RWTH AACHEN UNIVERSITY

MASTER THESIS

Converting histological records into structured data by using a Dependency Grammar

Author:
Julian DÖRENBERG

Examiner:
Prof. Dr. W. VAN DER AALST [1]
Prof. Dr. E. DAHL [2]

[1] Chair for Process- and Data Science, RWTH Aachen University
[2] Institute of Pathology, University Hospital Aachen

March 14, 2022

"You cannot hope to build a better world without improving the individuals. To that end, each of us must work for his own improvement and, at the same time, share a general responsibility for all humanity, our particular duty being to aid those to whom we think we can be most useful."

- Marie Curie

Acknowledgement

First and foremost I have to thank Univ.-Prof. Dr. rer. nat. Edgar Dahl and Univ.-Prof. Prof. hc. Dr. ir. Dr. hc. Wil van der Aalst who gave me the opportunity to put my ideas regarding this project into practice. I would also like to show my gratitude to the whole team of the RWTH centralized Biomaterialbank (RWTH cBMB), above all Ms. Jennifer Wipperfürth for offering me the necessary freedoms to write this thesis besides my work there. Last but not least I thank Univ.-Prof. Dr. nat. med. Dr. med. Nadine Gaisa for her support in relation to medical content-related questions.

Abstract

The availability of structured data is becoming an increasingly critical factor for today's medical research. In cancer research, data from histological reports are of special interest. Still, pathologists in Germany often document their findings in flowing text. In order to make these high-quality data ready to be processed by computers it is critical to convert them to a structured form. This thesis aims to describe and implement a model which performs relation extraction in three steps. After preprocessing a report, it's sentences are parsed into a tree of grammatical relations by using a Dependency Grammar parser. As an alternative to Dependency Grammars, Link Grammars are presented and their disadvantages are substantiated. Finally, the grammatical relations returned by the Dependency Grammar parser are filtered by using regular expressions and the ontology database Unified Medical Language System (UMLS). This approach then is evaluated for the performance of the Dependency Grammar parser as well aus for the performance of UMLS. The Dependency Grammar parser achieved scores of 94% for Unlabelled Attachment Score, 92% for Labelled Accuracy and 90% for Labelled Attachment Score on a corpus of 200 sentences randomly selected from a corpus of 205 reports (3195 sentences in total) diagnosing breast biopsies. These scores show that Dependency Grammars successfully can be used for parsing histological reports into a structured form. The German UMLS instance is evaluated by classifying words of the corpus as either medical or non-medical. It reached a recall score of 0.22, which shows that 22% of the medical terms were correctly classified as medical. The precision score was 0.66 and indicates that 66% of the non-medical terms were correctly classified as non-medical. The f1 score as the harmonic mean of the two previous scores was 0.33. These three scores show that UMLS currently does not provide sufficient performance to extract structured data from German histological reports. Hence, alternatives are discussed in the outlook of this thesis. Eventually, the whole approach including the filtering by using regular expressions was evaluated. To do so, UMLS errors were corrected manually. Eventually, the whole On a corpus of ten histological reports where ten different information were to be extracted, the approach extracted 98% of the information correctly.

Contents

Chapter 1

Introduction

1.1 State of the Art

Structured data take a more and more important position in today's medical research. This is due to the increasing popularity of Artificial intelligence in general and machine learning models in particular and their demand for sufficiently extensive training data. Hence, data need to be available in high quantities. In particular, data from the clinical praxis as a secondary use [23] are of interest as these are well-documented by default. In the context of biomaterial banking biomaterial samples become more useful for research purposes if they are associated with high-quality data such as histological reports. Although synoptic reporting [18] is on the rise in European medical research, it is still common practice among German pathologists to document their findings as histological reports in flowing text.

Although there already are multiple implementations for text mining tools in biomedical use cases [25, 29, 8, 1] there was no tool available addressing German histological reports. Most of the tools base on English medical reports. Due to the linguistic conditions in the English language and differences to other languages it is not always possible to use a tool developed for the English language on reports in other languages [1, 29]. For very specific information there already are tools with satisfyingly performance [1]. But if a general secondary use of German histological reports extracting all possible information is wanted, there currently is no tool available. The main part behind this work was inspired by X. Zhiu, H. Han, I. Chankai A. Prestud and A. Brooks [31]. Their approach utilizing an ontology database on the semantic and a link grammar on the syntactical side can be adapted to German reports. In general, ontology-based approaches are quite common in text mining [2, 25]. There are multiple databases available such as the Unified Medical Language System (UMLS) [3] and SNOMED CT [26] which was licensed for Germany by 1st of January 2021, but does not have a German instance yet. Hence, SNOMED CT does not contain German words. In order to be able to query such a database it is important to be able to preprocess the reports accordingly. If they are not semi-structured and contain grammatically complete sentences tools like the framework GATE [7] which can be used for English as well as for German can be used for tokenization and lemmatisation, although GATE just lemmatises German nouns. As shown in Chapter 2 neither of these conditions holds for the particular use case of this thesis and hence a more individual approach was chosen.

Following the approach from [31] for syntactical relation extraction, a link grammar parser is used. Link grammars are a concept introduced by D. Sleator and D. Temperley [24] in order to be able to parse English texts. Although there already is an implementation of such an algorithm [24] available, a number of changes to the formalism as well as to the parser are required. This is caused by the properties of German and grammatically incomplete texts [19]. This will be discussed further in Chapter 4. In order to receive a grammar the parser can make use of it is necessary to train it as the one given by Temperley and Sleator is created for English texts only. To do this, an unsupervised approach introduces by S. Kübler is used [19]. Kübler already described the most important changes to the link grammar formalism necessary for German and implemented the training algorithm based on German text and given word classes. Her quite generic algorithm is used as the basis for the algorithm implemented for this thesis which is presented in Chapter 4. Unsupervised approaches for text mining in general and for link grammars in particular have the advantage that they need less training data as a supervised approach [31].

A related grammar type are Dependency Grammars. In opposition to Link Grammars, there already is an implementation that supports parsing German available. This parser which is part of the Supar framework [27] supports different neural nets for Dependency Grammar parsing [11, 30]. All the models can be trained via Pytorch [22]. Additionally, a number of pretrained models can be downloaded. In opposition to the Link Grammar parser, neural nets support parsing of sentences containing unknown words by using a respective tag for unknown words in their word embeddings. There are current approaches to use these kinds of Grammars in order to parse German clinical texts such as by Kara et. al. [17]. At the moment, these tools still lack in precision on German medical texts making research in this field even more important.

1.2 Aims of this thesis

In order to make information from histological reports available for medical research, it is necessary to convert them from natural language to a structured form. In order to make information available for a wide range of currently undefined use cases, interoperability is crucial. Hence, the data are kept interoperable by being represented as relations. One example for such a relation is the size – German adjective: *groß* – of a carcinoma – German: *Karzinom* – can be given as
(Karzinom, groß, 3.5cm). This thesis aims to describe and implement a tool which performs relation extraction in three steps [31]. Figure 1.1 shows the workflow the tool implements.

At first, the report text is preprocessed in order to handle semi-structures such as enumerations and identify measurement values such as $3,4cm$ and abbreviations such as $v.A.$. For further processing, the text is split into sentences which are further split into words. Words of particular interest are identified within the text using the Unified Medical Language System

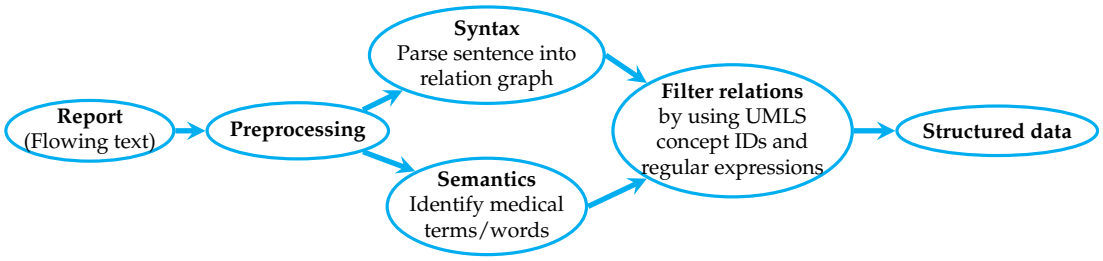


FIGURE 1.1: The pipeline that extracts relations from German histological reports.

database (UMLS)[3] by lemmatising each word within the text and querying UMLS for it. In the second step, the tool looks for grammatical relations between the words. This is either done by using a Link Grammar parser [24] or a Dependency Grammar parser [11]. In this thesis, Link Grammars and Dependency Grammars are compared and multiple models of Dependency Grammars are evaluated for the best performance. The parsing produces a graph fully connecting the words of each sentence. In the third step, relations are then extracted by traversing the relation graph and computing relations of different arity.

In the third step, the relations are filtered by the process described in Chapter 5. This is done by using regular expressions as well as the ontology database UMLS. This final step also determines the arity of the relations to be generated from the relation graph.

After presenting related work and the aims of this thesis here, it goes on by discussing the steps mentioned above. This is split into four main Chapters. At first, the preprocessing of histological reports is explained together with a number of examples as a motivation in Chapter 2. In Chapter 3, the ontology database UMLS, which is used to filter the reports for medical words, is presented. Its performance is evaluated on histological reports. Chapter 4 presents Link Grammars as well as Dependency Grammars together with their respective training and parsing methods. The disadvantages of Link Grammars are discussed. The Dependency Grammar model implemented by Dozat and Manning in 2017 is evaluated on a histological corpus in the same Chapter [11]. In Chapter 5, semantic and syntactic information is combined to extract information from the histological reports. This Chapter also provides an evaluation of the whole tool based on data manually extracted from a histological corpus. The thesis eventually is concluded by an outlook describing limitations of the approach and future work.

Chapter 2

The environment

In order to understand the steps required to process the histological reports before informations can be extracted, it is crucial to examine them closer. These usually are structured in multiple Sections namely *Klinische Angaben, Mikroskopische Begutachtung, Makroskopische Begutachtung, Molekularpathologische Begutachtung, Schnellschnittdiagnose and Gutachten*. After talking to several pathologists the *Gutachten* Section was identified as the only relevant one because it contains the most important informations regarding the diagnosis for the patient and the oncologist. The other Sections are discarded for the purpose of this thesis. Hence, the term *histological report* refers to the *Gutachten* Section exclusively. Nevertheless, the system also works for the other Sections as they have the same linguistic and structural properties as the *Gutachten* Section. This Section contains the diagnoses together with additional information. Diagnoses are usually described in flowing text and are also encoded as ICD-O codes [12], which can be extracted using regular expressions. Extracting well-defined codes by using regular expressions is implemented in the tool, but will not be discussed in this thesis, because it is trivial.

2.1 Histological reports compared to standard German texts

Histological reports at the Institute of Pathology at the University Hospital Aachen are generated from dictations by the pathologists. These dictations are converted to written text by typists. These make sure that the spelling is correct, but do not do any changes to the content of the dictation. Hence, we can assume a high quality regarding spelling and a grammatical quality and style depending on the particular pathologist. Native German speakers might have another dictation style than foreign speakers. But even within these two groups there can be different styles. However, this thesis focuses on dictations by native speakers, because that reduces the amount of required evaluation data.

After reading a couple of reports relevant differences between grammatically complete German texts and the pathologists dictations become clear: The first thing to be mentioned is the semi-structure. Quite often there are

enumerations of different types. Besides the classical enumeration 1. 2. 3. and so forth, there are two more which are given in Table 2.2. In most cases these enumerations have some kind of headline, which is the last sentences before the enumeration. Enumerations also end with a colon. There is always a linebreak symbol between headline and enumeration.

For instance in Figure 2.1

Randabstände des Plattenepithelkarzinoms:

is a headline for

- Zum Bronchusresektionsrand: 0,4cm.

Whenever measurements or percentages occur there is a space between the value and the unit or the % sign. This does not have any linguistic relevance, but it is important for the preprocessing in the next Section.

On the sentence level multiple deviations from standard German occur. Most of the sentences are not grammatically complete. In most cases at least the verb is missing – if not more – and most sentences appear to be more bullet points than full sentences. Additionally, there are no sub-clauses within sentences. They just consist of the pure main sentence. These deviations are done by the pathologists in order to write a more efficient report, because it covers the same amount of information in less text this way.

2.2 Preprocessing

The goal of the preprocessing is to receive a list containing sentences. Figure 2.1 shows an example report text before and after applying the preprocessing. Afterwards, each sentence is represented as a list of tokens. For the purpose of a more intuitive understanding, this thesis sticks to the term *word* instead of token, regardless of whether it is an actual word or for instance a measurement value such as $4,5cm$. These measurements usually have a space between the number and the unit which is removed, initially. For instance *4,5 cm* including the space is reduced to $4,5cm$ without the space.
In addition to this property, most of the reports are semi-structured. There are different types of enumerations which need to be handled. Table 2.2 shows a list of the different enumeration types that were implemented.

Enumerations are classified in two groups identified using regular expressions. One group has a headline above the enumeration, the other group does not. A sentence is a headline of an enumeration if it ends with a colon followed by a linebreak symbol. If the enumeration has no headline, the enumeration symbols are just removed and each enumeration point is treated as a sentence. If there is a headline then this headline is added to each of the enumeration points. Together they are treated like a sentence from now on. Figure 2.1 illustrates this. When reading the resulting sentence it becomes clear that this does not differ from the sentence structure of common sentences within the histological reports. However, adding the headlines to the beginning of the enumeration points has one disadvantage.

Before preprocessing	After preprocessing
Unterlappen mit einem 6,5cm großen mäßiggrdig differenzierten Plattenepithelkarzinom	[["Unterlappen", "mit", "einem", "6,5cm", "großen", "mäßiggradig", "differenzierten" "Plattenepithelkarzinom"],
Minimale Randabstände des Plattenepithelkarzinoms:	
- Zum Bronchusresektionsrand: 0,4 cm	["Minimale", "Randabstände", "des", "Plattenepithelkarzinoms", "Zum", "Bronchusresektionsrand", "0,4cm"],
- Zu Pleura visceralis: 0,1 cm	["Minimale", "Randabstände", "des", "Plattenepithelkarzinoms", "Zur", "Pleura", "visceralis", "0,4cm"],
- Zum chirurgischen Resektionsrand: 0,7cm	["Minimale", "Randabstände", "des", "Plattenepithelkarzinoms", "Zum", "chirurgischen", "Resektionsrand", "0,4cm"],
Nebenbefundlich abszendierende Retentionspneumonie und fibrosierte Pleura visceralis	["Nebenbefundlich", "abszendierende" "Retentionspneumonie", "und", "fibrosierte" "Pleura", "visceralis"]]

TABLE 2.1: An example report before preprocessing is shown on the left side. The same report is shown after preprocessing on the right side. The after state is given in forms of Python lists.

Enumeration type	Example
Natural numbers	1. Zum Bronchusresektionsrand 2. Zur Pleura visceralis 3. Zum Chirurgischen Resektionsrand
Letters	A) Zum Bronchusresektionsrand B) Zur Pleura visceralis C) Zum Chirurgischen Resektionsrand
Dashes	− Zum Bronchusresektionsrand − Zur Pleura visceralis − Zum Chirurgischen Resektionsrand

TABLE 2.2: This Table shows a list of enumeration types that were implemented. Besides the classical enumeration using numbers also letters as well as dashes can be found in the histological report texts.

The letter case of the first word of the enumeration point remains the same as it is impossible to determine whether it needs to be in lower or upper case. Hence, the letter case has to be ignored in all further processing of the report including parsing the report as well as filtering the relations in the end.

After handling enumerations, the text is split into sentences at each period. When splitting a report into sentences, abbreviations like $v.A.$ are treated as a single word in order to not split a sentence into two in the middle of the abbreviation. Table 2.3 shows the list of abbreviations that were implemented to prevent the preprocessing from doing to. Finally, sentences are split at spaces resulting in the requested list of words.

Abbreviation	**Long version**
v.A.	vor Allem
usw.	und so weiter
etc.	et cetera
s.u.	siehe unten
vgl.	vergleiche
St.	Stadium
Stad.	Stadium
Gr.	Grad
Grd.	Grad
Nr.	Nummer
Pos.	Position

TABLE 2.3: This Table shows a list of abbreviations that were implemented in the preprocessing.

Chapter 3

Semantics

After the preprocessing was applied, it is critical to be able to identify medically relevant terms in the histological reports. This is used to filter the output of the Dependency Grammar parser later. The necessity for this results from two aspects. Firstly, the German Language contains synonyms – two different words, but the same semantics – such as *Colon* and *Kolon* which shall be treated as equal by the tool. Secondly, it is critical to be able to search for sets of words. For instance, it can be requested to extract the type of a carcinoma, which can be *Plattenepithelkarzinom*, *Pankreastumor* or *Nierenzellkarzinom* for example. Hence, the Unified Medical Language System (UMLS) is introduced in order to resolve these challenges. Its performance on German histological reports then is evaluated.

3.1 Properties of the UMLS database

UMLS is an ontology database system developed by the U.S. national library of medicine containing medical words and concepts. Originally, it just contained English words, but by the time German became the third most popular language in the database after English and Spanish. The German federal institute of Drugs and medical devices (BfArM) supports UMLS by filling it with German concepts. It also contains a wide variety of semantic information that are not made use of in this thesis. As UMLS sometimes contains words where umlauts were replaced by the international variant – such as *ae* instead of *ä* – it is necessary to query for two versions of the same word for instance *lymphozytär* and *lymphozytaer* if there is an umlaut in the word. Additionally, letter cases are ignored.

In opposition to the English version the German UMLS just contains lemmatised forms of word such as the infinitive for verbs and the nominative singular for nouns. Hence, words need to be lemmatised before querying UMLS. There are libraries that perform this task such as Germalemma [20] and German-Lemmatizer [14]. These lemmatisers were not trained for medical words. Hence, it is expected that medical words might cause problems. As a countermeasure, Germalemma was chosen, because it has a pattern-based approach. Usually, lemmata are looked up in the TIGER corpus [5] and if they are not found, the pattern-based approach takes over. Like most lemmatisers Germalemma requires the word class to be given. Nouns, verbs, adverbs and adjectives are supported. In order to derive these word classes there are three options. The first one is to use an additional Part-Of.-peech Tagger (POS-Tagger). Usually, POS-Taggers take the

sentence structure into account which we also do in the syntax part. Avoiding this leads to just one model to be trained on the same information for the tool instead of two. The second option is to hand over each word to Germalemma four times, each time with a different of the four word classes and querying UMLS for each of the lemmata returned by Germalemma. In total, this leads to up to eight times as many queries to UMLS. Each of the four lemmata might be converted to the international variant by replacing umlauts. The third option just works for Dependency Grammars. These also predict the type of a grammatical relation. These types can be mapped to Germalemma word classes. This is discussed further in Section 5.2. In the following, the evaluation of the two components UMLS and Germalemma is presented.

3.2 Evaluation of the semantic part

In order to find out whether this approach is suitable for information extraction from histological reports, evaluation is split in two parts. In the first part the performance of the lemmatiser Germalemma is evaluated in the context of the vocabulary of histological reports. Therefore, the proportion of correctly lemmatised medical words is computed. In the second part, UMLS is evaluated with a confusion matrix and recall, precision and f_1 score are computed from there. In mathematical terms, UMLS performs a classification task. To do so, it assigns one of two possible classes to each word. The one class is the class of medical words and the other one is the class of non-medical words. If the word can be found in UMLS, it is classified as medical, if it can not be found in UMLS it is classified as non-medical. Germalemma's and UMLS' performance is evaluated in the context of histological reports. Therefore, a corpus from clinical routine was annotated. It consists of a total of 249 reports, containing 3195 sentences and 1653 distinct words in total. The corpus was restricted to three types of reports. 205 breast biopsy reports, 26 liver biopsy reports and 18 bladder biopsy reports were chosen randomly. In order to make sure, no changes in the guidelines pathologists use to create their reports occurred, the reports were restricted to be written in the year 2020. The reports were created by two senior pathologists. Information regarding word classes, lemmata and the information whether a word is medical or not were provided in the annotation created by me.

3.2.1 Evaluation of the Lemmatisation

As Germalemma was evaluated sufficiently on standard-German words the medical words from the evaluation corpus serve as evaluation data exclusively. The words and their word classes are then processed by Germalemma and returned the lemmata prediction. Comparing prediction and target lemmata from the evaluation data it turned out that Germalemma lemmatised a proportion of 85.78% of the words correctly.

Although the performance is not too bad, it is not sufficient. If the lemmatisation returns a wrong lemma UMLS will definitely not be able to

find the correct word in the database. Hence, the incorrectly lemmatised words were analysed and additional rules were added before Germalemma is called. Table 3.1 gives an overview of the rules newly added to Germalemma for future usage.

word stem	**endings**	**substitution**
karzinom	(es | en | e | s)	karzinom
befund	(es | en | s | e)	befund
untersuchung	(en)	untersuchung
m[aä]ngel	(s)	mangel
gen	(es | en | s | e)	gen
infiltrat	(es | en | s | e)	infiltrat
zelle	(n)	zelle
tös	(en | e | r | s)	tös
om	(es | en | e | s)	om
or	(es | en | s)	or
igne	(r | s | n)	igne
herd	(es | en | e | s)	herd
zylinder	(s)	zylinder
zyste	(n)	zyste
typ	(en | s)	typ
gewebe	(es | en | s | n)	gewebe
nekrose	(n)	nekrose
tisch	(er | es | en | e)	tisch
tumor	(es | en | e | s)	tumor
ation	(en)	ation
parenchym	(es | e | s)	parenchym
struktur	(en)	struktur
veränderung	(en)	veränderung
stanze	(n)	stanze
blase	(n)	blase
ase	(n)	ase
segment	(es | en | s | e)	segment
drüse	(n)	drüse
tiv	(es | en | e | r)	tiv
gefäß	(es | en | e)	gefäß
lär	(er | es | en | e)	lär
orid	(er | es | en | e)	orid
nd	(er | es | en | e)	nd

TABLE 3.1: The Table shows patterns that were added to Germalemma after evaluating it on the medical words of the corpus described in Section 3.2.

All of the patterns follow the same idea. Prefixes before the pattern itself are not treated in any way. If the word stem followed by an ending is found – i.e. ending appended to stem as a regular expression is matched – stem and ending are replaced by the entry of the column *substitution*. It is critical to differentiate between stemming and lemmatisation here. It is not sufficient to just cut the ending of the word as this leads to wrong lemmata in a number of cases cases. For instance *Mängeln*, which is in accusative plural form, would be stemmed to *Mängel*, which is in nominative plural form, but its correct lemma is *Mangel* (nominative singular).

3.2.2 Evaluation of the UMLS database

After improving the lemmatisation such that all of the medical words of the corpus can be lemmatised correctly, UMLS was queried for each word of the whole corpus. If the word was found, the class medical was assigned to the word and non-medical otherwise. Figure 3.2 shows the resulting confusion matrix for this classification task.

↓ target / prediction →	**non-medical**	**medical**
non-medical	1303	34
medical	248	68

TABLE 3.2: The Table shows the confusion matrix for the evaluation of UMLS classifying into the classes medical and non-medical when querying for the full word. The full corpus from the Section 3.2 was used as evaluation data.

The evaluation was based on four metrics. For the first two metrics, the recall score indicates the proportion of words correctly classified as medical while the precision score indicated the proportion of words correctly classified as non-medical. Recall was calculated as 0.22, precision was calculated as 0.67. The f_1 score is defined as $f_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$, which is the harmonic mean of precision and recall score. Thirdly, the f_1 score was calculated as 0.33 which gives a good indicated of the classifiers overall performance. Fourthly, the accuracy – the proportion of correctly classified words – is 0.83. Unfortunately, the precision as well as the recall score are not sufficient for a data extraction task even though the accuracy is quite high. The large difference between f_1 score and accuracy is caused by the higher number of non-medical words in the corpus which UMLS classifies correctly in a higher portion than medical words.

In order to evaluate recall and precision score even further, search in UMLS was extended as follows: Instead of searching for the word as it is, UMLS was queried for entries that contain the input word as a substring. Table 3.3 shows the resulting confusion matrix of this kind of search.

While recall increased to 0.54 precision decreased to 0.3. In consequence,

↓ target / prediction →	non-medical	medical
non-medical	928	409
medical	144	172

TABLE 3.3: The Table shows the confusion matrix for the evaluation of UMLS classifying into the classes medical and non-medical when querying for the full word. The full dataset from the Section 3.2 is used as evaluation data.

the f1 score increased to 0.38 showing a better performance of UMLS. In opposition to this the accuracy decreased to 0.67 Unfortunately, this also is not a sufficient performance for an information extraction task. Alternative approaches to querying the German UMLS instance are discussed in Chapter 6.

Chapter 4

Syntax

In order to extract grammatical relations which are filtered later, parsers for natural languages with respective grammars were used. There are two types of grammars presented in this thesis namely Link Grammars and Dependency Grammars. Link Grammars are discussed first. Their formalism is explained together with the respective training and parsing algorithms. As there was no implementation in Python available, both the training- and the parsing algorithm were implemented during the work on this thesis. A proof of concept for Link Grammar parsing was executed successfully. However, Link Grammars have a number of disadvantages, which are discussed later. After presenting Link Grammars, Dependency Grammars are discussed. There already is an implementation for a training- and a parsing algorithms available. This framework is called Supar [11] and supports different Neural Nets which predict grammatical relations between the words in a sentence. Finally, the Neural Net created by Dozat and Manning [11] is evaluated on a corpus which consists of histological reports.

4.1 Link Grammar approach

Before using Neural Nets and Dependency Grammars, the first parser for relation extraction presented here is based on a Link Grammar. Link Grammars were developed by Davy Temperley and Daniel Sleator in 1995 for the English language [24]. Due to the linguistic differences between German dn English, it is necessary to adapt their work for German. This was already done by Sandra Kübler in 1998 [19]. Accordingly, the following Sections explain the general concept of Link Grammars, a Link Grammar parser as well as adaptions made to Temperley/Sleator and Küblers work. Sandra Kübler also developed an unsupervised training algorithm for German Link Grammars [19] which is also presented in this Chapter.

In order to model grammatical behaviour of a natural language one needs to investigate two aspects. The first aspect are relations between the words in a sentence depending on the structure of the sentence. The second aspect is conjunction or declination of words depending on their current grammatical context. In order to model these two aspects, Link Grammars contain a dictionary of words where conjunctions and declinations of the same word are treated as different words. Each word in there is mapped to a list of possible grammatical contexts that the word can appear in. A grammatical context is called a *disjunct*.

Each disjunct contains so-called *connectors* which model grammatical relationships between the word and its grammatical neighbours. Connecting two connectors of two words creates a *link* between the respective words. For instance let *großes Plattenepithelkarzinom* be a partial sentence. Then there is a grammatical relation between the two words, because *großes* is an adjective describing *Plattenepithelkarzinom* further. For instance, *großes* can have the connector *=ADJs* and *Plattenepithelkarzinom* can have the connector *§ADJs*. These two connectors can create a link between the words.

Connectors consists of three parts. The first part, the control sign, denotes whether the link is controlling which, denoted by §, or whether it needs to be controlled, denoted by =. This aspect was introduced by Kübler [19] in order to model the more flexible word ordering in German compared to English. Intuitively, this can be understood as the direction of the link. In the given example, *großes* describes *Plattenepithelkarzinom* further and not the other way around. The second part of a connector is its type, which is denoted as a capital letter. It models the main grammatical relation such as *adjective to noun* relation. In the provided example, the type of both connectors is ADJ. The third part of a connector models its subtype. In the above example, the subtype of both connectors is *s*, which denotes that two words in singular form are connected. Two connectors can form a link if their type and subtype are equal and their control sign is different.

Connectors are arranged to disjuncts in two ordered lists. Figure 4.1 shows a number of disjuncts as examples. The list on the left side of a disjunct contains all connectors where links need to be attached to the left hand side of the word, the right list of the disjunct contains all connectors where links need to be attached to the right side of the word. For instance, the disjunct of *Plattenepithelkarzinom* has two *§ADJs* connectors to its left side showing that an adjective, which is located on the left side of *Plattenepithelkarzinom* within the sentence, can be linked to it. For instance, *differenziertes* can use its $= ADJs$ connector from the right side of its disjuncts, in order to create a link between *Plattenepithelkarzinom* and *differenziertes*. Likewise, the $= ADV$ connector in the disjunct of *mäßiggradig* can connect to the *§ADV* connector on the right side of the disjunct of *differenziertes*. A connector which is linked to another connector is *satisfied*.

Connectors within disjuncts are ordered and must be used in this order. For instance, the sentence *das große Plattenepithelkarzinom* contains the words *Plattenepithelkarzinom* with the disjunct *((§DETs, §ADJs), ())*, the word *große* with the disjunct *((), (§ADJs))* and the word *das* with the disjunct *((), (=DETs))*. There must be the adjective *große* as well as the determiner *das* on the left side of *Plattenepithelkarzinom*, because the respective connectors occur in the left side of the disjunct. The determiner must occur before the adjective in the sentence, because the connector *§DETs* is given before the *§ADJs* connector in the disjunct of *Plattenepithelkarzinom*. If all connectors within a disjunct are satisfied, the disjunct is also *satisfied*.

As the two examples above already suggest, each word can have multiple disjuncts representing multiple grammatical occasions the word can occur in. Hence, the dictionary of a Link Grammar stores multiple disjuncts per

Word	Disjunct
Plattenepithelkarzinom	((§ADJs, §ADJs))
mäßiggradig	((), (=ADV))
differenziertes	((§ADV), (=ADJs))
großes	((), (=ADJs))

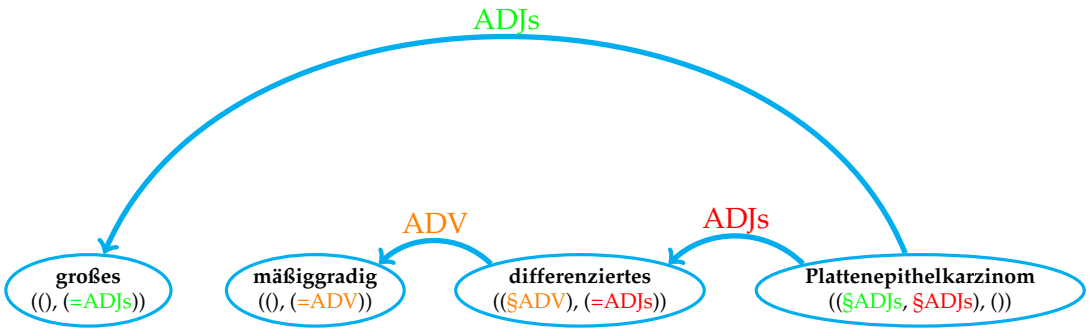


TABLE 4.1: The Figure shows the disjuncts required to parse the example sentence *großes mäßiggradig differenziertes Plattenepithelkarzinom*. Below the Table, which contains the words and their disjuncts, the resulting links between the words are given in forms of arrows.

word. If it is possible to choose one disjunct per word in a sentence such that each of the connectors can be used to create a link to another word, the sentence can be parsed based upon the grammar. In order to meet certain linguistic properties, the following meta-rules as defined by Temperley and Sleator must be met [24]. When using the words within a sentence as nodes and links as edges, the first three meta-rules are well-known properties of graphs:

- Planarity: Links are drawn above the sentence and do not cross
- Connectivity: The links suffice to connect all the words of the sentence together
- Exclusion: No two links connect the same pair of words
- Ordering: The order of the Connectors forming the links must not be changed from the order within the respective disjuncts.

The meta-rules particularly have to be met for the link between the verb of a sentence and the period at the end of the sentence. Unfortunately, histological reports usually do not have verbs in their sentences as explained in Chapter 2. Hence, periods cannot be connected to verbs and the respective type of a connector is simply not used anywhere. Hence, this thesis removes the final period from the end of a sentence and ignores it in the parsing and training process in opposition to Kübler.

The parsing process itself is nothing else than checking the satisfiability of the grammar denoted in propositional logic: The dictionary of the language and its disjuncts can be transformed to propositional logic as Sleator and Temperley did. Hence, a parser is just a satisfiability-checker making use of the meta-rules. The fact that Link Grammars need a dictionary containing each word makes them so-called lexicalized grammars. These have one disadvantage. Each measurement value such as $3,4cm$ or $6,5cm$ is treated as a separate word. In order to resolve this, each measurement is replaced by the token $< measurement >$ during the training and parsing preprocessing. If measurements were not replaced, $3,4cm$ and $6,5cm$ would be treated separately although both have the exact same grammatical features. Hence, they need the same disjuncts. In order to treat all measurements equally while parsing but still be able to extract the respective value from the specific report, the token is replaced by the original value after the parsing succeeded.

4.1.1 Parsing of Link Grammars

Although a parser for Link Grammars was already implemented in the programming language *C* the changes required for German were not implemented there. During the work for this theses the parser was re-implemented in the programming language Python while adaptations for the German language were made. The parser computes the number of possible parsings as well as the linkage graphs. This can be done quite efficient, because Link Grammars are context-free as proven by their inventors [24]. Hence, a polynomial-time parser exists [28]. Sleator and Temperley described such an algorithm making use of memoization to reduce asymptotic runtime complexity from exponentially to polynomial. This parser might come up with several linkage graphs possible for a sentence. This is not necessarily wrong: For instance in the sentence *Er sprach mit ihm über seine Vorlieben* it is ambiguous whether *seine* is related to *er* or *ihm*. Nevertheless, exactly one parsing should be chosen by the parser in our use case. As there is no way to choose a parsing supported by linguistics, the parsing coming up first returned by the parser. The resulting linkage graph corresponds to the relation graph, relations are taken from and filtered later.

4.2 Training of a Link Grammar

As there was no German grammar for a Link Grammar parser available, it was critical to train such a grammar during the work on this thesis. S. Kübler described a training algorithm [19] which already covers the differences in the Link Grammar formalism as presented in Section 4.1. Kübler's algorithm belongs to the class of unsupervised machine learning and hence does not require target values to be provided. The input for the algorithm contains two types of information. The first one is an initial grammar fragment, which will be extended iteratively. It corresponds the disjuncts for the example sentence in Figure 4.1 exemplarily. The second one is a list of sentences along with the word classes of the words, which forms the unsupervised training data. Based on the available information in the initial

grammar fragment, the algorithm generates all possible linkages. Then, it uses a metric in order to find the best linkage. This linkage and its disjuncts then are used to extend the grammar. This process then is repeated for each sentence in the training data.

In order for the training algorithm to work it is critical to know the word class of each word in the training data. Hence, data for the unsupervised learning need to be annotated such that the word class is given. This can either be done by hand or by using a POS-Tagger or by annotating the reports by hand. If a POS-Tagger is used, the tools performance will also be influenced by another machine learning model. POS-Taggers usually make use of th sentence structure. Hence, we would end up with our Link Grammar being influenced by another model which also relays on sentence structure. In order to eliminate this, training data were annotated by hand.

It is important to note that these unsupervised training data are not used in order to optimize the model, but just to extend the initial grammar fragment, which contains completely correct information. Optimization such as in training neural nets always causes the risk of overfitting and memorizing training examples. As there are no training examples given in the unsupervised training data, it is impossible for the grammar to memorize any. The same applies for the problem of overfitting. As there is no optimization executed, there is no chance of overfitting. Hence, the Link Grammar training algorithm explained in the following is immune against memorization and overfitting. This is an advantage over neural nets. Errors made by the parser are solely caused by wrong disjuncts being learned during the training. This will be discussed further in the end of Section 4.2.1 after the training algorithm was presented.

4.2.1 The training algorithm for Link Grammars

As already stated, there was no modern Python implementation of Küblers training algorithm available. Hence, the algorithm was implemented during the work on this thesis. Its basic idea is to generate as many linkages as possible as long as they fulfil the meta-rules. These are then evaluated using a metric called *membership value*. Eventually, the best linkage with the highest membership value is chosen and the grammar is extended accordingly.

Initially, the training algorithm loads the initial grammar fragment. This functionality can also be used in order to expand an already existing grammar even more, without the need to restart the training. After loading has finished, the sentences within the unsupervised training data are processed consecutively. The following steps are repeated for each sentence while expanding the grammar fragment. In step one the word classes of the words within the sentence are loaded. Step two of the original algorithm is ignored in our case due to the lack of verbs.

In step three, the algorithm loads all possible disjuncts known for the words in the sentence. It then connects the connectors in any possible way. This

way, multiple as complete linkages as possible are created. This is illustrated in Figure 4.1. There, the blue link is generated by connecting the connectors of the words *einem* and *Plattenepithelkarzinom*. The respective connectors used there were printed in bold characters.

In step four, the algorithm aims to completely satisfy all disjuncts which currently have at least one satisfied connector. This is done by linking the words with such a disjunct to other words in any possible way. It does not matter here, if the other word already has a satisfied connector. This step is illustrated in Figure 4.2. The blue link there was generated in the previous step. The yellow link is added in step four. There are two possible partial linkages generated in this step. The boldly printed connector of *Plattenepithelkarzinom* can be attached to *differenzierten* and to *mäßiggradig*. Both linkages are stored and forwarded to the next step.

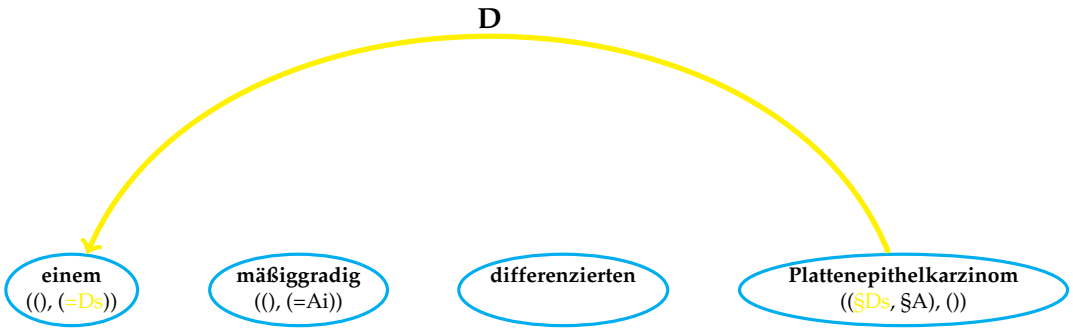


FIGURE 4.1: The Figure shows the training of a Link Grammar after the third step. Below the words of the sentence, the current disjuncts of the words are printed. Links between words are marked as arrows. The yellow link was added in step three.

Step five eventually links all remaining connectors to neighbouring words which are not linked until now, in any possible way. This step is illustrated in Figure 4.3. The example processes the upper linkage from Figure 4.2. The blue links there were generated in the previous steps. The yellow link is added in step five. There are two possible partial linkages generated. The boldly printed connector of *mäßiggradig* can be attached to *differenzierten* and to *Plattenepithelkarzinom*. Both linkages are stored and forwarded to the next step.

If there is a word left which does not have any link attached to it, disjuncts from words with the same word class are loaded from the grammar. Connectors from these disjuncts then are used to connect the words to neighbouring words in any possible way. The same is done if the linkage does not fulfil the connectivity meta-rule and there are words left where disjunct loading was not possible initially. Finally, the algorithm comes up with a set of possible linkages. Neither of the above steps is allowed to break the meta-rules. In consequence, all resulting linkages fulfil all meta-rules. Nevertheless, some words have an issue at the moment: In consequence of the steps four and five there might be links attached to them, although their disjunct does not cover a respective connector. This is then resolved by extending the disjuncts accordingly as a last step before the search for the best linkage is going on.

During this training, the grammar G is represented using fuzzy relations. Each pair of a word w and a disjunct d which was discovered has a rational value from zero to one assigned to it. This value is called *membership value*. The membership value represents, how sure the algorithm is that d is a correct disjunct for w. All pairs – denoted as (w, d) in the following – in the initial grammar fragment have a value of one, because they are definitely correct. By default, every other pair (w, d) has a membership value of zero. We can check, whether a (w, d) combination is already known to

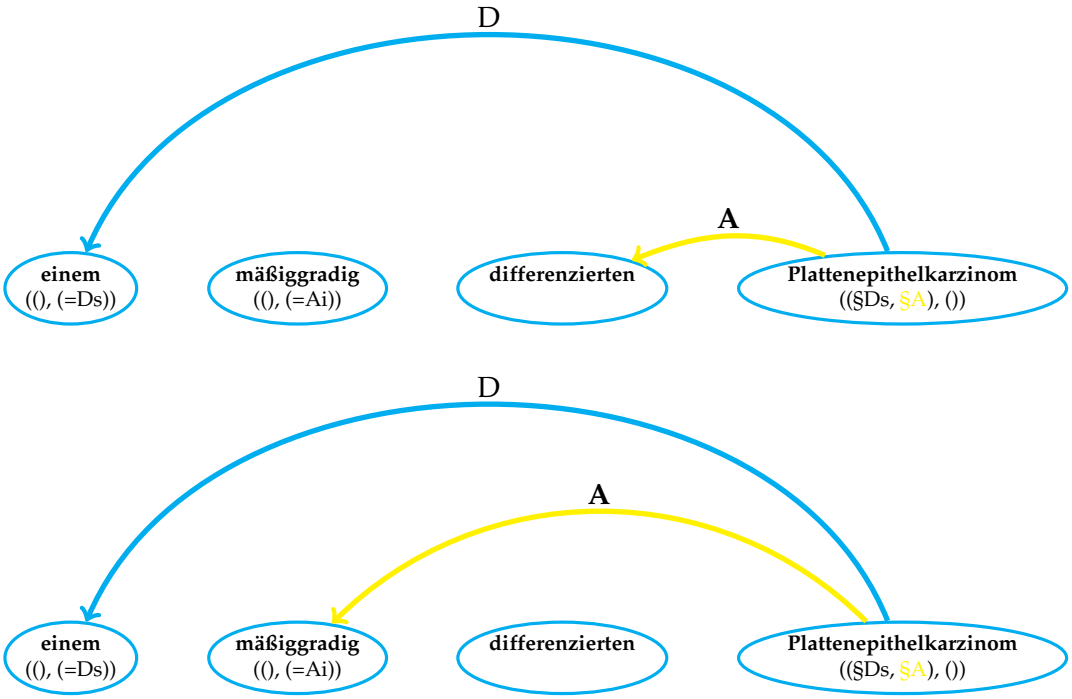


FIGURE 4.2: The Figure shows the training of a link grammar after the fourth step. Below the words of the sentence, the current disjuncts of the words are printed. Links between words are marked as arrows. The yellow links were added in step four.

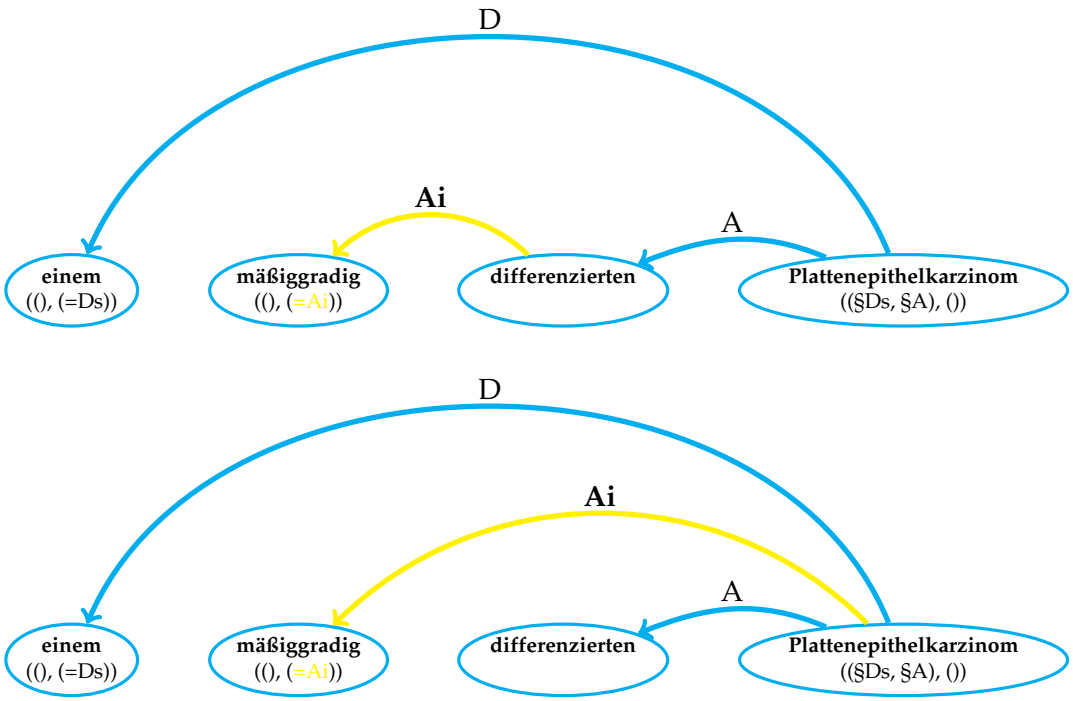


FIGURE 4.3: The Figure shows the training of a link grammar after the fifth step. Below the words of the sentence, the current disjuncts of the words are printed. Links between words are marked as arrows. The yellow links were added in step five.

the grammar denoted by $(w, d) \in G$, whether a disjunct is already known to the grammar denoted by $d \in G$ and whether a word is already known to the grammar $w \in G$. $membership\_value_G(w, d)$ denotes looking up the membership value of (w, d) from G. If a new (w, d) is discovered in the training, algorithm 1 is used to compute its membership value.

Algorithm 1 $membership\_value(w, d, G)$

if $(w, d) \in G$ **then**
  return $membership\_value_G(w, d)$
else if $w \in G$ **then**
  Search in G for other disjunct d' with maximal $membership\_value_G(w, d')$ and minimal $distance(d, d')$, prioritize membership value over distance if necessary
  return $membership\_value_G(w, d') - 0.1 - distance(d, d')$
else if $d \in G$ **then**
  Search in G for other word w' with maximal $membership\_value_G(w', d)$
  if wordclasses of w and w' are equal **then**
    return $membership\_value_G(w', d) - 0.1$
  else
    return $\frac{membership\_value_G(w', d)}{2}$
  end if
else
  return 0
end if

The distance function is needed in order to determine how large the difference between two disjuncts is. Let c, c' be connectors and d, d' disjuncts. $features(c)$ is the set of features of the connector. $control(c) \in \{, =\}$ is the control sign of the connector. The distance function then is defined as follows:

$$distance(d, d') := \sum_{c \in d, c' \in d'} \begin{cases} 0 & \text{if } c = c' \\ 0.05 & \text{if } features(c) = features(c') \\ 0.1 & \text{if } control(c) = \text{'§'} \\ 0.2 & \text{if } control(c) = \text{' ='} \end{cases}$$

The membership values of all (w, d) pairs in a sentence then are averaged. This results in the membership value of the whole linkage. The linkage with the highest membership value is chosen as the best one. After that, the membership values of the (w, d) pairs are stored in the grammar and replace the previous values. After the whole training data have been used, the grammar is converted from the fuzzy representation to the normal representation, which maps the words to their possible disjuncts. A disjunct d is included in the grammar G for a word w if the membership value $membership\_value_G(w', d)$ is larger than 0.7. This threshold is the only hyperparameter the link grammar model provides. It has been chosen the same as Kübler did. In the following, Section 4.2.2 argues why this threshold has not been used in order to optimize the models performance.

However, the training algorithm has one disadvantage. As the linkages are generated by connecting the words in any way provided by the grammar and the meta-rules, it can generate wrong links. These wrong links then lead to wrong disjuncts being learned. The wrong linkage then is available to be used for training on further sentences. There, the wrong connector can lead to even more wrong links. This way, the wrong connector spreads through the grammar. This effect is reduced by the membership

value function by reducing the membership value by 0.1 whenever a new disjunct is discovered for a word. The idea behind this is that if a new disjunct is discovered for a word, the new disjunct has to proof itself by being derived from the grammar multiple times. If the disjunct d occured a sufficient number of times for a word w – depending on the context it appeared in due to the distance function – the membership value of the (w, d) pair will exceed the threshold to be included in the final grammar. Due to this, the threshold for the membership value is critical: If it is set too low, many wrong disjuncts might appear in the grammar. If it is set too high, correct disjuncts might be excluded from the grammar, because they did not occur enough times.

4.2.2 Proof-of-concept

As a proof of concept for the implementation fo the training- and the parsing algorithm, 20 randomly selected sentences from the liver biopsies were annotated with respective disjuncts. These serve as the initial grammar fragment. 100 sentences from the breast biopsy reports were annotated with word classes to serve as unsupervised training data. The training algorithm successfully learned disjuncts for those 100 sentences. Afterwards, the parser successfully parsed them into linkage graphs by using the Grammar trained before.

At this point, the model just parsed sentences it already had seen in the training. When extending the proof of concept to unknown sentences one big problem arose. The parser was not able to parse sentences which contain unknown words. In particular, it is impossible to extract any information from such a sentence by using a Link Grammar. This applies for unknown words as well as for typos which may be included in the report text. As this is a quite big drawback to the Link Grammar model, it is not evaluated further. Additionally, parts of the Link Grammar formalism are not chosen very well. In particular, it is allowed to have cycles in linkage graphs which is not supported by linguistics. This can lower its performance, because it enables the parser to create parsings which are linguistically impossible.

4.3 Dependency grammar based approach

In order to resolve the two disadvantages Link Grammars have, Dependency Grammars can be used. The First problem of cycles in the relation graphs is resolved, because the output of Dependency Grammar parsers can be restricted to being a tree [11]. Secondly, Dependency Grammar parsers base can handle unknown words, because they are based upon neural nets. These use word embeddings, which support unknown words and typos. For working with Dependency Grammars, useful tools such as the data annotation tool Arborator [13] are provided by the Universal Dependencies project [9]. Additionally, a Dependency Grammar parser called Supar exists, which also supports training of respective neural nets that can be used for Dependency Grammar parsing.

4.3.1 Parsing with a Dependency Grammar by using Supar

Supar supports multiple neural nets designed for the purpose of Dependency Grammar parsing. For this thesis, the model of Dozat and Manning [11] was chosen as it showed the best performance in the literature. There is also a model by Zhang et. al. [30] which showed slightly lower performance.

In order to represent Dependency Grammar parsing trees, the Universal Dependencies project introduced the CoNLL-U format [6]. The CoNLL-U format defines how a string must to be formatted in order to represent a Dependency Grammar parsing tree. It can easily be converted into nested python dictionaries. Figure 4.2 shows an example sentence, which is represented as a tree, in the CoNLL-U format and as a Table.

Each line represents a word with the following information separated by tab symbols: At first there is the index of the word. The root node in the parsing tree is implicitly given at index zero. Hence, the first word in the sentence has the index one. The following four information are not used in this thesis and hence are crossed out as $-$. After these, the arcs connecting the nodes in the parsing tree are given. For each word the index of the father node is stated. For the word connected to the root node the index zero is set. Succeeding the arcs, the types of the grammatical relations between the words are given. The word connected to the root node always has the relation type tag *root*. There is always exactly one word in a parsing tree attached to this implicit root node. In the relation types, *nummod* stands for a numerical modifier, *amod* for an adjectival modifier and *advmod* for an adverbial modifier.

When a sentence as given in the example is presented to Supar, it uses the given neural net in order to predict the father node of each word as well as the relation type between son and father node. Besides these two information, the CoNLL-U format supports the representation of more linguistic information than required for this thesis. For instance, it is also possible to represent word classes and lemmata in the CoNLL-U format.

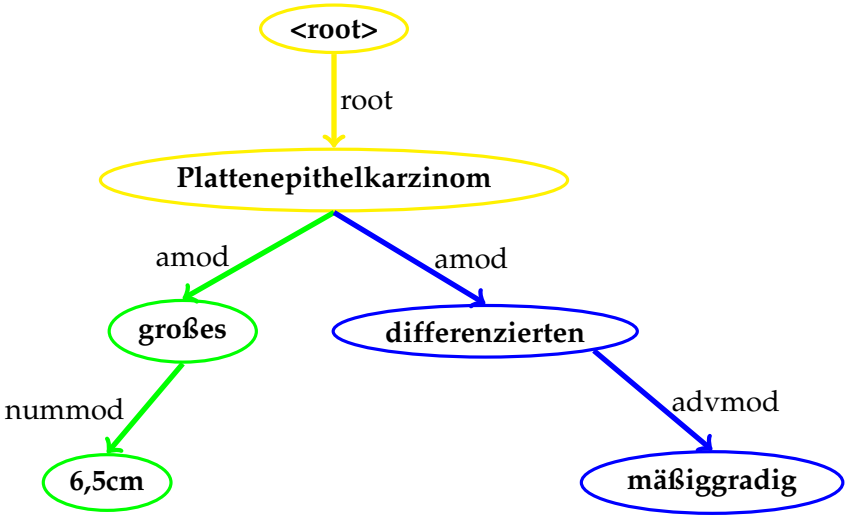
4.3.2 Training of a Dependency Grammar by using Supar

In order to train the neural net developed by Dozat and Manning, a large set of training data annotated in the CoNLL-U format is necessary. As it is out of scope for a master thesis to annotate such a large data set, two existing data sets were merged. The German GSD data set was annotated in the Universal Dependencies project [21]. This dataset is combined with the tweeDe data set [16]. The data were shuffled and a train-dev-test split with 70% training, 20% development and 10% test data was created.

For comparison purposes, this thesis also evaluates the performance of Dozat's and Manning's model pretrained by them. They used the data annotated in the Universal Dependencies project from multiple languages. In particular, Bulgarian, Catalan, Czech, German, English, Spanish, French, Italian, Dutch, Norwegian, Romanian and Russian.

CoNLL-U:

1	6,5cm	-	-	-	-	2	nummod	-	-
2	großes	-	-	-	-	5	amod	-	-
3	mäßiggradig	-	-	-	-	4	advmod	-	-
4	differenzierts	-	-	-	-	5	amod	-	-
5	Plattenepithelkrzinom	-	-	-	-	0	root	-	-



Index	Word	Arc	Relation type
1	6,5cm	2	nummod
2	großes	5	amod
3	mäßiggradig	4	advmod
4	differenzierts	5	amod
5	Plattenepithelkarzinom	0	root

TABLE 4.2: The Figure shows the sentence *6,5 cm großes mäßiggradig differenziertes Plattenepithelkarzinom* parsed into the CoNLL-U format at the top, as a relation tree in the middle. Additionally, the informations required for he purpose of this thesis are given in the table at the bottom. In the tree, relation types are denoted in the rectangle next to each node. The column *Arc* in the table corresponds to the index of the word which is the father node in the tree.

The training- and hyperparameters for training on the GSD and tweeDe data were chosen the same as Dozat and Manning did. However they did not give all of the hyperparameters, which hence were set to the default values provided by Supar. Table 4.3 shows the training- and hyperparameters chosen for training Dozat's and Manning's model on the GSD and tweeDe data. As Supar relies on Pytorch for training neural nets, the parameter names are given in the way they are named in Pytorch. As Supar supports early stopping the number of epochs is not executed completely. The parameters *weight_decay*, *decay* and *decay_steps* belong tu Pytorch's ExponentialLR library. Supar uses Adam as the optimizer.

As Supar expects word embeddings to be provided for training, the tool

parameter	**value**
learningrate	$1 \cdot 10^{-3}$
update_steps	1
batch_size	1000
numEpochs	500
mu	0.9
nu	0.9
eps	$1 \cdot 10^{-8}$
weight_decay	0
decay	0.75
decay_steps	5000
buckets	64
min_freq	3

TABLE 4.3: The Table shows the hyper- and training parameters used o train Dozat's and Manning's model on the GSD/tweeDe data. The parameter names are identical to their python implementation in Pytorch.

Fasttext [4] was used to train word embeddings on the GSD/tweeDe data set. Table 4.4 shows the hyperparameters chosen for Fasttext.

parameter	**explanation**	**value**
model	skip-gram (skipgram) or continuous bag of words (cbow)	skipgram
minCount	count a word has to appear in the corpus to be added to the word embeddings dictionary	3
dim	length of the word embeddings vectors	300

TABLE 4.4: The Table shows the hyperparameters for the Fasttext word embeddings. A skip-gram model is trained. A word must occur in the corpus at least three times in order to be included in the dictionary. The length of the word embeddings vector for each word is 300.

Besides the decision for a skip-gram instead of a CBOW model, there are just two hyperparameters. The dimension of the word embeddings vector was set to 300. In order to appear in the dictionary, a word has to appear in the training data at least three times. The hyperparameters were set to Fasttext default values.

4.3.3 Evaluation

The performance of Dozat's and Manning's model – trained by them as well as on the GSD/tweeDe data – is evaluated by following an idea published by Gomez-Rodriguez et al. (2019) [15]. Besides the three metrics Unlabelled Attachment Score, Labelled Accuracy and Labelled Attachment Score, which are the gold standard for evaluating the performance of a Dependency Grammar parser, Gomez-Rodriguez et. al. recommended to also perform an application-specific evaluation. In this thesis, this is the task of relation extraction in the first and information extraction in the second step. In this first step, the proportion of correctly generated relations was evaluated. In the second step, data were written to a Table where they can be used from for different applications in medical research. It was then evaluated, how many entries in the Table were filled correctly.

In order to evaluate the performance of the Dependency Grammar parser based on the three standard metrics, evaluation data were annotated in the CoNLL-U format by annotating 200 sentences randomly selected from the 205 breast biopsy reports that were already used in Section 3.2.

The three standard metrics are based on the components of a Dependency Grammar parsing tree. Firstly, the parsing tree consists of arcs between the words. Accordingly, the Unlabelled Attachment Score (UAS) is defined as the proportion of arcs correctly predicted by the parser. Secondly, the trees consist of the relation type tags assigned to a node in the tree. The Labelled Accuracy (LA) is defined as the proportion of correctly predicted tags, accordingly. Combining UAS and LA, the Labelled Attachment Score (LAS) is defined as the third metric. The LAS is defined as the proportion of words where the parser assigned the correct tags as well as the correct arc to. By definition, LAS must be smaller or equal to LA and UAS.

metric	**Pretrained**	**Trained on GSD/tweeDe**
Unlabelled Attachment Score (UAS)	0.94	0.83
Labelled Accuracy (LA)	0.92	0.80
Labelled Attachment Score (LAS)	0.9	0.74

TABLE 4.5: The Table shows the evaluation results by using the three standard metrics for measuring Dependency Grammar parsing performance. As evaluation data, 2005 sentences from breast biopsy reports were randomly selected and annotated. The values were rounded to the second decimal.

Table 4.5 shows the UAS, LA and LAS score for the model pretrained by

Dozat and Manning [11] as well as for the model trained for this thesis. Although both models were trained using the same training- and hyperparameters, the performance of the model trained on the GSD and tweeDe data is significantly worse than the performance of the pretrained model. It is worse by more than 10% in all three scores. This must not be caused by the data resulting in a worse performing model. The hyper- and/or training parameters maybe were not the same. As the paper of Dozat and Manning [11] does not show all training parameters required by Supar, a number of parameters just were concluded from information in the paper or simply left as Supar's default values. The other factor that can explain the difference in performance is that the model trained on the GSD/tweeDe data set has different word embeddings. Due to these two aspects, the two models are not comparable and more investigation is required.

Regardless of the worse performance of the GSD/tweeDe model, it is interesting to compare the scores for the pretrained model with each other. UAS is the best of the three with 0.94 followed by LA with 0.92 and LAS with 0.9. As LA is smaller than UAS, the model is worse in tagging words with correct relation types than in attaching arcs to the words. When investigating parser output and the evaluation data, it becomes clear why this happens. There is one class of sentences and one class of relation types causing problems. The three relation types $fixed$, $flat$ and *compound* are so-called Multi Word Expressions. Their property is that they connect words such as *Carcinoma lobulare*. The term *Carcinoma lobulare* then is one term describing one concept. The whole term *Carcinoma lobulare* just makes sense, because both words are standing together. Quite often, the parser failed to tag the word *lobulare* correctly, but it succeeds in attaching *lobulare* to *Carcinoma* in the parsing tree. The same occurs for different MWEs. Due to this, MWEs reduce performance with respect to LA and LAS but not to UAS.

Besides the poor performance on tagging MWEs, the parser also failed on a whole class of sentences. The first sentence of every report contains information regarding the localization of the biopsy in the human body. For breast biopsies, for instance, it can be *links oben außen*. In terms of linguistics, this is not a proper sentence. There is unambiguous way to add relation types to the words here. A common grammatically complete German sentence must contain a verb. The *root* relation type is assigned to this verb. If there is no verb (as in most of the histological reports) a noun can take over the role of the root element in the parsing tree without causing errors in the rest of the parsing. This is possible, because nouns stand higher in grammatical hierarchy than most other words classes. Either way, there is no word in the given sentence which stands higher in the grammatical hierarchy than all of the other words. There are just words – here: $links$, *oben* and *außen* – on the same hierarchy level. If there was a noun – such as *Mamma* – all three words would be connected to this noun. As there is none, the parser needs to attach these words to each other. This results in every word, but the one tagged with the *root* relation type, having wrong arcs as well as wrong relation type tags. The evaluation data were annotated in a consistent way in order to compensate this ambiguity, but the parser was not able to meet these parsings. In order to make sure this linguistic defectiveness

does not harm the models performance, Table 4.6 shows the same evaluation as before while cleaning the localisation sentences from the corpus. After cleaning, 165 sentences remained.

Metric	**Pretrained no localisations**	**GSD/tweeDe no localisations**
Unlabelled Attachment Score (UAS)	0.96	0.84
Labelled Accuracy (LA)	0.95	0.83
Labelled Attachment Score (LAS)	0.93	0.78

TABLE 4.6: The Table shows the evaluation results by using the three standard metrics for measuring Dependency Grammar parsing performance. As evaluation data, the 165 sentences which do not contain localisations such as *linksobenauen* were extracted from the 200 sentence breast biopsy corpus. In all of the three metrics, the model pretrained by Dozat and Manning [11] – column *Pretrained* – is superior compared to the model trained on the German GSD and tweeDe data – column *GSD/tweeDe* – although both were trained using the same training- and hyperparameters. The values were rounded to the second decimal.

Again, the pretrained model is superior over the one trained on the GSD/tweeDe data. For the pretrained model UAS increased by 0.02 to 0.96, LA increased by 0.03 to 0.95 and LAS increased by 0.03 to 0.93. From the observation that LA increased more than UAS the conclusion arises that localisation sentences cause more problems in tagging relation types than they cause in attaching arcs to the words when compared to the evaluation data. However the evaluation data are unable to give correct parsings here as already mentioned above. Now UAS and LA are quite similar. The remaining difference can be explained with the MWE tagging problem which was discussed above.

Unfortunately, there currently is no different model available to compare to. The work by Kara et. al. [17] is based on nephrological reports. These differ from histological report. For instance, it is more common to use abbreviations in clinical reports than in histological reports. Hence, comparison requires both models to be evaluated and trained on the same data sets.

Chapter 5

Information Extraction

After semantic – as presented in Chapter 3 – and syntactical informations – as presented in Chapter 4 – regarding the histological records are available, it is critical to combine them in order to be able to extract information from the histological reports. In order to do so, two steps are performed. In the first step, grammatical relations of different arities are generated based on the output of the Dependency Grammar parser. In the second step, these relations are filtered by using UMLS and regular expressions. The requested information then is returned.

5.1 Relation Extraction

Identification of medical words using UMLS as well as constructing grammatical relations from a preprocessed sentence using a Dependency Grammar parser is possible now. Further it is necessary to combine both in order to extract relations. After the preprocessing has taken place, the following is repeated for each sentence. The relation graph is constructed by parsing the sentence using the Dependency Grammar parser trained by Dozat and Manning [11] as discussed in Chapter 4. The resulting 2-ary relations then are used in order to generate longer relations. This is done by attaching a 2-ary relation to an other relation if they share exactly one entry. This way, 2-ary relations are extended to 3-ary, which then can be extracted to 4-ary and so on. The maximum arity has to be defined beforehand and is derived from the longest required relation. This is carried out further in Section 5.2

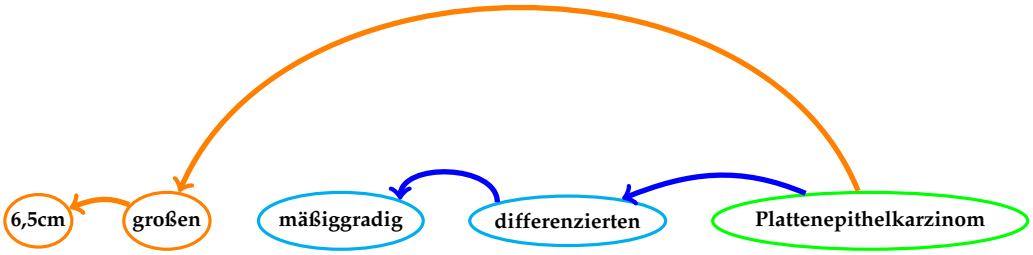


FIGURE 5.1: This Figure shows the traversal of a relation graph. The green word (Plattenepithelkarzinom) is used to attach it to green, which is attached to 6, 5cm among the orage path. The same applies to Plattenepithelkarzinom, differenzierten and mäßiggradig in the blue path. Each of the two paths then forms a 3-ary relation.

Figure 5.1 illustrates the relation attachment process on an example sentence graphically. The relations between *Plattenepithelkarzinom* and *großen* as well as between *großen* and $6,5cm$ are attached to each other. They form the relation
(Plattenepithelkarzinom, großen, 6.5cm). The same is done for relations including the words *mäßiggradig* and *differenzierten*, which eventually come up with the 3-ary relation *(Plattenepithelkarzinom, differenzierten, mäßiggradig)*.

5.2 Relation filtering

Although relations are a well-defined construct, this form of representation has a number of disadvantages. It is not a format medical researchers use to work with, usually. Hence, it has to be transformed into a more useful format. In most cases, it is still necessary to map parts of relations to fields within an appropriate data model. For valuation and presentation purposes, a Table form is used here. In order to extract the data into the Table, it is necessary for the tool to be able to map relations and their content to column names in the Table. In order for such mappings to become possible it is necessary to represent the relations in a more standardised manner. Currently, words within a relation can occur in an arbitrary word form. This problem can be resolved by lemmatising each word – for instance by using Germalemma – as described in Chapter 3. Besides the word itself, Germalemma requires the input of the word class of the word. In order to retrieve the word classes, the relation types from the Dependency Grammar parser output were mapped to Germalemma word class tags as given in Table 5.1. Germalemma supports the word classes noun N, adjective ADJ, adverb ADV and verb V. The Dependency Grammar relation types were defined in the Universal Dependencies project [9]. The mapping was retrieved by looking at the definition of the Dependency Grammar relation type and choosing an appropriate word class. For instance, an adjectival modifier, which has the Dependency Grammar relation type *amod*, must be an adjective, which has the Germalemma shortcut ADJ.

After the relations were generated and the words were lemmatised, the extraction of relations follows the workflow given in Figure 5.2. In an additional script, a set of tuples has to be defined. The elements of each tuple can be regular expressions or UMLS concept IDs. For instance, in python such a tuple can look like

[0-9]+(\.[0-9+])?[cdm]m, groß | Größe, <C00007137>).

Those tuples then are compared to the relations extracted by the process in the previous Section. To do so, each regular expression from the tuple has to be fulfilled by an element of the relation. For instance, if the relation is *(Plattenepithelkarzinom, groß, 6.5cm)*, the first regular expression in the tuple mentioned before matches the last element of the relation. The second element of the relation fulfils the second entry of the tuple. *C00007137* is the UMLS concept ID of *Plattenepithelkarzinom*, which can be figured out by querying UMLS for *Plattenepithelkarzinom* and returning the concept ID. Finally, $6,5cm$ is returned as the extracted information. In this process, the

order of the elements in the relation does not matter. For instance, it is not important whether *gro* or *Plattenepithelkarzinom* comes first in the relation. This is done to make sure the word order in the sentence does not affect the filter results. For instance, it is not relevant whether the sentence is *ein 6,5cm großes Plattenepithelkarzinom* or *das Plattenepithelkarzinom ist 6,5cm groß* any more. Both contain the same information.

Then, the whole process is repeated for a predefined set of tuples which define the data set to be extracted. The length of the longest tuple determines the longest arity of the relations generated in Section 5.1.

Dependency Grammar relation type	**Germalemma word class**
root	N
nsubj	N
iobj	N
obl	N
vocative	N
expl	N
dislocated	N
nmod	N
appos	N
nummod	N
csubj	V
advcl	ADV
acl	ADJ
advmod	ADV
amod	ADJ
aux	V
cop	N
clf	N

TABLE 5.1: The Table shows the mapping from Dependency Grammar relation types to Germalemma word classes. Relation types not given in the Table do not have word classes supported by Germalemma.

5.3 Information extraction evaluation

The relations extracted before and after they were filtered can be used to perform an application-specific evaluation as recommended by Rodriguez et. al. [15].
In order to evaluate the performance of the approach before relations are filtered, all 2-, 3- and 4-ary relations were extracted from the breast biopsies reports. Table 5.2 shows the proportions of relations that were extracted correctly from the respective Dependency Grammar parser output.

For information extraction in a medical environment – such as histological reports – relations containing at least one medical word are relevant. Hence, the relations included in the evaluation were restricted to the ones

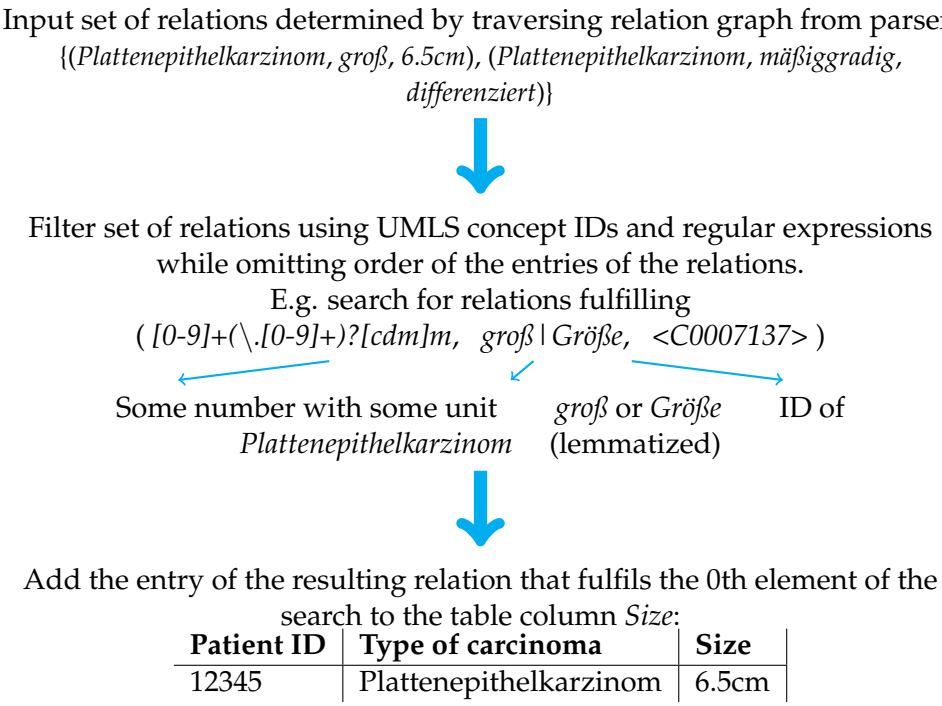


FIGURE 5.2: The Figure shows the information extraction process from a set of relations. A tuple of regular expressions and UMLS concept IDs is defined which then is used to filter the relations extracted by the Dependency Grammar parser. Finally, the requested information is extracted from the relation found and is written to the table.

Proportion of n-ary relations	Pretrained all	Pretrained no localisations	GSD/tweeDe all	GSD/tweeDe no localisations
$n = 2$	0.95	0.97	0.86	0.87
$n = 3$	0.91	0.93	0.73	0.74
$n = 4$	0.88	0.89	0.62	0.63

TABLE 5.2: The Table shows the proportions of relations correctly extracted by the Dependency Grammar parser. Each relation contains at least one medical word. Relations with an arity from two to four were included in the evaluation. Two models were evaluated. The left model is the one pretrained by Dozat and Manning [11] the other right is the model trained during this thesis on the German GSD and tweeDe data. The values were rounded to the second decimal.

containing at least one medical word. The evaluation was performed on the pretrained model as well as on the model trained on the GSD and tweeDe data. It was repeated for the corpus including the localisation sentences as well as for the 165 sentences without the localisation sentences. Again, the pretrained model was superior over the model trained on the GSD and tweeDe data. For both corpora, the one including and the one not including the localisation sentences, the performance of the pretrained model correlates negatively with the arity of the relations that shall be extracted. This is not surprising as the extraction of relations with an arity of more than two solely depends on the set of 2-ary relations. This set of 2-ary relations is directly taken from the parser output. Hence, it corresponds to the set of arcs in the parsing tree if all arcs connecting two non-medical words are removed. In conclusion, the proportion of correctly extracted 2-ary relations corresponds to the UAC score when ignoring arcs connecting two non-medical words. Due to the observation that the proportion of these 2-ary relations is higher than the respective UAC score from Chapter 4, it can be concluded that the presence of medical words in a sentence does not negatively affect the performance of the Dependency Grammar parser. Even though Multi Word Expressions seem to limit the performance of the Dependency Grammar parser as discussed in Chapter 4 this observation cannot be extended to medical word in general. According to the evaluation results, the parser causes more errors in creating the arcs between two non-medical word than it does if there is at least one medical word. Hence, it can be concluded that Dependency Grammar parsers are suitable to parse histological reports even if they were not trained on any medical corpus.

After evaluation took place on unfiltered relations, it is important to inspect at the performance of the whole approach in information extraction. For this purpose, a dataset of ten relevant parameters was defined by the clinic of surgery from the University Hospital Aachen. This dataset was created in order to research a certain kind of liver carcinoma, the so-called hepatocellular carcinoma (HCC). Hence, a corpus of ten reports was taken to extract information from them. The results are shown in Table 5.3. There are three different kinds of variable types namely boolean, measurement –

# HCCs	Fibrosis	Vascular invasion	Tumor diameter	Infammation	Inflammation degree	Distance to reSection area	Desmet stage	Steatosis	Cirrhosis
1			1,4cm			1mm			TRUE
1		TRUE	5,5cm			0,3cm			FALSE
1	FALSE		4,2cm	FALSE		0,3cm			FALSE
1	TRUE	TRUE	8,5cm	TRUE	—		3		
1			16cm			0,1cm			
1	TRUE	TRUE	4,2cm	TRUE		1,5mm		TRUE	FALSE
1									
1		FALSE	9,5cm			1cm			
1		FALSE	8,5cm	TRUE				TRUE	
1	TRUE		3,6cm			0,2cm	1-2		

TABLE 5.3: The Table shows a dataset extracted by the tool developed in this thesis. Each row represents one report. The column names are information defined to be relevant for researching HCC by the clinic of surgery within the University Hospital Aachen. Information printed in black were extracted correctly. The data can have the two data types boolean, integer or measurement value with respective unit. The Information in the column *Inflammation degree* was extracted wrong, which is denoted by the red dash.

for instance $1,4cm$ – and integers. For the purpose of evaluating the approach presented here, the semantics of the data is irrelevant.

However, almost all of the requested information were extracted successfully. One entry in the column *Inflammation degree* is missing. No information is given at the entry — although it is given in the report. This error was caused by the sentence

(...) mit milder entzündlicher Aktivität und portaler sowie septenbildender Fibrose mit Architekturstörung (Grad 2, Stadium 3 nach Desmet).

In order to extract the information, it is necessary for the 4-ary relation *(entzündlich, Aktivität, Grad, 2)* to be generated by the parser. Unfortunately, this was impossible, because the parser did not find the relation between *Aktivität* and *Grad*. This is not supported by lingustics. From a grammatical point of view, there is no relation between the words. The only way to Figure out this relation is to make use of the semantics of the words. To do so, the Ontology database should contain the information that inflammation has a degree and this degree ranges from one to four. Then, the tool could guess that *entzündliche Aktivität* and *Grad 2* belong together. However, this is not unambiguous. There could be another medical word in the sentence which can also be related ot a degree value. If there is just one degree – for instance because the degree of either inflammation or the other value was not measured by the pathologist – it is completely impossible to determine which medical word *Grad 2* belongs to.

Chapter 6

Outlook

The work presented in this thesis demonstrates two major points. Firstly, Link Grammars have a number of disadvantages when used to parse German medical texts. Their linguistic motivation is unsatisfying because Link Grammars allow grammatical relations between word which are not supported by linguistics for instance when cycles in the linkage/relation graph occur as discussed in Section 4.2.2. Additionally, Link Grammars are lexicalized, which makes them unable to handle unknown words and typos. If one would still want to use Link Grammars it is possible to train word embeddings and use these as the input for the Link Grammar parser. This resolved the problem of the Link Grammar not being able to parse sentences containing unknown words or typos. The bad linguistic motivation however cannot be resolved that easy.

The other important result this thesis shows is that Dependency Grammars and in particular the Supar tool offer a great alternative which shows good performance in parsing histological reports. The full approach including the lemmatisation as well as the relation filtering and information extraction shows satisfyingly well results. However, a number of issues still have to be resolved before the tool developed in this project can be used in practice.

The first challenge is that there are aspects of the lemmatisation that can be improved. For non-medical words Germalemma yields satisfactory results. Medical words however are often derived from Latin or ancient Greek and hence are lemmatised less correctly by Germalemma. As both of these languages are dead, they do not change any more. Hence, it is possible to get the final set of Latin and ancient Greek word endings and add them to Germalemma as patterns. As Germalemma also looks up lemmata in the TIGER corpus it is possible to extend the respective dictionary of words, for instance by adding all words and lemmata from the German GSD and tweeDe data and even more corpora annotated in the Universal Dependencies project. As a dictionary lookup has a constant and fast asymptotic runtime runtime, this does not even slow down the tool.

A second way to improve lemmatisation is to use a different approach in obtaining the word classes of the words in a sentence. Mapping the relation types created by the parser to Germalemma word classes yields wrong results in a number of cases. In most cases, this mapping is sufficient, for instance when the relation type *amod* (adjectival modifier) is mapped to ADJ (adjective). But a number of relation types can be used by several

word classes. One example is the *root* relation type. In a parsing tree generated by a Dependency Grammar parser this is always assigned to the root of the tree regardless of its word class. For instance, in the sentence *mäßiggradig differenziertes Plattenepithelkarzinom* it is assigned to the word *Plattenepithelkarzinom*, which is a noun. In the sentence *Tumor wurde nicht ausreichend erfasst*, the *root* relation type is assigned to the word *erfasst*, which is a verb. Currently, the tool developed in this thesis would treat both of the words *Plattenepithelkarzinom* and *erfasst* as nouns. This leads to a wrong lemma for *erfasst* being returned by Germalemma. Hence, it is necessary to use a tool performing part of speech tagging by tagging each word with its respective word class. The Supar tool also supports this if the model is trained accordingly. Due to the restricted time for this thesis however it was not possible to evaluate it properly on histological reports.

Even though Dozat's and Manning's Dependency Grammar parser is not trained on a corpus of medical texts, it shows excellent performance in parsing such texts. However, this can be further optimized. It seems as the parser is often unable to parse Multi Word Expressions correctly. This might be caused by the fact that these mostly consist of medical word coming from ancient Greek or Latin. Creating a large corpus of histological reports annotated in the CoNLL-U format and training the parser on this corpus might resolve this problem. It should consist of reports written by more than just two pathologists. One of the limitations of this thesis is that the evaluation data set is quite small. It also does not contain reports from non-native speakers. Hence, it remains unclear whether the parsing performance might decrease if the report is written by a non-native speaker.

Besides linguistic considerations there is also one improvement regarding the preprocessing possible. Currently, the preprocessing removes each type of punctuation symbol such as semicolon, colon or comma. This is due to the fact that it was initially unclear whether and to what extent the Link Grammar would be able to support punctuation within sentences. For Dependency Grammar however this is not a problem. A relation type which attaches punctuation symbols to words in the sentence exists. Additionally, the *parataxis* relation type is defined, which enables parsing of two sentences which are connected by a colon instead of splitting them. Hence, punctuation symbols can be part of sentences as long as they are treated as separate words and are not merged with any word. Regardless of the improvements mentioned above, the syntactical part of the tool supports an excellent performance of the tool.

In contrast to this, the semantic part of the tool requires more attention. The performance of UMLS is too poor to be useful. As shown in Chapter 1.1, the approach of using an Ontology database is quite common and leads to sufficient results if the database is large enough. Unfortunately, this is not the case in the context of German. But the English UMLS instance consists of approximately 50 times as many medical concepts and hence is much more useful. In order to be able to use this, an idea is to translate words into English before querying UMLS. For instance, this can be done by using the machine translation tool DeepL[10]. However, DeepL was not trained on medical texts. Hence, it is necessary to evaluate its translations before

utilizing it in the context of information extraction.

As soon as the performance of the German UMLS instance will be improved, further options can be explored to improve its usefulness in the context of information extraction. Currently, it only allows for the filtering of relations by using the UMLS concept entry itself. Depending on the application case, it may be useful to be able to filter relations for a whole class of concepts such as all carcinoma types. UMLS contains a hierarchy of concepts which can theoretically be used for this.

However, before UMLS will become more helpful it is critical to validate the existing tool further. Hence, a larger evaluation data set is going to be annotated for further evaluation. The tool will be optimized and evaluated on a larger data set of the HCC cohort used in Section 5.3 and further data from different clinical data sets. This way, high performance on arbitrary clinical data sets is ensured. Eventually, the tool will be evaluated on corpora from other clinical texts than histological reports which enables it to provide arbitrary data from arbitrary clinical texts. So far, the tool cannot handle distinct language patterns such as hyphenation. In order to resolve this problem, the tool is going to be developed further to support these patterns.

After the tool is finished completely, it will be hard to decide how to progress further. On the one hand, this AI-based approach offers pathologists a simple way to provide their findings for medical research. On the other hand, AI-based approaches are always going to make mistakes which can affect research results negatively. Hence, synoptical reporting is on the rise in Europe [18]. Currently, it is impossible to determine whether synoptic reporting or AI-based approaches will be used more frequently. However, an approach combining the advantages of both approaches is possible. It contains three steps. Firstly, the verbal dictations of the pathologists are converted to written text in real-time by using an AI-based text-to-speech system. Secondly, the generated text is converted into a structured table form as described in this thesis. Thirdly, this table is presented to the pathologist after finishing the dictation. The pathologist ten can make corrections to the data: Based on the performance of the tool, this should not lead to too many mistakes. Eventually, this approach combines the advantage of AI-based systems that data can be structured very comfortably with the advantage of synoptic reporting that a high quality of the data is assured.

Bibliography

[1] M. Becker, S. Kasper, B. Böckmann, K.-H. Jöckel, and I. Virchow. Natural language processing of German clinical colorectal cancer notes for guideline-based treatment evaluation. en. *International Journal of Medical Informatics*, 127:141–146, July 2019. ISSN: 13865056. DOI: 10.1016/j.ijmedinf.2019.04.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S1386505619301145 (visited on 05/01/2021).

[2] S. Bloehdorn, P Cimiano, A. Hotho, and S. Staab. An Ontology-based Framework for Text Mining. *Forum American Bar Association*, 20, 2004.

[3] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. en. *Nucleic Acids Research*, 32(90001):267D–270, Jan. 2004. ISSN: 1362-4962. DOI: 10.1093/nar/gkh061. URL: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkh061 (visited on 05/01/2021).

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *arXiv:1607.04606 [cs]*, June 2017. URL: http://arxiv.org/abs/1607.04606 (visited on 11/15/2021). arXiv: 1607.04606.

[5] S. Brants, S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. en. *Research on Language and Computation*, 2(4):597–620, Dec. 2004. ISSN: 1570-7075, 1572-8706. DOI: 10.1007/s11168-004-7431-3. URL: http://link.springer.com/10.1007/s11168-004-7431-3 (visited on 08/29/2021).

[6] CoNLL-U. URL: https://github.com/EmilStenstrom/conllu.

[7] H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. en. *PLoS Computational Biology*, 9(2):e1002854, Feb. 2013. A. Prlic, editor. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002854. URL: https://dx.plos.org/10.1371/journal.pcbi.1002854 (visited on 12/29/2021).

[8] H. Dalianis. *Clinical Text Mining*. en. Springer International Publishing, Cham, 2018. ISBN: 978-3-319-78502-8 978-3-319-78503-5. DOI: 10.1007/978-3-319-78503-5. URL: http://link.springer.com/10.1007/978-3-319-78503-5 (visited on 05/01/2021).

[9] M.-C. de Marneffe, C. D. Manning, J. Nivre, and D. Zeman. Universal Dependencies. en. *Computational Linguistics*:1–54, May 2021. ISSN: 0891-2017, 1530-9312. DOI: 10.1162/coli_a_00402. URL: https://direct.mit.edu/coli/article/doi/10.1162/coli_a_00402/98516/Universal-Dependencies (visited on 11/14/2021). https://universaldependencies.o

[10] DeepL. URL: https://www.deepl.com/translator.

[11] T. Dozat and C. D. Manning. Deep Biaffine Attention for Neural Dependency Parsing. *arXiv:1611.01734 [cs]*, Mar. 2017. URL: http://arxiv.org/abs/1611.01734 (visited on 10/28/2021).

[12] A. G. Fritz. *International classification of diseases for oncology: ICD-O*. English. World Health Organization, Geneva, 2013. ISBN: 978-92-4-069212-1. URL: http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=1681147 (visited on 05/02/2021).

[13] K. Gerdes. Collaborative Dependency Annotation. *Proceedings of the Second International Conference on Dependency Linguistics*:88–97, 2013.

[14] German-lemmatizer. URL: https://github.com/jfilter/german-lemmatizer.

[15] C. Gómez-Rodríguez, I. Alonso-Alonso, and D. Vilares. How important is syntactic parsing accuracy? An empirical evaluation on rule-based sentiment analysis. en. *Artificial Intelligence Review*, 52(3):2081–2097, Oct. 2019. ISSN: 0269-2821, 1573-7462. DOI: 10.1007/s10462-017-9584-0. URL: http://link.springer.com/10.1007/s10462-017-9584-0 (visited on 10/28/2021).

[16] Ines Rehbein and Josef Ruppenhofer and Bich-Ngoc Do. tweeDe – A Universal Dependencies treebank for German tweets. URL: https://aclanthology.org/W19-7811.pdf.

[17] E. Kara, T. Zeen, A. Gabryszak, K. Budde, D. Schmidt, and R. Roller. A Domain-adapted Dependency Parser for German Clinical Text. en. In *A Domain-adapted Dependency Parser for German Clinical Text*. Verlag der Österreichischen Akademie der Wissenschaften, 2018. ISBN: 978-3-7001-8437-9. DOI: 10.1553/0x003a12bd. URL: https://hw.oeaw.ac.at/8437-9 (visited on 01/24/2022). Pages: 0xc1aa5576_0x003a12bd.

[18] R. Z. Karim, K. S. Van Den Berg, M. H. Colman, S. W. McCarthy, J. F. Thompson, and R. A. Scolyer. The advantage of using a synoptic pathology report format for cutaneous melanoma: Synoptic pathology reports in melanoma. en. *Histopathology*, 52(2):130–138, Dec. 2007. ISSN: 03090167. DOI: 10.1111/j.1365-2559.2007.02921.x. URL: http://doi.wiley.com/10.1111/j.1365-2559.2007.02921.x (visited on 05/02/2021).

[19] S. Kübler. Learning a Lexicalized Grammar for German, 2002.

[20] M. Konrad. Germalemma. URL: https://github.com/WZBSocialScienceCenter/germalemma.

[21] R. McDonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. Hall, S. Petrov, H. Zhang, O. Täckström, and others. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, 2013.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[23] U. K. Schneider. *Sekundärnutzung klinischer Daten - rechtliche Rahmenbedingungen*, number Band 12 in Schriftenreihe der TMF-Technologie- und Methodenplattform für die Vernetzte Medizinische Forschung e.V. MWV Medizinisch WissenschaftlicheVerlagsgesellschaft, Berlin, 2015. ISBN: 978-3-95466-142-8.

[24] D. Sleator and D. Temperley. Parsing English with a Link Grammar. *CoRR*, abs/cmp-lg/9508004, 1995.

[25] I. Spasic, S. Ananiadou, J. McNaught, and A. Kumar. Text mining and ontologies in biomedicine: Making sense of raw text. en. *Briefings in Bioinformatics*, 6(3):239–251, Jan. 2005. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/6.3.239. URL: https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/6.3.239 (visited on 05/01/2021).

[26] M. Q. Stearns, C. Price, K. A. Spackman, and A. Y. Wang. SNOMED clinical terms: overview of the development process and project status. eng. *Proceedings. AMIA Symposium*:662–666, 2001. ISSN: 1531-605X.

[27] Supar. URL: https://github.com/yzhangcs/parser.

[28] L. G. Valiant. General context-free recognition in less than cubic time. en. *Journal of Computer and System Sciences*, 10(2):308–315, Apr. 1975. ISSN: 00220000. DOI: 10.1016/S0022-0000(75)80046-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0022000075800468 (visited on 08/27/2021).

[29] R. Weegar and H. Dalianis. Creating a rule based system for text mining of Norwegian breast cancer pathology reports. en. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 73–78, Lisbon, Portugal. Association for Computational Linguistics, 2015. DOI: 10.18653/v1/W15-2609. URL: http://aclweb.org/anthology/W15-2609 (visited on 05/01/2021).

[30] Y. Zhang, Z. Li, and M. Zhang. Efficient Second-Order TreeCRF for Neural Dependency Parsing. en. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.302. URL: https://www.aclweb.org/anthology/2020.acl-main.302 (visited on 10/28/2021).

[31] X. Zhou, H. Han, I. Chankai, A. Prestrud, and A. Brooks. Approaches to text mining for clinical medical records. en. In *Proceedings of the 2006 ACM symposium on Applied computing - SAC '06*, page 235, Dijon, France. ACM Press, 2006. ISBN: 978-1-59593-108-5. DOI: 10.1145/1141277.1141330. URL: http://portal.acm.org/citation.cfm?doid=1141277.1141330 (visited on 05/01/2021).