurs between the noun and "TRME" in relevant sequences.

As with verb patterns, if there is more than one occurrence of "DISY" or "TRME", then both occurrences should only be extracted if they are connected by a conjunction and otherwise only the first occurrence should be extracted. Also as with verb patterns, the number of words occurring between items is usually bigger for irrelevant occurrences than for relevant occurrences.

Apart from that, in all of the patterns (i.e. verb and noun patterns) about half of the irrelevant occurrences are irrelevant because one of the entities is mistagged. Usually, this is the case when a word is able to denote both a syndrome and a treatment method, such as I mentioned in section 2.2. Obviously, this influences both the learning phase and the testing phase. For learning, it means that patterns will be learnt which would not be learnt with correct tagging and the given support threshold. For testing, it means that the precision will possibly decrease.

Filtering: Based on the analysis of the verb and noun patterns produced by method OneSent + OnlyTwo, I conclude that the performance of these patterns for extraction of the treatment-relation will be better if the patterns are filtered, i.e. if patterns which introduce a lot of noise are excluded. This is also done by Cellier et al. (2010), who have their sequential patterns validated by a domain expert.

Therefore, firstly I have excluded the patterns with *cause*, *type* and the irrelevant *treat* pattern, as all their sequences are irrelevant. Furthermore, for noun patterns, I have excluded all patterns which do not have one of the following orders:

- `<(noun) (DISY) (TRME)>`
- `<(TRME) (DISY) (noun)>`

"DISY" or "TRME" may occur multiple consecutive times in one pattern and the noun may be modified with adjectives. If the entities do occur

multiple times, both occurrences are only extracted if they are connected by *and*, *or* or a comma. Only the first occurrence is extracted if the two occurrences are otherwise connected. Due to the fact that BIDE+ mines closed sequences, sometimes the same pattern with one occurrence instead of two occurrences is missing (namely when they have the same support). Therefore, I furthermore say that the second occurrence is optional.

The filtered set of patterns can be found in appendix D. On the test set, this method of filtering gives the results in table 8.

	Precicion	Recall	F_1-measure
OneSent, AllTypes	29%	31%	30%
OneSent, RelevantTypes	27%	25%	26%
OneSent, OnlyTwo	**30%**	**50%**	**37%**
OneSent, Two&Others	23%	56%	33%

Table 8: Quantitative evaluation of OneSent patterns after filtering

Table 8 shows that the precision increases for all methods in comparison with table 6. Except in one case, the recall does not decrease. Therefore, the filtering method is able to reduce noise while keeping almost all of the useful patterns. Furthermore, now one of the methods, OneSent + OnlyTwo, is able to outperform the baseline result.

Gap constraints: Next, since the gap between items from the pattern is usually bigger in irrelevant occurrences than in relevant occurrences, I have decided to implement a gap constraint for the patterns, i.e. to not allow an unlimited number of tokens between items. Using the development set, I heuristically determined the maximum number of tokens in the gap. This means that I have tried to maximize the precision on the development set, while keeping the recall as high as possible. This was done for each pattern separately. By way of example, the patterns with their corresponding gaps for method OneSent + OnlyTwo can be found in appendix E. Using the resulting patterns on the test set gives the results in table 9.

	Precicion	Recall	F_1-measure
OneSent, AllTypes	50%	31%	38%
OneSent, RelevantTypes	60%	19%	29%
OneSent, OnlyTwo	**50%**	**50%**	**50%**
OneSent, Two&Others	35%	50%	41%

Table 9: Quantitative evaluation of OneSent patterns after filtering and with gap constraints

Table 9 shows that the precision is much higher for all methods, while the recall decreases a little for method OneSent + RelevantTypes and OneSent

+ Two&Others. This still results in a higher F_1-measure for all methods. Furthermore, now all methods except for OneSent + RelevantTypes are able to outperform the baseline result. Finally, both in table 8 and 9, the superiority of method OneSent + OnlyTwo over the other methods is more pronounced than it was in table 6.

4.4 Short Analysis of Other Sequential Patterns

Based on the results from the previous section, I have decided to test the other input methods with the type transformation method OnlyTwo.

TwoSent + OnlyTwo: The method TwoSent + OnlyTwo includes the patterns of OneSent + OnlyTwo, but it also has additional ones. They can be found in appendix F. As I said in section 4.1, the sequences are usually relevant when the entities occur in only one of the sentences, while they are usually irrelevant when they occur over two sentences.

The patterns include a new verb, *perform*, which semantically looks promising. However, analysis of its patterns and their sequences shows that most of them are irrelevant. Even though there are also relevant sequences occurring in one sentence, those are not frequent enough to be learnt with the method OneSent + OnlyTwo and with the given support threshold. Examples of a relevant and irrelevant occurrence are sequences 25 and 35, respectively.

> TwoSent - 25: extracranial-intracranial bypass graft was performed , followed by excision of the arterial lesion . pathological examination revealed fibromuscular dysplasia and dissecting aneurysm .
>
> # TwoSent - 35: a retrospective analysis of all patients with renal artery rafmd who underwent transcatheter therapy between january 1999 and december 2009 was performed . blood pressure (bp) measurement , number of bp medications , and hypertension defined by a systolic bp >140 diastolic bp >90 were recorded .

Before testing the TwoSent + OnlyTwo patterns on the test set, I have applied filtering to them in the same way as for OneSent + OnlyTwo. The set of filtered patterns can be found in appendix G. I have compared the performance of a gap constraint of 0 to 20 tokens[24] to the gap constraint adjusted using the development set. The results of this are shown in table 10.

It can be seen that even though the recall is higher than for method OneSent + OnlyTwo, the precision is lower, resulting in a lower F_1-measure.

[24] since the average sentence length is close to that and since there are usually two gaps, the first item in the pattern might be at the beginning of one sentence and the last item at the end of the following sentence with this gap constraint

	Precicion	Recall	F_1-measure
gap 0-20	16%	81%	27%
gap adjusted on d.s.	**33%**	**56%**	**42%**

Table 10: Quantitative evaluation of TwoSent + OnlyTwo after filtering

Paragraph + OnlyTwo: Patterns by strategy Paragraph + OnlyTwo mainly contain multiple occurrences of "DISY" or "TRME", but not many different verbs or nouns. They can be found in appendix H. Since the occurrences are distributed over a whole paragraph, not each of them participates in the treatment-relation. I have analyzed two patterns more closely to see if there is a way of telling which occurrences of "DISY" and "TRME" should be extracted.

- <`(DISY) (undergo) (TRME) (DISY) (DISY)`> (pattern 16 in appendix H)
- <`(DISY) (report) (DISY) (DISY) (DISY) (DISY) (artery) (TRME) (DISY)`> (pattern 45 in appendix H)

In five sequences of pattern 16, only the first occurrence of "DISY" is relevant, where in two of those "DISY" is not in the same sentence as *undergo* and "TRME" (e.g. sequence 24). In four sequences, only the second occurrence of "DISY" is relevant and it is always in the same sentence as *undergo* and "TRME". One sequence is completely irrelevant.

> # Paragraph - 24: fibromuscular dysplasia of the aorta is an exceedingly rare disease with 26 cases reported to date in medline . we present a case of stenosis of the infrarenal abdominal aorta in a 49 - year-old woman with a history of intermittent claudication . the patient underwent aortic endarterectomy , and subsequent anatomopathologic examination of the specimen revealed fibromuscular dysplasia . the possible causes of aortic stenosis in this case , its angiographic findings , and the alternatives of treatment are discussed .

For pattern 45, in two sequences, only the last occurrence of "DISY" is relevant, and it is in the same sentence with "TRME" but with nothing else from the pattern. In three sequences, either the forth, fifth or both occurrences of "DISY" are relevant, where one of them is in the same sentence with "TRME" and the others are not.

This short analysis already shows that in order for the patterns to be useful for relation extraction, each one would have to be analyzed with respect to which entities participate in the relation and which ones are context. Furthermore, even within one pattern, there is variation with respect

to that. Therefore, I conclude that it would take too much work to analyze the patterns before the context can be useful.

Paragraph&Sent + OnlyTwo: Patterns learnt by Paragraph&Sent + OnlyTwo will also require too much work. Since there are still 1375 patterns after postprocessing, only a sample of them can be found in appendix I. In fact, those patterns might require even more work than the method Paragraph + OnlyTwo since while the sentences are ordered in them, the words are considered as unordered. As we have seen in section 4.3 that not every order is equally good, these patterns will introduce a lot of noise.

Therefore, I conclude that the idea of using paragraphs, even though it might make sense in theory, is not practical without extra-work. Even then, the precision and recall still might be worse than with OneSent or TwoSent patterns.

To sum up, I have firstly shown that anaphora resolution can improve both precision and recall for extraction of the treatment-relation. Then, based on preliminary test results, I have analyzed the patterns of the method OneSent + OnlyTwo as the most promising method.

Based on the analysis, I have come up with a way of filtering the patterns and constraining the gap between items in the pattern. Furthermore, I have discovered some linguistic considerations, which will be interesting for future work. Firstly, labeling with grammatical roles, such as subject and object, is promising. However, the grammatical roles are not completely fixed for each pattern and could therefore not be exploited for all of them. Secondly, for verb patterns semantic role labeling would be interesting. It is not clear, however, if and how this could help with noun patterns.

With both filtering and gap constraints, the F_1-measure has reached 50% for method OneSent + OnlyTwo. This remains the best result, after also having filtered and constrained the gaps of the other, less promising methods.

5 Discussion and Conclusion

Some points in the previous work need improvement or further discussion. Firstly, the dataset seems to be fairly limited. For example, promising patterns, such as with the verb *perform*, are only learnt with method TwoSent and not with the best method, i.e. OneSent. While the pattern does not help method TwoSent to outperform method OneSent, it is a pattern we intuitionally would like to keep. Some other verbs and nouns occurring in the test set are too infrequent in the training set to be learnt as a pattern with any given method and given support threshold, e.g. *manage* and *management*. Furthermore, some of the verbs occur with different orders of the

named entities, while only one of them has been learnt as a pattern, e.g. the verb *undergo*.

Therefore, the performance of relation extraction would probably be better if the dataset was bigger. Since I have taken all the existing PubMed abstracts talking about treatment of fibromuscular dysplasia, the only possibility would be to mix abstracts talking about different diseases. This would be interesting for future work.

Then with a bigger dataset, the importance of anaphora resolution is not entirely clear. While it is able to improve the performance of relation extraction with the given test set, this might not be the case for a bigger one. This is due to the assumption that in a bigger dataset, there would be redundancy in the instances of the treatment-relation. Not every anaphora would have to be resolved then in order to capture each instance of the relation.

Furthermore, with a bigger dataset, it might be possible that the postprocessed patterns have to be reduced in a more efficient way than manual filtering. It would be interesting to see how recursion as proposed by Cellier et al. (2010) works. However, since they only give examples of final patterns and not of originally learnt patterns, it is hard to know if this is a useful technique. The original patterns, even though more numerous, might be more accurate.

Moreover, the gap constraints have been adjusted for each pattern by hand. It would be better if this could be done automatically, so that we can be sure that the best possible performance is reached.

Finally, the best F_1-measure that has been reached is only 50%. This is much less than the results by Cellier et al. (2010), where the best F_1-measure is 88%. Obviously, the results are not entirely comparable since they have not been created on the same test set.

Taking into account the limitations of the data set, the ceiling imposed by the semantic tagging and the manual adjustment of the gap constraints, I think that the approach described here is useful and can be improved by taking care of the above things.

To sum up, in this thesis my aim was to extract instances of a treatment-relation between treatment methods and syndromes of fibromuscular dysplasia. The motivation for this is the need to create review articles on rare diseases, in order to make information on them more accessible. Therefore, we would like to automatically extract information for the articles.

I have developed relation extraction rules for the treatment-relation, using sequential pattern mining. The preprocessing for this includes POS tagging, lemmatization, named entity recognition using MetaMap tagging, and anaphora resolution using mostly string comparison. The learnt patterns are postprocessed using constraints. Furthermore, based on a manual analysis, the postprocessed patterns are filtered and gap constraints are im-

plemented.

The best method makes use of grouped MetaMap tags which are replaced by "DISY" or "TRME", and single sentences as the input for sequential pattern mining. Its F_1-measure reaches 50%. While this result is not as high as, for example, the one by Cellier et al. (2010), who also use sequential pattern mining, I think it can be improved by gathering more data and using an automatic method for adjusting gaps. Furthermore, it would be interesting to use grammatical role labeling and semantic role labeling during preprocessing and see how they can improve the results. Given that, I think that sequential pattern mining is a useful method for creating relation extraction rules.

References

R. Agrawal and R. Srikant. Mining sequential patterns. In *International Conference on Data Engineering*, 1995.

R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, 1993.

R. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

J. Castaño, J. Zhang, and J. Pustejovsky. Anaphora resolution in biomedical literature. In *International Symposium on Reference Resolution*, 2002.

P. Cellier, T. Charnois, M. Plantevit, and B. Crémilleux. Recursive sequence mining to discover named entity relations. In *The Ninth International Symposium on Intelligent Data Analysis (IDA 2010)*, pages 30–41. Springer-Verlag, 2010.

H. Cherfi, A. Napoli, and Y. Toussaint. Toward a text mining methodology using frequent itemset and association rule extraction. In *Journées de l'informatique Messine (JIM-2003)*, pages 285–294, 2003.

A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71, 2005.

M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press, 1999.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: an architecture for development of robust HLT applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175. Association for Computational Linguistics, 2002.

H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (Version 6)*. 2011.

M. Delgado, M. J. Martín-Bautista, D. Sánchez, and M. A. Vila. Mining text data: Special features and patterns. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, pages 140–153. Springer-Verlag, 2002.

European Commission. Rare diseases - policy | public health. Website, 2012. Available online at `http://ec.europa.eu/health/rare_diseases/policy/index_en.htm`; visited on June, 14th 2012.

M. A. Hearst. Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 3–10. Association for Computational Linguistics, 1999.

K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 5, pages 502–513, 2000.

R. J. Kate and R. J. Mooney. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 203–212. Association for Computational Linguistics, 2010.

J.-H. Kim, A. Mitchell, T. K. Attwood, and M. Hilario. Learning to extract relations for protein annotation. *Bioinformatics*, 23:i256–i263, 2007.

T. Liang and Y.-H. Lin. Anaphora resolution for biomedical literature by exploiting multiple resources. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, pages 742–753. Springer-Verlag, 2005.

R. J. Mooney and R. Bunescu. Mining knowledge from text using information extraction. In *SIGKDD Explorations*, volume 7.1, pages 3–10, 2005.

U. Y. Nahm and R. J. Mooney. Using soft-matching mined rules to improve information extraction. In *Proceedings of the AAAI-2004 Workshop on Adaptive Text Extraction and Mining (ATEM-2004)*, pages 27–32, 2004.

C. Nédellec. Machine learning for information extraction in genomics - state of the art and perspectives, text mining and its applications. In S. Sirmakessis, editor, *Results of the NEMIS Launch Conference Series: Studies in Fuzziness and Soft Computing*, pages 99–118. Springer-Verlag, 2004.

G. Nenadić, H. Mima, I. Spasić, S. Ananiadou, and J. ichi Tsujii. Terminology-driven literature mining and knowledge acquisition in biomedicine. *International Journal of Medical Informatics*, 67:33–48, 2002.

T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17:155–161, 2001.

M. Plantevit, T. Charnois, J. Kléma, C. Rigotti, and B. Crémilleux. Combining sequence and itemset mining to discover named entities in biomedical texts: a new type of pattern. *International Journal of Data Mining, Modelling and Management*, 1(2):119–148, 2009.

P.-F. Plouin, J. Perdu, A. L. Batide-Alanore, P. Boutouyrie, A.-P. Gimenez-Roqueplo, and X. Jeunemaitre. Fibromuscular dysplasia. *Orphanet Journal of Rare Diseases*, 2(28), 2007.

J. Pustejovsky, J. Castafio, J. Zhang, M. Kotecki, and B. Cochran. Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 7, pages 362–373, 2002.

G. Schneider, K. Kaljurand, and F. Rinaldi. Detecting protein-protein interactions in biomedical texts using a parser and linguistic resources. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 406–417. Springer-Verlag, 2009.

U.S. National Library of Medicine. MEDLINE citation counts by year of publication. Website, 2012. Available online at `http://www.nlm.nih.gov/bsd/medline_cit_counts_yr_pub.html`; visited on June, 14th 2012.

J. Wang and J. Han. BIDE: Efficient mining of frequent closed sequences. In *Proceedings of the 20th International Conference on Data Engineering*, ICDE '04. IEEE Computer Society, 2004.

X. Yang, J. Su, G. Zhou, and C. L. Tan. An NP-cluster based approach to coreference resolution. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04. Association for Computational Linguistics, 2004.

A Semantic Groups of MetaMap Types

Note: this is a shortened list of semantic groups, while the full list can be found here: http://metamap.nlm.nih.gov/SemGroups_2011.txt

CHEM—Chemicals & Drugs—T116—Amino Acid, Peptide, or Protein
CHEM—Chemicals & Drugs—T195—Antibiotic
CHEM—Chemicals & Drugs—T123—Biologically Active Substance
CHEM—Chemicals & Drugs—T122—Biomedical or Dental Material
CHEM—Chemicals & Drugs—T118—Carbohydrate
CHEM—Chemicals & Drugs—T103—Chemical
CHEM—Chemicals & Drugs—T120—Chemical Viewed Functionally
CHEM—Chemicals & Drugs—T104—Chemical Viewed Structurally
CHEM—Chemicals & Drugs—T200—Clinical Drug
CHEM—Chemicals & Drugs—T111—Eicosanoid
CHEM—Chemicals & Drugs—T196—Element, Ion, or Isotope
CHEM—Chemicals & Drugs—T126—Enzyme
CHEM—Chemicals & Drugs—T131—Hazardous or Poisonous Substance
CHEM—Chemicals & Drugs—T125—Hormone
CHEM—Chemicals & Drugs—T129—Immunologic Factor
CHEM—Chemicals & Drugs—T130—Indicator, Reagent, or Diagnostic Aid
CHEM—Chemicals & Drugs—T197—Inorganic Chemical
CHEM—Chemicals & Drugs—T119—Lipid
CHEM—Chemicals & Drugs—T124—Neuroreactive Substance or Biogenic Amine
CHEM—Chemicals & Drugs—T114—Nucleic Acid, Nucleoside, or Nucleotide
CHEM—Chemicals & Drugs—T109—Organic Chemical
CHEM—Chemicals & Drugs—T115—Organophosphorus Compound
CHEM—Chemicals & Drugs—T121—Pharmacologic Substance
CHEM—Chemicals & Drugs—T192—Receptor
CHEM—Chemicals & Drugs—T110—Steroid
CHEM—Chemicals & Drugs—T127—Vitamin
DEVI—Devices—T203—Drug Delivery Device
DEVI—Devices—T074—Medical Device
DEVI—Devices—T075—Research Device
DISO—Disorders—T020—Acquired Abnormality
DISO—Disorders—T190—Anatomical Abnormality
DISO—Disorders—T049—Cell or Molecular Dysfunction
DISO—Disorders—T019—Congenital Abnormality
DISO—Disorders—T047—Disease or Syndrome
DISO—Disorders—T050—Experimental Model of Disease
DISO—Disorders—T033—Finding
DISO—Disorders—T037—Injury or Poisoning
DISO—Disorders—T048—Mental or Behavioral Dysfunction
DISO—Disorders—T191—Neoplastic Process

DISO—Disorders—T046—Pathologic Function
DISO—Disorders—T184—Sign or Symptom
PROC—Procedures—T060—Diagnostic Procedure
PROC—Procedures—T065—Educational Activity
PROC—Procedures—T058—Health Care Activity
PROC—Procedures—T059—Laboratory Procedure
PROC—Procedures—T063—Molecular Biology Research Technique
PROC—Procedures—T062—Research Activity
PROC—Procedures—T061—Therapeutic or Preventive Procedure

B Mappings from Abbreviations to Full MetaMap Types

Note: this is a shortened list of mappings, while the full list can be found here: http://metamap.nlm.nih.gov/SemanticTypeMappings_2011AA.txt

acab—Acquired Abnormality
anab—Anatomical Abnormality
antb—Antibiotic
bodm—Biomedical or Dental Material
cgab—Congenital Abnormality
clnd—Clinical Drug
comd—Cell or Molecular Dysfunction
drdd—Drug Delivery Device
dsyn—Disease or Syndrome
emod—Experimental Model of Disease
fndg—Finding
inpo—Injury or Poisoning
medd—Medical Device
mobd—Mental or Behavioral Dysfunction
neop—Neoplastic Process
patf—Pathologic Function
phsu—Pharmacologic Substance
sosy—Sign or Symptom
topp—Therapeutic or Preventive Procedure

C Patterns OneSent + OnlyTwo after Postprocessing

1. <(patient) (TRME) (DISY)> SID: 343 479 204 374 66 311 232 468 47 223 158 332 56 268 SUP: 14

2. <(patient) (DISY) (TRME)> SID: 221 204 35 375 223 445 263 311 173 127 332 435 SUP: 12
3. <(TRME) (artery) (DISY)> SID: 343 201 217 202 128 214 176 56 12 390 302 SUP: 11
4. <(TRME) (DISY) (artery)> SID: 137 236 217 232 432 128 468 390 271 14 151 SUP: 11
5. <(treat) (TRME) (DISY)> SID: 479 204 311 294 163 144 407 448 SUP: 8
6. <(case) (TRME) (DISY)> SID: 343 236 338 202 22 432 454 439 SUP: 8
7. <(case) (DISY) (TRME)> SID: 236 200 83 261 212 127 454 245 SUP: 8
8. <(TRME) (patient) (DISY)> SID: 394 374 144 42 433 57 211 SUP: 7
9. <(artery) (TRME) (DISY)> SID: 394 21 113 123 467 407 390 SUP: 7
10. <(patient) (TRME) (DISY) (DISY)> SID: 479 223 66 468 56 268 SUP: 6
11. <(TRME) (DISY) (patient)> SID: 394 374 294 42 264 57 SUP: 6
12. <(artery) (DISY) (TRME)> SID: 356 173 113 83 451 303 SUP: 6
13. <(TRME) (carotid) (DISY)> SID: 100 217 128 214 12 390 SUP: 6
14. <(undergo) (TRME) (DISY)> SID: 260 158 232 468 176 SUP: 5
15. <(treat) (DISY) (TRME)> SID: 204 375 311 294 144 SUP: 5
16. <(TRME) (DISY) (internal) (carotid) (artery)> SID: 217 432 128 390 14 SUP: 5
17. <(describe) (DISY) (TRME)> SID: 173 83 127 176 245 SUP: 5
18. <(TRME) (internal) (carotid) (artery) (DISY)> SID: 217 128 214 12 390 SUP: 5
19. <(case) (DISY) (DISY) (TRME)> SID: 261 212 127 454 245 SUP: 5
20. <(patient) (TRME) (TRME) (DISY)> SID: 479 374 223 232 332 SUP: 5
21. <(patient) (DISY) (DISY) (TRME)> SID: 223 263 127 332 435 SUP: 5
22. <(TRME) (type) (DISY)> SID: 415 214 302 439 SUP: 4
23. <(case) (TRME) (TRME) (DISY)> SID: 236 338 22 439 SUP: 4
24. <(TRME) (DISY) (DISY) (artery)> SID: 236 128 468 14 SUP: 4
25. <(TRME) (DISY) (renal) (artery)> SID: 137 236 468 271 SUP: 4
26. <(report) (DISY) (TRME)> SID: 236 261 212 332 SUP: 4
27. <(DISY) (patient) (TRME) (DISY)> SID: 374 332 56 268 SUP: 4
28. <(TRME) (TRME) (DISY) (artery)> SID: 236 232 390 14 SUP: 4
29. <(DISY) (DISY) (treat) (TRME)> SID: 263 212 435 245 SUP: 4
30. <(DISY) (DISY) (TRME) (artery)> SID: 137 203 10 435 SUP: 4
31. <(TRME) (treatment) (DISY)> SID: 255 204 100 454 SUP: 4
32. <(TRME) (DISY) (patient) (DISY)> SID: 394 374 42 57 SUP: 4
33. <(TRME) (cause) (DISY)> SID: 21 270 12 151 SUP: 4

D Patterns OneSent + OnlyTwo after Filtering

Note: Since I define that the second occurrence of "DISY" or "TRME" is optional, I delete the pattern with a single occurrence when both occur (i.e. when they do not have the same support). This is done simply to avoid redundancy.

1. <(treat) (TRME) (DISY)> SID: 479 204 311 294 163 144 407 448 SUP: 8
2. <(TRME) (DISY) (patient)> SID: 394 374 294 42 264 57 SUP: 6
3. <(artery) (DISY) (TRME)> SID: 356 173 113 83 451 303 SUP: 6
4. <(undergo) (TRME) (DISY)> SID: 260 158 232 468 176 SUP: 5
5. <(TRME) (DISY) (internal) (carotid) (artery)> SID: 217 432 128 390 14 SUP: 5
6. <(describe) (DISY) (TRME)> SID: 173 83 127 176 245 SUP: 5
7. <(case) (DISY) (DISY) (TRME)> SID: 261 212 127 454 245 SUP: 5
8. <(patient) (DISY) (DISY) (TRME)> SID: 223 263 127 332 435 SUP: 5
9. <(TRME) (DISY) (DISY) (artery)> SID: 236 128 468 14 SUP: 4
10. <(TRME) (DISY) (renal) (artery)> SID: 137 236 468 271 SUP: 4
11. <(report) (DISY) (TRME)> SID: 236 261 212 332 SUP: 4
12. <(TRME) (TRME) (DISY) (artery)> SID: 236 232 390 14 SUP: 4
13. <(DISY) (DISY) (treat) (TRME)> SID: 263 212 435 245 SUP: 4
14. <(TRME) (treatment) (DISY)> SID: 255 204 100 454 SUP: 4

E Patterns OneSent + OnlyTwo after Filtering and with Gap constraints

Note: the number of tokens allowed in the gap is simply written in between the items, where " * " stands for "0 or more" and [x,y] expresses a range from x to y. As I said in section 4.3, this number has been heuristically determined on the development set for each pattern.

1. <(treat) * (TRME) * (DISY)>
2. <(TRME) [0,4] (DISY) [0,21] (patient)>
3. <(artery) [0,2] (DISY) [0,10] (TRME)>
4. <(undergo) [0,7] (TRME) [0,7] (DISY)>
5. <(TRME) * (DISY) * (internal) * (carotid) * (artery)>
6. <(describe) * (DISY) * (TRME)>
7. <(case) [0,4] (DISY) [0,1] (DISY) [0,11] (TRME)>

8. <(patient) [0,11] (DISY) [0,1] (DISY) [0,14] (TRME)>
9. <(TRME) [0,2] (DISY) [0,1] (DISY) [0,5] (artery)>
10. <(TRME) [0,4] (DISY) [0,4] (renal) [0,1] (artery)>
11. <(report) * (DISY) * (TRME)>
12. <(TRME) [0,1] (TRME) [0,4] (DISY) [0,3] (artery)>
13. <(DISY [0,1]) (DISY) [0,11] (treat) [0,7] (TRME)>
14. <(TRME) [0,6] (treatment) [0,5] (DISY)>

F Patterns TwoSent + OnlyTwo after Postprocessing

1. <(patient) (TRME) (DISY)> SID: 478 343 479 342 204 474 65 66 203 468 13 194 259 154 222 223 157 263 158 267 332 265 331 268 373 35 375 374 445 311 310 232 231 108 41 46 437 47 55 127 56 423 SUP: 42
2. <(TRME) (artery) (DISY)> SID: 343 342 139 272 206 201 203 202 128 12 285 393 154 18 216 398 217 82 213 214 147 451 390 389 328 210 175 232 432 108 55 430 127 176 56 420 301 122 183 302 SUP: 40
3. <(artery) (DISY) (TRME)> SID: 342 69 201 143 203 202 9 12 21 216 397 83 82 146 94 451 149 441 173 310 172 162 437 356 288 358 116 113 355 112 247 419 56 123 301 302 121 422 303 SUP: 39
4. <(patient) (DISY) (TRME)> SID: 479 204 273 477 474 203 468 220 221 259 222 223 263 267 332 331 373 441 35 375 374 444 445 173 311 172 310 42 433 435 41 437 376 294 355 127 56 481 SUP: 38
5. <(artery) (TRME) (DISY)> SID: 342 474 201 65 143 203 467 407 394 393 154 21 216 20 147 451 390 389 210 368 370 310 432 162 315 437 356 252 113 112 127 58 56 123 122 301 121 SUP: 37
6. <(TRME) (DISY) (artery)> SID: 343 137 136 139 201 202 128 468 199 13 14 220 18 216 217 82 214 270 390 271 389 150 448 151 236 232 432 231 433 356 359 431 294 127 122 302 SUP: 36
7. <(TRME) (patient) (DISY)> SID: 478 272 203 143 13 220 394 154 393 144 210 211 373 375 374 445 370 42 432 433 41 49 294 57 56 481 183 420 SUP: 28
8. <(DISY) (artery) (TRME) (DISY)> SID: 342 474 65 203 143 407 393 21 20 451 210 389 310 370 432 315 437 356 252 127 58 56 301 122 121 SUP: 25
9. <(treat) (TRME) (DISY)> SID: 478 479 204 442 375 311 143 203 447 310 163 162 407 194 356 289 393 18 263 293 294 144 212 448 SUP: 24
10. <(TRME) (DISY) (patient)> SID: 373 204 32 374 272 415 370 432 42 41 12 439 394 393 49 154 293 263 294 332 57 264 56 147 SUP: 24
11. <(case) (DISY) (TRME)> SID: 236 101 65 200 143 202 432 199 288 157 397 260 83 355 261 82 127 212 454 453 125 146 245 422 SUP: 24
12. <(DISY) (DISY) (TRME) (artery)> SID: 137 342 69 136 101 175 203 202 9 468 10 435 107 284 55 397 127 247 213 419 451 301 209 448 SUP: 24
13. <(DISY) (DISY) (artery) (TRME)> SID: 101 369 65 143 203 9 468 435 107 284 288 252 20 397 294 355 127 58 419 149 209 302 448 SUP: 23

14. <(DISY) (artery) (DISY) (TRME)> SID: 441 342 143 203 310 172 9 437 288 358 116 21 397 355 82 419 56 451 149 301 121 302 422 SUP: 23

15. <(case) (TRME) (DISY)> SID: 343 342 236 201 65 338 143 337 202 432 438 439 21 157 431 22 261 127 212 454 453 SUP: 21

16. <(patient) (TRME) (TRME) (DISY)> SID: 478 373 479 35 374 232 231 108 41 13 154 222 223 263 127 332 56 331 423 SUP: 19

17. <(patient) (DISY) (DISY) (TRME)> SID: 373 479 441 374 42 468 435 437 376 259 222 223 263 355 127 267 332 56 331 SUP: 19

18. <(patient) (TRME) (DISY) (DISY)> SID: 478 373 479 374 65 66 232 468 41 13 154 222 223 55 158 127 267 56 268 SUP: 19

19. <(TRME) (artery) (DISY) (DISY)> SID: 272 201 202 128 393 154 55 216 398 217 176 56 147 420 390 183 301 328 302 SUP: 19

20. <(artery) (TRME) (DISY) (DISY)> SID: 368 65 143 370 394 393 154 252 21 216 113 112 127 56 147 123 390 301 389 SUP: 19

21. <(DISY) (patient) (TRME) (DISY)> SID: 373 342 375 374 65 203 310 231 46 437 222 55 157 267 332 56 331 423 268 SUP: 19

22. <(artery) (DISY) (DISY) (TRME)> SID: 69 143 202 9 162 437 288 116 397 113 355 112 247 56 94 149 302 422 SUP: 18

23. <(DISY) (TRME) (artery) (DISY)> SID: 342 175 201 203 393 55 216 398 127 213 176 56 451 301 389 210 302 SUP: 17

24. <(DISY) (DISY) (patient) (TRME)> SID: 373 69 342 375 374 65 203 42 107 55 157 263 267 332 331 423 422 SUP: 17

25. <(DISY) (DISY) (treat) (TRME)> SID: 441 442 374 143 203 172 162 435 288 392 263 293 355 212 245 302 448 SUP: 17

26. <(case) (DISY) (DISY) (TRME)> SID: 101 65 143 202 288 157 260 397 261 355 127 212 454 453 245 422 SUP: 16

27. <(patient) (DISY) (TRME) (DISY)> SID: 373 35 375 474 445 41 437 259 222 223 263 127 267 332 56 331 SUP: 16

28. <(TRME) (DISY) (internal) (artery)> SID: 136 432 433 128 13 14 359 216 217 431 127 214 390 389 448 SUP: 15

29. <(TRME) (DISY) (carotid) (artery)> SID: 432 433 128 13 14 18 359 216 217 431 127 214 390 389 448 SUP: 15

30. <(DISY) (TRME) (patient) (DISY)> SID: 373 375 374 445 143 203 370 432 42 41 393 294 144 56 210 SUP: 15

31. <(present) (DISY) (TRME)> SID: 175 173 172 202 381 288 259 157 397 454 213 453 146 176 422 SUP: 15

32. <(TRME) (carotid) (DISY)> SID: 100 99 432 128 12 18 216 217 430 127 213 214 390 328 389 SUP: 15

33. <(DISY) (treat) (TRME) (DISY)> SID: 442 375 143 203 310 163 162 407 356 289 393 263 293 144 212 SUP: 15

34. <(artery) (DISY) (TRME) (DISY)> SID: 342 201 143 162 437 356 21 216 113 112 56 451 123 301 121 SUP: 15

35. <(woman) (DISY) (TRME)> SID: 101 474 175 278 41 259 252 358 157 397 58 176 149 269 SUP: 14

36. <(TRME) (DISY) (patient) (DISY)> SID: 373 272 374 370 432 42 41 394 49 154 393 294 57 56 SUP: 14

37. <(TRME) (TRME) (DISY) (artery)> SID: 136 236 232 231 13 14 18 294 127 214 270 390 150 389 SUP: 14

38. <(treat) (DISY) (TRME)> SID: 479 204 375 442 374 143 203 311 310 263 293 294 144 448 SUP: 14

39. <(TRME) (DISY) (DISY) (artery)> SID: 136 236 202 128 468 13 14 359 294 127 270 390 302 448 SUP: 14

40. <(DISY) (patient) (DISY) (TRME)> SID: 373 273 375 374 203 172 310 42 437 222 294 332 56 331 SUP: 14

41. <(TRME) (DISY) (internal) (carotid) (artery)> SID: 432 128 433 13 14 359 216 217 431 127 214 390 389 448 SUP: 14

42. <(patient) (TRME) (DISY) (TRME)> SID: 373 479 204 374 203 311 310 468 41 222 223 263 332 56 SUP: 14

43. <(TRME) (DISY) (artery) (DISY)> SID: 139 201 232 432 128 18 216 217 82 127 122 390 302 389 SUP: 14

44. <(angiography) (DISY) (TRME)> SID: 96 97 231 9 132 437 438 220 116 428 209 301 210 SUP: 13

45. <(TRME) (carotid) (artery) (DISY)> SID: 432 128 12 18 216 217 430 127 213 214 390 328 389 SUP: 13

46. <(undergo) (TRME) (DISY)> SID: 35 175 232 231 108 468 259 154 157 458 158 260 176 SUP: 13

47. <(DISY) (TRME) (DISY) (artery)> SID: 137 136 236 231 356 216 294 431 127 270 150 302 389 SUP: 13

48. <(TRME) (DISY) (TRME) (patient)> SID: 373 204 32 374 415 42 432 41 12 293 294 264 56 SUP: 13

49. <(renal) (artery) (DISY) (TRME)> SID: 342 397 83 310 82 247 56 451 301 149 302 422 SUP: 12

50. <(cause) (DISY) (TRME)> SID: 356 358 116 21 278 355 77 146 270 12 149 151 SUP: 12

51. <(TRME) (TRME) (artery) (DISY)> SID: 18 201 398 232 127 213 214 108 420 390 389 285 SUP: 12

52. <(TRME) (perform) (DISY)> SID: 136 35 359 279 21 398 261 25 24 301 328 302 SUP: 12

53. <(case) (TRME) (DISY) (DISY)> SID: 236 338 65 337 143 202 432 212 127 454 438 439 SUP: 12

54. <(DISY) (DISY) (TRME) (patient)> SID: 373 375 374 415 293 263 203 143 144 247 332 56 SUP: 12

55. <(artery) (DISY) (TRME) (artery)> SID: 356 342 69 216 201 397 202 247 9 419 451 301 SUP: 12

56. <(TRME) (examination) (DISY)> SID: 206 279 398 368 157 158 370 163 25 183 328 285 SUP: 12

57. <(TRME) (TRME) (patient) (DISY)> SID: 373 394 393 154 374 294 432 42 56 41 420 210 SUP: 12

58. <(TRME) (left) (artery) (DISY)> SID: 18 216 398 217 82 213 214 420 390 389 302 285 SUP: 12

59. <(TRME) (internal) (carotid) (artery) (DISY)> SID: 216 217 430 432 127 213 128 214 12 390 389 328 SUP: 12

60. <(TRME) (TRME) (DISY) (patient)> SID: 394 393 154 32 374 263 42 332 56 41 439 SUP: 11

61. <(DISY) (artery) (DISY) (DISY) (TRME)> SID: 288 116 397 143 355 9 56 437 149 302 422 SUP: 11

62. <(patient) (DISY) (DISY) (TRME) (DISY)> SID: 373 259 222 223 263 127 267 332 56 437 331 SUP: 11

63. <(artery) (TRME) (TRME) (DISY)> SID: 394 393 252 154 21 127 58 123 390 389 210 SUP: 11

64. <(patient) (DISY) (TRME) (TRME)> SID: 373 441 204 35 222 223 263 127 42 332 331 SUP: 11

65. <(DISY) (DISY) (artery) (TRME) (artery)> SID: 101 397 127 9 468 419 435 209 107 448 284 SUP: 11

66. <(describe) (DISY) (TRME)> SID: 175 173 83 355 172 82 127 315 176 245 107 SUP: 11

67. <(renal) (artery) (TRME) (DISY)> SID: 394 342 393 252 154 474 310 56 451 467 301 SUP: 11

68. <(year) (woman) (DISY) (TRME)> SID: 259 252 101 175 278 157 397 58 176 149 269 SUP: 11

69. <(case) (DISY) (artery) (TRME)> SID: 288 101 200 65 397 355 432 127 199 146 422 SUP: 11

70. <(patient) (DISY) (patient) (TRME)> SID: 373 204 375 222 477 374 263 42 267 332 331 SUP: 11

71. <(present) (TRME) (DISY)> SID: 259 175 157 65 66 370 454 213 453 176 13 SUP: 11

72. <(DISY) (DISY) (patient) (TRME) (DISY)> SID: 373 342 375 374 55 157 65 203 267 331 423 SUP: 11

73. <(TRME) (case) (DISY)> SID: 100 359 223 201 21 260 370 82 199 124 438 SUP: 11

74. <(TRME) (DISY) (artery) (TRME)> SID: 220 294 82 432 468 199 271 122 302 448 151 SUP: 11

75. <(TRME) (patient) (DISY) (DISY)> SID: 478 394 49 374 143 144 42 41 420 183 13 SUP: 11

76. <(TRME) (renal) (artery) (DISY)> SID: 393 154 139 272 55 398 82 56 451 301 302 SUP: 11

77. <(report) (TRME) (DISY)> SID: 373 342 236 65 261 212 332 331 438 439 SUP: 10

78. <(patient) (TRME) (TRME) (DISY) (DISY)> SID: 478 479 154 374 223 232 127 56 41 13 SUP: 10

79. <(treat) (TRME) (DISY) (TRME)> SID: 479 204 442 293 263 311 294 203 310 448 SUP: 10

80. <(woman) (TRME) (DISY)> SID: 259 252 474 175 157 58 176 41 407 269 SUP: 10
81. <(present) (DISY) (DISY) (TRME)> SID: 259 175 157 397 202 381 454 213 453 422 SUP: 10
82. <(patient) (DISY) (DISY) (DISY) (TRME)> SID: 373 259 222 374 223 263 355 42 332 331 SUP: 10
83. <(DISY) (TRME) (DISY) (patient)> SID: 373 393 374 293 263 370 42 332 56 41 SUP: 10
84. <(DISY) (renal) (artery) (DISY) (TRME)> SID: 342 397 310 82 56 451 301 149 302 422 SUP: 10
85. <(angiography) (TRME) (DISY)> SID: 220 97 231 213 214 437 301 132 438 210 SUP: 10
86. <(TRME) (reveal) (DISY)> SID: 414 415 398 157 260 158 232 25 270 271 SUP: 10
87. <(reveal) (DISY) (TRME)> SID: 216 415 260 261 231 9 271 209 389 210 SUP: 10
88. <(DISY) (DISY) (artery) (TRME) (TRME)> SID: 252 397 203 127 58 419 149 107 448 284 SUP: 10
89. <(TRME) (show) (DISY)> SID: 375 206 108 124 56 316 301 302 285 378 SUP: 10
90. <(report) (DISY) (TRME)> SID: 373 358 236 65 260 261 212 332 94 331 SUP: 10
91. <(artery) (TRME) (artery) (DISY)> SID: 342 216 201 127 147 451 390 301 389 210 SUP: 10
92. <(case) (DISY) (TRME) (DISY)> SID: 236 157 65 143 261 432 212 127 454 453 SUP: 10
93. <(DISY) (artery) (TRME) (DISY) (DISY)> SID: 393 252 21 65 143 370 127 56 301 389 SUP: 10
94. <(TRME) (DISY) (case)> SID: 100 223 201 21 260 158 82 280 124 438 SUP: 10
95. <(treat) (TRME) (DISY) (DISY)> SID: 478 479 393 18 294 143 212 163 144 448 SUP: 10
96. <(artery) (DISY) (treat) (TRME)> SID: 356 288 441 173 143 355 172 162 302 303 SUP: 10
97. <(TRME) (DISY) (TRME) (artery)> SID: 136 201 294 202 433 468 270 271 448 SUP: 9
98. <(perform) (TRME) (DISY)> SID: 137 136 35 279 21 22 25 24 135 SUP: 9
99. <(patient) (DISY) (treat) (TRME)> SID: 441 477 374 263 173 355 172 433 435 SUP: 9
100. <(DISY) (DISY) (patient) (TRME) (TRME)> SID: 373 374 203 42 332 331 107 423 422 SUP: 9
101. <(show) (DISY) (TRME)> SID: 337 112 162 125 56 376 301 302 284 SUP: 9
102. <(year) (TRME) (DISY)> SID: 478 373 220 479 65 294 41 46 407 SUP: 9
103. <(DISY) (treat) (DISY) (TRME)> SID: 442 375 374 293 263 203 143 310 144 SUP: 9
104. <(woman) (DISY) (TRME) (DISY)> SID: 259 252 474 175 157 58 176 41 269 SUP: 9
105. <(TRME) (treatment) (DISY)> SID: 254 255 204 154 100 99 203 454 453 SUP: 9
106. <(TRME) (associate) (DISY)> SID: 236 223 431 143 430 432 144 265 167 SUP: 9
107. <(TRME) (type) (DISY)> SID: 220 414 415 213 214 301 438 302 439 SUP: 9

108. <(patient) (DISY) (DISY) (TRME) (TRME)> SID: 373 441 222 223 263 127 42 332 331 SUP: 9

109. <(case) (TRME) (TRME) (DISY)> SID: 236 21 338 337 22 432 127 438 439 SUP: 9

110. <(report) (DISY) (DISY) (TRME)> SID: 373 358 65 260 261 212 332 94 331 SUP: 9

111. <(patient) (DISY) (artery) (TRME)> SID: 220 441 221 474 294 355 127 468 435 SUP: 9

112. <(patient) (undergo) (TRME) (DISY)> SID: 259 35 154 157 158 232 231 468 108 SUP: 9

G Patterns TwoSent + OnlyTwo after Filtering

1. <(artery) (DISY) (DISY) (TRME)> SID: 69 143 202 9 162 437 288 116 397 113 355 112 247 56 94 149 302 422 SUP: 18

2. <(DISY) (DISY) (treat) (TRME)> SID: 441 442 374 143 203 172 162 435 288 392 263 293 355 212 245 302 448 SUP: 17

3. <(case) (DISY) (DISY) (TRME)> SID: 101 65 143 202 288 157 260 397 261 355 127 212 454 453 245 422 SUP: 16

4. <(TRME) (DISY) (internal) (artery)> SID: 136 432 433 128 13 14 359 216 217 431 127 214 390 389 448 SUP: 15

5. <(TRME) (DISY) (carotid) (artery)> SID: 432 433 128 13 14 18 359 216 217 431 127 214 390 389 448 SUP: 15

6. <(woman) (DISY) (TRME)> SID: 101 474 175 278 41 259 252 358 157 397 58 176 149 269 SUP: 14

7. <(TRME) (TRME) (DISY) (artery)> SID: 136 236 232 231 13 14 18 294 127 214 270 390 150 389 SUP: 14

8. <(TRME) (DISY) (DISY) (artery)> SID: 136 236 202 128 468 13 14 359 294 127 270 390 302 448 SUP: 14

9. <(TRME) (DISY) (internal) (carotid) (artery)> SID: 432 128 433 13 14 359 216 217 431 127 214 390 389 448 SUP: 14

10. <(angiography) (DISY) (TRME)> SID: 96 97 231 9 132 437 438 220 116 428 209 301 210 SUP: 13

11. <(undergo) (TRME) (DISY)> SID: 35 175 232 231 108 468 259 154 157 458 158 260 176 SUP: 13

12. <(renal) (artery) (DISY) (TRME)> SID: 342 397 83 310 82 247 56 451 301 149 302 422 SUP: 12

13. <(TRME) (perform) (DISY)> SID: 136 35 359 279 21 398 261 25 24 301 328 302 SUP: 12

14. <(artery) (DISY) (TRME) (artery)> SID: 356 342 69 216 201 397 202 247 9 419 451 301 SUP: 12

15. <(TRME) (TRME) (DISY) (patient)> SID: 394 393 154 32 374 263 42 332 56 41 439 SUP: 11

16. <(describe) (DISY) (TRME)> SID: 175 173 83 355 172 82 127 315 176 245 107 SUP: 11

17. <(year) (woman) (DISY) (TRME)> SID: 259 252 101 175 278 157 397 58 176 149 269 SUP: 11

18. <(present) (TRME) (DISY)> SID: 259 175 157 65 66 370 454 213 453 176 13 SUP: 11
19. <(report) (TRME) (DISY)> SID: 373 342 236 65 261 212 332 331 438 439 SUP: 10
20. <(present) (DISY) (DISY) (TRME)> SID: 259 175 157 397 202 381 454 213 453 422 SUP: 10
21. <(patient) (DISY) (DISY) (DISY) (TRME)> SID: 373 259 222 374 223 263 355 42 332 331 SUP: 10
22. <(reveal) (DISY) (TRME)> SID: 216 415 260 261 231 9 271 209 389 210 SUP: 10
23. <(TRME) (DISY) (case)> SID: 100 223 201 21 260 158 82 280 124 438 SUP: 10
24. <(treat) (TRME) (DISY) (DISY)> SID: 478 479 393 18 294 143 212 163 144 448 SUP: 10
25. <(artery) (DISY) (treat) (TRME)> SID: 356 288 441 173 143 355 172 162 302 303 SUP: 10
26. <(perform) (TRME) (DISY)> SID: 137 136 35 279 21 22 25 24 135 SUP: 9
27. <(patient) (DISY) (treat) (TRME)> SID: 441 477 374 263 173 355 172 433 435 SUP: 9
28. <(show) (DISY) (TRME)> SID: 337 112 162 125 56 376 301 302 284 SUP: 9
29. <(TRME) (treatment) (DISY)> SID: 254 255 204 154 100 99 203 454 453 SUP: 9
30. <(patient) (DISY) (DISY) (TRME) (TRME)> SID: 373 441 222 223 263 127 42 332 331 SUP: 9
31. <(report) (DISY) (DISY) (TRME)> SID: 373 358 65 260 261 212 332 94 331 SUP: 9
32. <(patient) (DISY) (artery) (TRME)> SID: 220 441 221 474 294 355 127 468 435 SUP: 9
33. <(patient) (undergo) (TRME) (DISY)> SID: 259 35 154 157 158 232 231 468 108 SUP: 9

H Patterns Paragraph + OnlyTwo after Postprocessing

1. <(DISY) (report) (DISY) (DISY) (TRME)> SID: 35 64 11 46 73 15 50 49 18 55 24 57 29 28 61 SUP: 15
2. <(DISY) (report) (DISY) (DISY) (DISY) (TRME)> SID: 35 64 11 46 73 15 50 49 18 55 57 29 28 61 SUP: 14
3. <(DISY) (internal) (carotid) (artery) (TRME) (DISY)> SID: 0 35 32 3 65 77 46 74 15 20 25 57 62 SUP: 13
4. <(DISY) (report) (TRME) (DISY) (DISY)> SID: 35 49 64 55 76 59 24 57 11 46 28 15 SUP: 12
5. <(DISY) (internal) (carotid) (artery) (DISY) (TRME)> SID: 0 35 32 3 65 25 77 57 46 62 74 15 SUP: 12
6. <(DISY) (internal) (carotid) (artery) (TRME) (TRME) (DISY)> SID: 0 35 32 3 65 20 25 77 57 46 15 SUP: 11
7. <(DISY) (report) (DISY) (artery) (TRME)> SID: 50 35 18 64 55 57 11 29 73 61 15 SUP: 11

8. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (DISY)> SID: 0 35 32 3 65 20 25 77 46 62 15 SUP: 11

9. <(DISY) (report) (case) (TRME) (DISY)> SID: 50 35 49 76 59 24 57 11 46 61 15 SUP: 11

10. <(DISY) (report) (DISY) (DISY) (DISY) (DISY) (TRME)> SID: 50 49 64 55 57 11 46 29 73 61 15 SUP: 11

11. <(DISY) (undergo) (TRME) (DISY)> SID: 35 16 81 64 23 6 24 27 11 46 47 SUP: 11

12. <(DISY) (report) (DISY) (DISY) (DISY) (artery) (TRME)> SID: 50 18 64 55 57 11 29 73 61 15 SUP: 10

13. <(DISY) (internal) (carotid) (artery) (DISY) (TRME) (DISY)> SID: 35 32 3 65 25 77 57 46 62 15 SUP: 10

14. <(DISY) (report) (DISY) (DISY) (TRME) (DISY) (DISY)> SID: 35 49 64 55 24 57 11 46 28 15 SUP: 10

15. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (artery)> SID: 0 35 32 3 65 20 77 62 74 15 SUP: 10

16. <(DISY) (undergo) (TRME) (DISY) (DISY)> SID: 35 16 64 23 6 24 27 11 46 47 SUP: 10

17. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (DISY) (DISY) (DISY)> SID: 0 35 32 3 65 20 25 77 46 15 SUP: 10

18. <(DISY) (report) (artery) (TRME) (DISY)> SID: 50 35 64 55 57 11 29 28 61 15 SUP: 10

19. <(DISY) (internal) (carotid) (artery) (DISY) (TRME) (artery)> SID: 0 35 32 3 65 77 62 74 15 SUP: 9

20. <(DISY) (report) (DISY) (DISY) (DISY) (TRME) (artery)> SID: 50 35 18 64 57 11 28 61 15 SUP: 9

21. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (TRME)> SID: 0 32 65 25 77 57 46 74 15 SUP: 9

22. <(DISY) (report) (artery) (DISY) (DISY) (TRME)> SID: 35 18 64 55 57 11 73 28 61 SUP: 9

23. <(DISY) (report) (TRME) (DISY) (DISY) (DISY)> SID: 35 64 76 59 57 11 46 28 15 SUP: 9

24. <(DISY) (report) (DISY) (DISY) (DISY) (DISY) (artery) (TRME)> SID: 50 64 55 57 11 29 73 61 15 SUP: 9

25. <(DISY) (report) (DISY) (DISY) (DISY) (TRME) (DISY) (DISY)> SID: 35 49 64 55 57 11 46 28 15 SUP: 9

26. <(DISY) (report) (case) (DISY) (DISY) (TRME) (DISY)> SID: 50 35 49 24 57 11 46 61 15 SUP: 9

27. <(DISY) (internal) (carotid) (artery) (DISY) (DISY) (TRME) (TRME) (DISY)> SID: 35 32 3 65 25 77 57 46 15 SUP: 9

28. <(DISY) (internal) (carotid) (artery) (DISY) (TRME) (DISY) (DISY)> SID: 35 32 3 65 25 77 46 62 15 SUP: 9

29. <(DISY) (report) (DISY) (artery) (TRME) (DISY)> SID: 50 35 64 55 57 11 29 61 15 SUP: 9

30. <(DISY) (internal) (carotid) (artery) (TRME) (TRME) (DISY) (artery)> SID: 0 35 32 3 65 20 77 15 SUP: 8

31. <(DISY) (report) (TRME) (DISY) (DISY) (DISY) (DISY)> SID: 35 64 76 57 11 46 28 15 SUP: 8

32. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (carotid) (artery)> SID: 32 3 65 20 77 62 74 15 SUP: 8

33. <(DISY) (report) (present) (DISY) (DISY) (TRME)> SID: 35 64 55 24 11 46 29 73 SUP: 8

34. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (DISY) (DISY) (DISY) (DISY)> SID: 0 32 3 65 20 25 77 15 SUP: 8

35. <(DISY) (internal) (carotid) (artery) (TRME) (TRME) (DISY) (DISY) (DISY) (DISY)> SID: 0 35 3 65 20 25 77 15 SUP: 8

36. <(DISY) (report) (DISY) (DISY) (TRME) (DISY) (case)> SID: 50 35 49 55 24 57 46 15 SUP: 8

37. <(DISY) (report) (artery) (DISY) (DISY) (DISY) (TRME)> SID: 18 64 55 57 11 73 28 61 SUP: 8

38. <(DISY) (report) (DISY) (artery) (TRME) (TRME)> SID: 50 35 18 64 55 57 11 15 SUP: 8

39. <(DISY) (internal) (carotid) (artery) (TRME) (DISY) (DISY) (artery)> SID: 0 32 3 65 20 77 62 15 SUP: 8

40. <(DISY) (report) (DISY) (artery) (DISY) (DISY) (TRME)> SID: 35 18 64 55 57 11 73 61 SUP: 8

41. <(DISY) (undergo) (TRME) (TRME) (DISY)> SID: 35 16 81 64 23 6 46 47 SUP: 8

42. <(DISY) (perform) (TRME) (DISY)> SID: 50 49 2 3 21 6 57 31 SUP: 8

43. <(DISY) (report) (TRME) (artery) (DISY)> SID: 50 35 64 59 57 11 28 15 SUP: 8

44. <(DISY) (report) (patient) (TRME) (DISY) (DISY)> SID: 35 64 55 59 24 57 11 46 SUP: 8

45. <(DISY) (report) (DISY) (DISY) (DISY) (DISY) (artery) (TRME) (DISY)> SID: 50 64 55 57 11 29 61 15 SUP: 8

I Patterns Paragraph&Sent + OnlyTwo after Post-processing

Note: This sample includes a random selection of patterns with support 8.

1. <(report) (DISY artery) (TRME) (DISY)> SID: 50 32 64 55 11 29 61 15 SUP: 8

2. <(treat TRME) (TRME DISY) (DISY)> SID: 51 32 64 65 25 77 88 47 SUP: 8

3. <(DISY treat) (TRME DISY) (DISY)> SID: 51 32 64 65 25 77 47 13 SUP: 8

4. <(DISY fmd) (DISY) (TRME) (DISY TRME)> SID: 32 2 52 46 88 47 13 15 SUP: 8

5. <(year-old DISY) (artery) (artery) (TRME) (TRME)> SID: 32 2 18 21 65 25 43 63 SUP: 8

6. <(DISY fmd) (DISY TRME) (DISY)> SID: 2 65 23 25 9 78 46 15 SUP: 8
7. <(year-old DISY) (artery) (DISY) (TRME) (patient)> SID: 32 71 18 21 65 57 63 31 SUP: 8
8. <(DISY) (DISY) (TRME DISY) (TRME artery)> SID: 32 48 21 23 77 11 74 31 SUP: 8
9. <(internal carotid) (DISY) (DISY) (DISY) (TRME) (DISY)> SID: 35 32 65 25 77 57 46 15 SUP: 8
10. <(DISY artery) (artery DISY) (TRME)> SID: 19 22 25 78 11 12 28 15 SUP: 8
11. <(artery) (DISY) (TRME) (DISY) (DISY artery)> SID: 32 2 3 21 77 63 31 15 SUP: 8