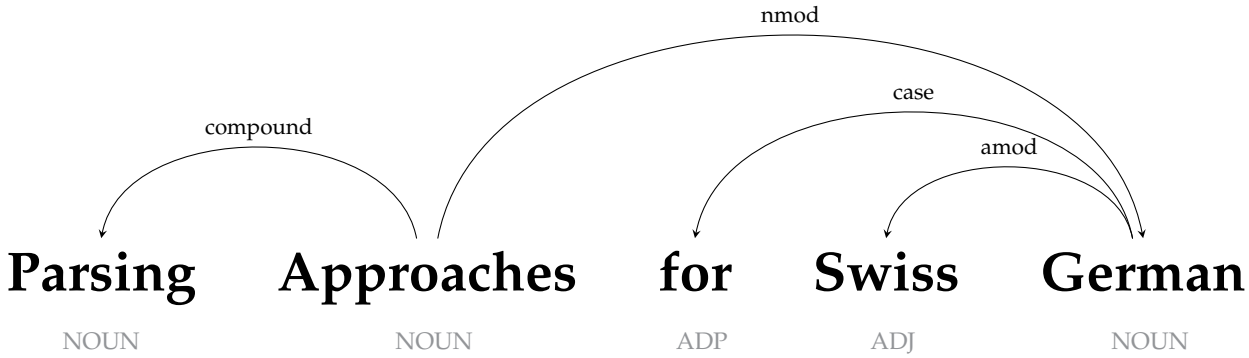
University of Zurich[UZH]

Master's thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zurich

for the degree of
Master of Arts UZH

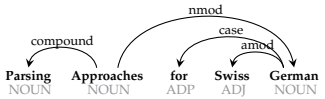


Author: Noëmi Aepli
Student ID Nr.: 10-719-250

Examiner: Prof. Dr. Martin Volk
Supervisor: Dr. Simon Clematide

Institute of Computational Linguistics

Submission date: 10.01.2018



Abstract

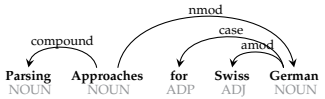
This thesis presents work on universal dependency parsing for Swiss German. Natural language parsing describes the process of syntactic analysis in Natural Language Processing (NLP) and is a key area as many applications rely on its output. Building a statistical parser requires expensive resources which are only available for a few dozen languages. Hence, for the majority of the world's languages, other ways have to be found to circumvent the low-resource problem. Triggered by such scenarios, research on different approaches to cross-lingual learning is going on. These methods seem promising for closely related languages and hence especially for dialects and varieties.

Swiss German is a dialect continuum of the Alemannic dialect group. It comprises numerous varieties used in the German-speaking part of Switzerland. Although mainly oral varieties (*Mundarten*), they are frequently used in written communication. On the basis of their high acceptance in the Swiss culture and with the introduction of digital communication, Swiss German has undergone a spread over all kinds of communication forms and social media. Considering the lack of standard spelling rules, this leads to a huge linguistic variability because people write the way they speak. Differences within the dialect continuum do not only apply to the pronunciation but also to all linguistic aspects, including syntactic variations.

Such a situation is a challenging task for NLP. In the case of Swiss German dialects, the closely related, resource-rich language Standard German is available. The *Universal Dependencies* (*UD*) project for example provides a German treebank consisting of 15,590 dependency parsed sentences.

In this thesis, different cross-lingual parsing strategies are applied to Swiss German, exploiting the Standard German resources. The methods applied are the lexicalised annotation projection approach and the delexicalised model transfer approach, as well as direct cross-lingual transfer as a comparison setting. While for model transfer the German *UD* treebank can be applied, the annotation projection approach requires parallel sentences. In order to get parallel data, Standard German translations for Swiss German sentences are crowdsourced, resulting in a parallel corpus of 26,015 sentences, containing several translations for each of the 6,197 Swiss German sentences taken from the *NOAH* corpus and two books.

The results show around 60% Labelled Attachment Score (LAS) for all approaches and provide a first step towards Swiss German dependency parsing. The resources are available for further research on NLP applications for Swiss German dialects.



Zusammenfassung

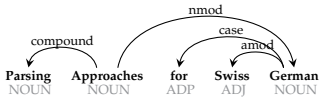
Diese Arbeit behandelt Dependenzparsing für Schweizerdeutsch. Parsing für natürliche Sprachen beschreibt den Prozess der syntaktischen Analyse in der Verarbeitung natürlicher Sprache (Natural Language Processing, NLP). Parsing ist ein Schlüsselbereich, da viele Anwendungen auf dessen Analyse beruhen. Der Aufbau eines Parsers erfordert teure Ressourcen, welche nur für einige Dutzend Sprachen zur Verfügung stehen. Daher müssen für die Mehrheit der Sprachen andere Wege gefunden werden, um das Problem der Ressoursenknappheit umzugehen. Ausgelöst durch solche Situationen werden verschiedene Ansätze zu sprachübergreifendem Lernen erforscht. Diese Methoden erscheinen für eng verwandte Sprachen vielversprechend und eignen sich daher besonders für Dialekte und Varietäten.

Schweizerdeutsch ist ein Dialektkontinuum der Alemannischen Dialektgruppe und besteht aus zahlreichen in der Deutschschweiz gesprochenen Varietäten. Obwohl es sich hauptsächlich um mündliche Sprachen (*Mundarten*) handelt, werden sie häufig in schriftlicher Kommunikation verwendet. Auf der Basis ihrer hohen Akzeptanz in der Schweizer Kultur und mit der Einführung der digitalen Kommunikation hat sich Schweizerdeutsch auf alle Arten von Kommunikationsformen und Social Media ausgebreitet. In Anbetracht der fehlenden Rechtschreibregeln führt dies zu einer grossen sprachlichen Variabilität, da die Leute schreiben, wie sie sprechen. Unterschiede innerhalb des Dialektkontinuums betreffen nicht nur die Aussprache, sondern alle sprachlichen Aspekte, einschliesslich syntaktischer Variationen.

Eine solche Situation ist eine schwierige Ausgangslage für die Verarbeitung natürlicher Sprache. Für die Schweizer Dialekte ist jedoch die eng verwandte, ressourcenreiche Sprache Standarddeutsch verfügbar. Das *Universal Dependencies* (*UD*) Projekt stellt eine deutsche Baumbank zur Verfügung, die aus 15'590 dependenzgeparsten Sätzen besteht.

In dieser Arbeit werden verschiedene sprachübergreifende Parsing-Strategien für Schweizerdeutsch angewendet, welche die standarddeutschen Ressourcen nutzen. Die angewandten Methoden sind der lexikalische Annotationsprojektionsansatz und der delexikalisierte Modelltransferansatz sowie als Vergleich der direkte sprachübergreifende Transferansatz. Während für die Modellübertragung die deutsche *UD*-Baumbank angewendet werden kann, erfordert der Annotationsprojektionsansatz parallele Sätze. Um parallele Daten zu erhalten, werden deutsche Standardübersetzungen für schweizerdeutsche Sätze crowdsourced, was in einem parallelen Korpus von 26'015 Sätzen resultiert. Dieses enthält jeweils mehrere Übersetzungen für 6'197 schweizerdeutschen Sätze, die aus dem *NOAH*-Korpus und zwei Büchern stammen.

Die Ergebnisse zeigen rund 60% Labeled Attachment Score (LAS) für alle Ansätze und stellen einen ersten Schritt in Richtung schweizerdeutsches Dependenzparsing dar. Die Ressourcen stehen zur weiteren Erforschung von NLP-Anwendungen für schweizerdeutsche Dialekte zur Verfügung.



Acknowledgements

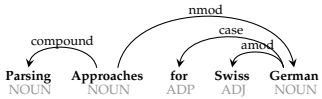
I would like to thank my supervisor Simon Clematide for the ideas and motivating support during the work on this thesis. Also, I very much appreciated Simon's thoroughly prepared lectures, which I followed with great interest. Furthermore, I thank Martin Volk for the enormous support and encouragement during my studies, and the whole Institute of Computational Linguistics at the University of Zurich for being an incredibly motivating and inspiring environment.

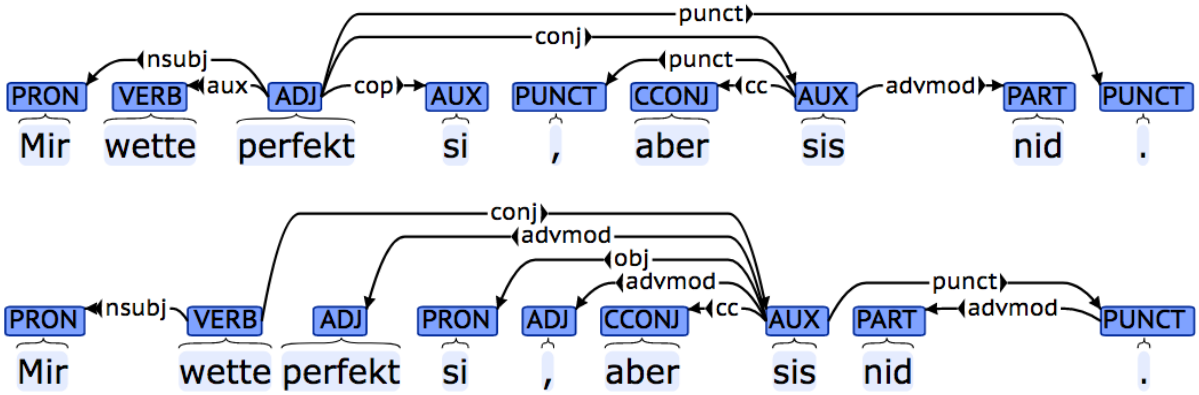
Moreover, many thanks to Pedro Lenz and Renato Kaiser for the permission to use their books for this project.

Thanks to the Linguistic Citizen Scientists for translating Swiss German sentences into Standard German and to the project managers of the *AGORA* project[1] who allowed me to work with the data.

Many thanks also to Anne Göhring for discussing some tricky gold standard sentences with me, as well as to Emilie Boillat, André Aepli and Martin Sigrist for proofreading this work.

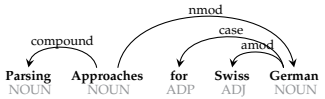
[1] http://www.snf.ch/de/foerderung/wissenschaftskommunikation/agora/Seiten/default.aspx



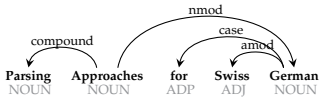


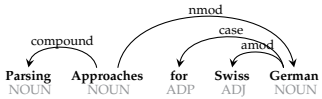
We want to be perfect, but we are not.

Top: gold standard, bottom: system. "LAS": not quite perfect.



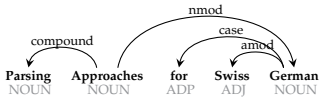
Contents





List of Abbreviations

ASR	Automatic Speech Recognition
CoNLL	Conference on Computational Natural Language Learning
DCA	Direct Correspondence Assumption
DE	German
GSW	Swiss German
IPA	International Phonetic Alphabet
LAS	Labelled Attachment Score
MT	Machine Translation
MST	Maximum Spanning Tree
NLP	Natural Language Processing
POS	Part of Speech
SMT	Statistical Machine Translation
STTS	Stuttgart-Tübingen-TagSet
UAS	Unlabelled Attachment Score
UD	Universal Dependencies
UPOS	Universal Part of Speech



1. Introduction

Dealing with non-standardised languages is a demanding task, which becomes even more challenging in the absence of standard orthographic rules, as is the case for Swiss German.

Swiss German dialects, as opposed to the Swiss Standard German recognised as one of the four official languages of Switzerland, feature a huge variety. Unlike other dialect situations, the Swiss German dialects are deeply rooted in the Swiss culture and enjoy a high reputation, i.e. dialect speakers are not considered less educated as is the case in other countries. Despite being oral languages, the dialects are used increasingly in written contexts where writers spell as they please.

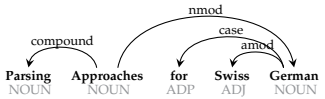
A situation as described above implies a lack of resources and tools in the field of NLP, which is the case for the majority of languages. Compiling such resources from scratch is a laborious and expensive process. Thus, in such cases, cross-lingual approaches offer a perspective to get started with automatic processing of the respective language. Cross-lingual approaches are especially promising if a closely related resource-rich language is available, which is the case for Swiss German. However, not every low-resourced language is in such an advantageous position, which is a motivation for increasing research in cross-lingual NLP methods. The renowned *Universal Dependencies* (*UD*) project aims at developing and setting a standard for cross-linguistically consistent treebanks (i.e. annotated text corpora) in order to facilitate multilingual parsing research.

The information about which word of the sentence is dependent on which other one is important in order to correctly understand the meaning of a sentence. Thus, it is needed for numerous NLP applications like information extraction or grammar checking. The task of identifying these dependencies is done by a dependency parser, which comes in different types depending on the method they apply.

For the purposes of this thesis, I engaged in two different cross-lingual dependency parsing strategies, namely annotation projection as lexicalised approach, and model transfer as delexicalised approach. The goal is to identify strengths and weaknesses for every approach and to discover if they are prone to dialectal differences. In order to do so, I created a gold standard consisting of different dialects, two of which featuring major differences in word ordering. Furthermore, I parsed a *silver standard treebank* which, compared to manually annotating from scratch, accelerates the process of building a treebank representing the training set for a monolingual Swiss German parser.

1.1. Outline

Section 2 provides more details on the motivation behind this research paper well as background information. It describes the Swiss German dialect situation including the differences to Standard German. Furthermore, it explains natural language parsing and crowdsourcing for NLP. Section 3 discusses work conducted in the fields of NLP for Swiss German and especially cross-lingual parsing, also describing the *Universal Dependencies* project. Regarding practical aspects, Section 4 specifies the data which was used in this thesis, most importantly manually annotated gold standard. The cross-lingual statistical dependency parsing methods applied are characterised in Section 5 along with short presentations of the tools used. Finally, Sections 6 and 7 present the results and summarise the findings of this thesis.



2. Background and Motivation

In this section, I present Swiss German along with its differences to Standard German and its peculiarities. Furthermore, this chapter explains the concept of Natural Language Parsing, its different types along with their applications. In addition, this section treats the problem of data acquisition with one possible solution, crowdsourcing, that I applied in order to get part of my data, i.e. the Standard German translations for Swiss German sentences in order to generate a parallel corpus.

2.1. Swiss German Dialects

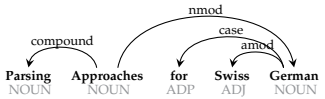
Schwiizerdütsch [ʃʊit͡sərdyt͡ʃ] is not to be confused with Swiss Standard German, which is the variety of Standard German used in Switzerland. In Switzerland, we use the term *Schriftsprache* (literary language) to refer to Standard German, which suggests the restricted usage of Standard German in Switzerland to written contexts. De jure, Swiss German is not the official language in Switzerland, de facto however, it is the main means of (official) oral communication in the German-speaking part of the country. Swiss German is a special case in many aspects which also go beyond linguistics. Dialects have a special status in Switzerland; not only do we strictly separate them from Standard German but also use them as a form of identification with our home village. The dialects enjoy high acceptance, and using them does not convey any social or educational inferiority as it is the case with many non-standard varieties. Discussing this situation with an Italian native speaker shows an example of the opposite case, which is more common. He would speak his dialect (from Rome) only with his friends from school but not with his parents (speaking a dialect from Naples instead); needless to mention he would never use his dialect in any formal situation, as he would be considered uneducated. Even in Germany, German-dialect speakers are considered less intelligent and uneducated (Stukenberg, 2015). This is definitely not the case in Switzerland.

There are a lot of linguistic aspects which make Swiss German special and this is one of the reasons why dialectal research has been an active area and has gained more attention recently. In Switzerland, dialects are not only used in spoken language and for private purposes. People also write and even publish in their own dialect and use it in formal situations. In contrast, Standard German is used in some specific situations only. Hence, the situation in the German-speaking part of Switzerland is described as diglossia; two languages or varieties of a language are used under different conditions (Siebenhaar and Wyler, 1997). Swiss Standard German is expected to be used in schools as well as some official news broadcasts in TV and radio. With the introduction of digital communication and social media, Swiss German is used increasingly in written form. What started in the context of text messages among youngsters and adolescents with the introduction of mobile phones soon spread over most generations and communication forms.

Some authors like e.g. Lenz (2013); Kaiser (2012); Schobinger (2014) publish entire novels in their dialect. Some special editions of news papers like *Blick am Abend* (Ringier AG, 2013) and even company reports of *Swatch* (The Swatch Group AG, 2012) or *Luftseilbahn Jakobsbad-Kronberg* (Luftseilbahn Jakobsbad-Kronberg AG, 2016) were issued in dialect. Swiss German literature has a long tradition with many books published[2]. However, the expansion to media, annual reports etc. is a recent phenomenon.

The spread of dialect use to cover almost all communication situations raised the desire to be able to automatically process different dialects as well. The interest of companies has grown, eager to be the first ones to "understand" dialect speech with automatic speech recognition and hence conquer the

[2] https://www.idiotikon.ch/literatur/mundartliteratur



Swiss market, to analyse social media data in order to get real feedback for their newest products. This is making the need for Swiss German NLP more pressing. The ongoing interest in Swiss dialects is also opening doors for industry startups which specialise in commercial software able to deal with Swiss German. The field of Automatic Speech Recognition (ASR) has especially been active, with players like the companies *spi:tch*[3] and *recapp*[4].

2.1.1. Typological Classification

Swiss German is a dialect continuum belonging to the Alemannic group of dialects (except one special case in Samnaun, Engadine). This group forms part of the Germanic languages and by extension of the Indo-European language family. Alemannic dialects are also spoken in Alsace, in the south-western part of Baden-Württemberg, in Liechtenstein and Vorarlberg (Glaser, 2003). The Alemannic dialects can be split into three dialect groups namely Low, High and Highest Alemannic; for instance, dialects from the Basel region belong to the Low Alemannic group and southern and western Swiss dialects to the Highest Alemannic group. For the sake of simplicity, dialects are referred to as dialects of the different cantons although the linguistic features are not congruent with political borders and features are spread differently. However, it bears pointing out that the dialectal differences increase with geographical distance and that in rather remote areas (such as the valleys in Valais) the dialects differ to a greater degree compared to other areas characterised by more (linguistic) interchange due to increased mobility of the population Of course, the Swiss German dialects are also influenced by other languages, especially in the western part of the German-speaking area where French evidently has (had) some impact.

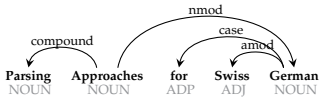
2.1.2. Differences Between Dialects

Differences between the dialects can be found in every aspect. Starting with the most obvious one, of course there are differences in the lexicon like e.g. *hitzgi* vs. *gluggsi* for *hickup*. There are also mere gender differences as in the word *coffee es kafi* (neuter) vs. *en kafi* (masculine). Beyond these, pronunciation can differ significantly, especially regarding vowels. Measuring the formants (f1 and f2) of `/a/` and `/ä/` in two dialect regions, Aepli and Allemann (2016) found a significant difference in their frequencies. In the region of Bern, the `/ä/` is lower, i.e. more open than in the region around St. Gallen while the `/a/` is produced further back. While the `/ä/` around St. Gallen is quite close to the Standard German one, the Bernese `/ä/` is so different that it can be classified as a different vowel which does not even exist in Standard German. The *Dialäkt Äpp*[5] for example makes use of such differences in order to locate a given dialect utterance. Furthermore, there are syntactic differences like the ordering of verbs and auxiliary verbs, or different ways to express final clauses. Figure 2.1, taken from Hollenstein and Aepli (2014) shows examples for these cases. The inversion of verbs *la ga* vs. *gha lah* for instance, as well as auxiliaries *het gha* vs. *gha het*. It also demonstrates the expression of final clauses with *zum* vs. *für ... z*, the latter being influenced by the French *pour*.

[3] https://spitch.ch/

[4] https://www.recapp.ch/de/

[5] http://www.dialaektaepp.ch/index.html



Dialect around Bern:	Si **het** ne **la ga**, wüu er ne gnue Gäu **het gha**, **für** es Billet **z'**löse.
Dialect around Zurich:	Si **hät** ihn **gah lah**, wil er nöd gnueg Gäld **gha het**, **zum** es Billet löse.
Standard German:	Sie **liess** ihn **gehen**, weil er nicht genug Geld **hatte**, **um** ein Billet **zu** kaufen.
English:	She **let** him **go** because he **did** not **have** enough money **to** buy a ticket.

Figure 2.1: Differences between dialects and Standard German (Hollenstein and Aepli, 2014).

2.1.3. Missing Orthographical Rules

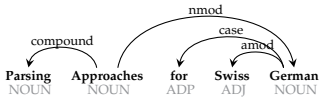
However, the aforementioned differences are not the only challenge when dealing with Swiss dialects. As there is no writing standard, people spell as they please without taking care of consistency. This means that one and the same person would write the same word differently – according to the device they are using or just without any reason. What makes language processing especially challenging is the random merging of words, which means that white spaces cannot be relied on in order to find word boundaries. Also in Standard German, words are frequently joined to form compound words, i.e. words consisting of more than one stem act as one word with one part of speech like e.g. *Schneeschuh* (*snow shoe*). However, in Swiss German merging words rather resembles the phenomenon of clitics, i.e. phonologically bound words (Loos et al., 2004) like e.g. *chömmer* (*can we*). This word cannot be split because, separated, the parts would be the verb *chönd* and the pronoun *mir*. Thus, these words consist of different part-of-speech tags, they are grammatically different words which are phonologically bound and therefore cannot stand alone. A phonological word (transcribed as an alphabetic string delimited by white spaces) can even be more complex and contain the subject, object as well as the finite verb of the sentence. An example for this is *hätsen* consisting of the verb *hät* (*to have* in the third person singular), the pronoun *si* (*she*) as subject and the pronoun *en* (*him*) as object. Such problems often arise when dealing with actual speech and thus also apply to dialects which in turn are the written form closest to actual speech. However, dealing with actual speech also introduces additional issues as the transcription of an utterance varies a lot and depends on the dialectal background of the transcriber (Zampieri et al., 2017).

2.1.4. Differences to Standard German

As with the differences within the Swiss dialects, the differences between Swiss German and Standard German concern every aspect of the languages. Starting with phonetics, one obvious difference is the *ch* which is the velar consonant χ[6] in Switzerland but a palatal consonant ç or alveolo-palatal consonant ɕ in Germany. As with the different dialects, there are obviously differences in the lexicon, which even required the introduction of a new part-of-speech tag not present in Standard German (see Section 3.1.1). In some cases where the same words are used, the gender changes. The "radio" for instance is of masculine gender in Switzerland instead of neuter in the German standard.

Regarding syntax and morphology, Swiss German simplifies many aspects like tenses or cases and relaxes some requirements like for instance the overt expression of the subject. The Swiss German dialects feature no preterite tense (Präteritum) and the plusperfect (Plusquamperfekt) is used very rarely. The past is simply expressed using the perfect tense (Perfekt), as in Figure 2.1 exemplified by the preterite forms *liess* and *hatte* becoming perfect tense. In addition, the use of auxiliary verbs *to be* and *to have* may differ from Standard German like for example to express *I am cold* with *ich **ha** chalt* instead of the German way of saying *mir **ist** kalt*. Furthermore, there is more freedom in the word order, especially

[6] I'm using the symbols of the International Phonetics Alphabet (IPA): https://www.internationalphoneticassociation.org/.



regarding verbs as we have seen before (see Figure 2.1). Moreover, the explicit specification of the subject in German is not applied in Swiss German the subject can be dropped as in *chunnsch au?* instead of *Kommst **du** auch?*. Usually, in these cases the information about the person is given in the conjugation of the verb instead of overtly expressed. Beyond that, the four cases of Standard German (nominative, accusative, dative and genitive) are not all in use (Siebenhaar and Voegeli, 1997). The genitive case is generally not used apart from a few exceptions in Valais. Instead, it is replaced by a possessive dative or a phrase with prepositions. In order to express something like the Standard German genitive phrase *die Augen des Frosches* (*the frog's eyes*), we would say ***am** frosch **sini** auge* (using the expressive dative) or *d'auge **vom** frosch* (using a preposition). Also, only the dative case is marked with its own determiner and endings for adjectives and nouns whereas nominative and accusative forms only differ in personal pronouns.

2.2. Natural Language Parsing

The term parsing comes from the latin word *pars (orations)*, meaning *part (of speech)*[7] and describes the process of syntactic analysis in Natural Language Processing (NLP).

2.2.1. Syntactic Structures

There are two types of parses; *dependency parse* and *constituency* or *phrase-structure parse*. The latter breaks a sentence into sub-phrases (see Figure 2.2 for an example) while a dependency parse determines the structure by the relation between words and their dependents (like in Figure 2.3).

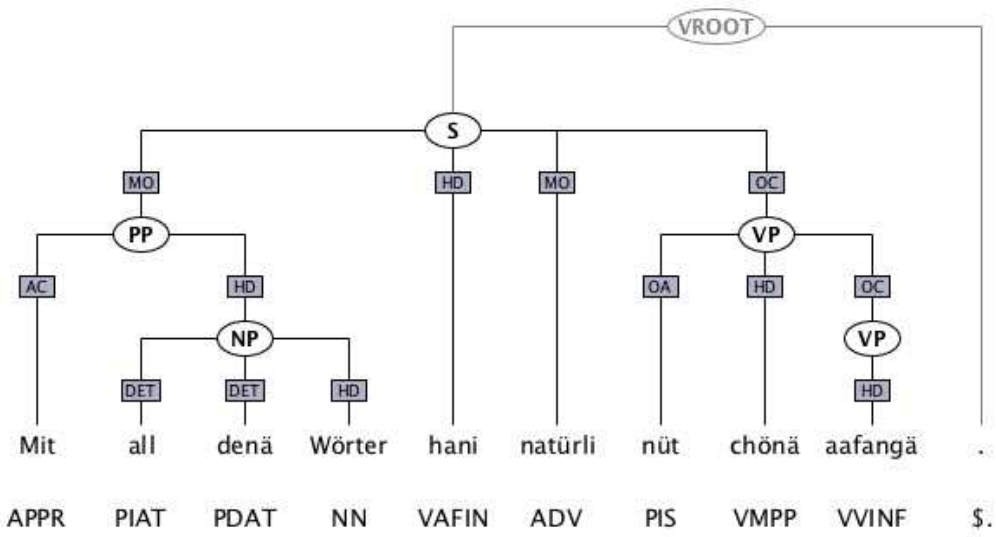
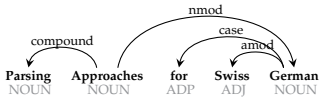


Figure 2.2: Constituency or phrase-structure parse tree breaking a sentence into sub-phrases. Translation: *Of course, I could not do anything with all these words.*

[7] http://www.bartleby.com/cgi-bin/texis/webinator/sitesearch?query=parser



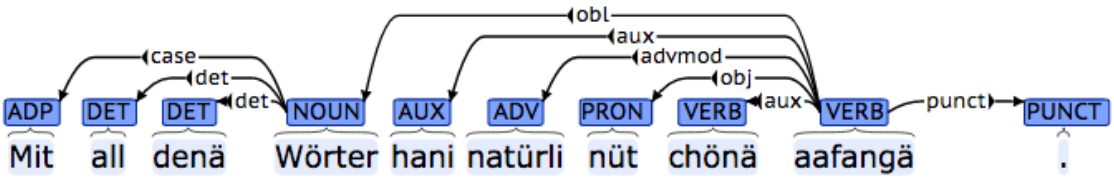


Figure 2.3: Dependency parse tree showing the dependency relations between the words.
Translation: *Of course, I could not do anything with all these words.*

The two Figures (2.2 and 2.3) are only one way of visualising[8] parses: the representation of a parse is dependent on the applied grammar formalism/theory. Figure 2.2 is drawn according to the *TIGER* annotation guidelines (TIGER Project, 2003) and Figure 2.3 according to the *Universal Dependencies* guidelines[9]. The *UD* project provides a linguistically motivated, computationally useful and cross-linguistically applicable standard for dependency treebanks (Jurafsky and Martin, 2017). There are also alternative standards to dependency parsing like for example the constraint dependency grammar for German by Foth (2004). However, I follow the *UD* guidelines because of its cross-linguistic motivation. It is further described in Section 3.2.2.

A phrase-structure parser deals with the way in which sequences of words combine to form constituents. Each node in a parse tree is called a constituent and split into subconstituents according to a specified grammar. The words are grouped into (sub)phrases like for instance a nominal phrase (`NP`) in Figure 2.2, which are further combined with additional words or (sub)phrases like the prepositional phrase (`PP`) or verbal phrase (`VP`) in the example tree. There is always one root node in a sentence (`S`). The arcs between the constituents are labelled with the syntactic relation that holds between them, e.g. adpositional case marker (`AC`). A context-free phrase-structure parse can be represented with parentheses:

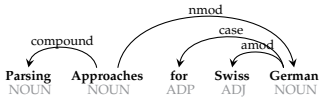
```
S( PP(Mit NP(all denä Wörter)) hani natürli VP(nüt chönä VP(aafangä)))
```

In contrast to constituency structures where words are combined, the grammar formalism called dependency parsing focuses on how words relate to other words. Hence, dependency parse is a distinct and complementary approach. The dependency relation is binary, i.e. holds between two words, and asymmetric, i.e. there is a head and its dependent. A dependency representation is therefore a labelled directed graph `G = (V,A)` with words as vertices (`V`) and labelled arcs (`A`) capturing the head-dependent and grammatical function relationships between the elements in `V`. The arc points from the head to its dependent and is labelled with the corresponding grammatical function. Further constraints on dependency structures are specific to the grammatical formalism, which is usually a dependency tree. A dependency tree is a directed graph with a single root node where every vertex (except the root) has one incoming arc and a unique path from the root to each vertex (Jurafsky and Martin, 2017). In Figure 2.3, the root of the sentence is *aafangä* (*to start*) with a dependent *nüt* (*nothing*) as direct object (`obj`) for instance. An additional constraint derived from the word order is projectivity. Visualising a dependency graph using the standard graphical method, a dependency tree is projective if it can be drawn with no crossing edges (Jurafsky and Martin, 2017).

In a constituency parse, the finite verb is the head of a verb phrase or rather sentence. A dependency parse, on the other hand, does not consider auxiliaries as heads and therefore finite verbs are usually

[8] The visualisation of the constituency parse tree in Figure 2.2 is done with the *TIGERSearch* tool: http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/tigersearch.html. All visualisations of dependency parse trees like in Figure 2.3 in this thesis are done with the *UD* visualisation tool: http://universaldependencies.org/visualization.html.

[9] http://universaldependencies.org/guidelines.html



not the head of the sentence. Hence, the head of a sentence typically is the verb containing the meaning. In that sense, dependency structures are closer to the semantics of a sentence. Also, head-dependent relations provide an approximation to the semantic relationship between predicates and their arguments. Another advantage of dependency grammars is their ability to deal with languages featuring a relatively free word order and rich morphology because they abstract away from word-order information (Jurafsky and Martin, 2017).

2.2.2. Building a Parser

Syntactic parsing is a key area in NLP. Numerous end-user applications like information extraction, grammar checking, question answering or sentiment analysis are built upon its base and hence, depend on a parser's output. To exemplify this, consider the importance of knowing the scope of the little negation particle *not* in a sentence. This information is exactly what a parser provides, which makes it a crucial step for any application.

In order to build a statistical parser, resources are needed. This means either a reasonably-sized treebank, which is a syntactically annotated (i.e. parsed) text corpus, or a human-designed formal grammar. Such kind of resources are rare because they are time-consuming and expensive to build up. The lack of resources and NLP tools is a big problem for the majority of languages. As a consequence, the development of language-specific resources is too costly to be developed from scratch for every language and its varieties. A more efficient approach is to make use of existing resources and tools of a resource-rich language and apply transfer methods in order to get tools and resources for a resource-poor language.

As Swiss German is a small variety with very few resources (up to now), this work focuses on cross-lingual parsing. The two main approaches *model transfer* and *annotation projection* are described in detail in Section 3.3.

2.2.3. Parser Evaluation

In order to evaluate statistical parsers, suitable metrics as well as a gold standard for comparison of the output is needed.

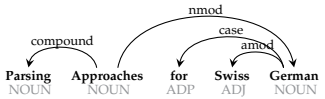
A gold standard for parser evaluation is a small treebank containing manually labelled sentences providing the correct part-of-speech (POS) tags as well as dependency labels for the test set against which the parsers are evaluated. The gold standard I used for this thesis is described in Section 4.2.

The most commonly used metrics for parser evaluation are those from the *CoNLL shared tasks* (Buchholz and Marsi, 2006): *Labelled Attachment Score* (LAS), *Unlabelled Attachment Score* (UAS) and *Label Accuracy* (LA). UAS was introduced by Nivre et al. (2004), and both other metrics by Eisner (1996). LAS provides the percentage of tokens which have been assigned both the correct syntactic head (`HEAD`) and the correct dependency label (`DEPREL`). UAS is the percentage of tokens with the correct syntactic head (`HEAD`) and LA the percentage of tokens with the correct dependency label (`DEPREL`).

2.2.4. Parser for Swiss German

Writing my thesis about parsing in the area of Computational Linguistics, I cannot withhold the famous example of Shieber (1985) where dependency edges cross: *mer wänd d'Chind am Hans s'Huus laa hälfe aaschtriiche*. Due to this sentence, Swiss German is known as a context-sensitive language because of its non context-free phenomenon of cross-serial dependencies.

Although most native speakers of Swiss German would argue that this sentence is a constructed example which nobody would actually ever utter, it has persisted. I do not discuss Shieber's claim



nor his example sentence further. However, during the work on this thesis I was surprised by the complexity of certain phrases and sentences I encountered, even though I was not expecting parsing Swiss German to be a trivial task at any point. Section 4.2.2 shows some empirical evidence for this.

2.3. Crowdsourcing for NLP

Most applications in Natural Language Processing rely on labelled data to obtain statistical or rather machine-learned models. The quality of the input for such an approach is crucial because a trained model can only be as good as the annotations it is fed with. However, obtaining high-quality labels manually annotated by a professional is usually not an option due to a costly process (regarding time as well as money). Therefore the alternative is crowdsourcing. To overcome the quality issue this approach generates, the strategy is to make use of quantity and majority vote because redundancy in data filters out noise (Wang et al., 2013).

A central point in crowdsourcing is the motivator in order to make the crowd participate. One obvious motivator is money like e.g. in *Amazon Mechanical Turk*[10]. However, as usually crowdsourcing in NLP is used because of a lack of funds, a more suitable approach is *Games with a Purpose* (von Ahn, 2006) where the main motivator is fun.

My choice of data and methods requires the use of crowdsourcing, which is conducted within the *SNF-AGORA Citizen Linguistics* project.[11] The centre of the project is the online platform *Tour de Suisse: din dialäkt/ton accent*[12] hosting games in order to get data via crowdsourcing. For my project, I provide registered users with dialect sentences (for information about the source see Section 4.1) and ask them to translate the sentences into Standard German (as in Figure 2.4). There are many ways to translate a Swiss German sentence into Standard German. Some of them tried to stay as close to the Swiss German original as possible, others produced a nice sentence closer to Standard German language usage. The user-provided sentences are not gold standard translations but as we get several translations for each sentence, the result is a distribution of possibilities.

[10]https://www.mturk.com/mturk/welcome

[11]https://www.linguistik.uzh.ch/de/forschung/agora.html

[12]http://www.dindialaekt.ch

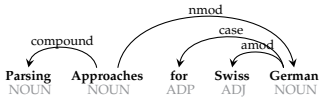


Figure 2.4: Website *dindialaekt.ch* with the translation task: Given the Swiss German sentence, translate it to Standard German.

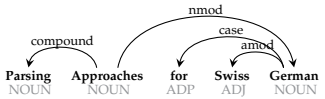
In the game statistics we could see that the activity of the players reduced at some point, which was unfortunate as it happened before I could upload the data I actually wanted to use. So I got the permission to send an email to the 30 top scorers of the translation task in order to ask them if they would consider translating some more sentences. In the email I described my project and the purpose and got a lot of positive feedback. The top contributing users were in general very interested about the project, asked more questions and provided me with some of their insights. Apparently it was very motivating for them to see that their work is appreciated and used for certain research projects.

2.4. Summary

Dealing with a non-standardised language is a demanding task, especially for statistically-driven systems. The huge variety in the Swiss German dialects together with the lack of orthographic spelling rules is a challenging mixture.

Automatically analysing the syntax of a sentence and building a dependency structure is the task of a dependency parser. The state-of-the-art method for parser creation is to use treebanks in order to train machine learning systems. Supervised machine learning methods like statistical parsers require labelled data. As annotated resources are rare, the tools of resource-rich languages are used together with cross-lingual transfer methods in order to get tools and resources for related resource-poor languages.

The *Universal Dependency* project provides cross-linguistically valid guidelines in order to facilitate the application of cross-lingual transfer methods.



3. Related Work

In this section, I present some relevant related work concerning NLP for Swiss German (Section 3.1), the *Universal Dependencies* project (Section 3.2) as well as cross-lingual parsing (Section 3.3). Cross-lingual parsing is a very active field, which is emphasised by the fact that, in 2017 only, there were two shared tasks: The *CoNLL 2017* shared task *Multilingual Parsing from Raw Text to Universal Dependencies* as well as the *Cross-lingual Dependency Parsing* shared task at *VarDial 2017*. This section will therefore only present the most relevant work going on in the area of parsing in NLP.

3.1. NLP for Swiss German

There have been several projects involving Swiss German. Within the projects *sms4science* (Dürscheid and Stark, 2011) and *What's up, Switzerland?* (Stark et al., 2018), resources are generated on the base of actual short text messages and whatsapps, which also contain a lot of dialect messages. Further corpora (i.e. collections of text) have been compiled like the *NOAH* corpus (Hollenstein and Aepli, 2014), consisting of written Swiss German texts in different genres and the *ArchiMob* corpus (Samardzic et al., 2016), a corpus of spoken Swiss German aligned with transcriptions. Several scientific projects which included Swiss German have been carried out, for example *SNF-AGORA Citizen Linguistics*[13], *Kleiner Sprachatlas der deutschen Schweiz*[14] as well as *Syntaktischer Atlas*[15]. Also, there is an institute for documenting and researching the Swiss German dialects called *Schweizerisches Idiotikon*[16].

Many of aforementioned projects focus on researching linguistic aspects. However, the resources produced can be used for NLP applications; work has been done on POS tagging and dialect identification (Hollenstein and Aepli, 2014; Zampieri et al., 2017; Hollenstein and Aepli, 2015), normalisation (Samardžić et al., 2015; Scherrer, 2007) and morphological analysis (Baumgartner, 2016). In addition, independently of the projects mentioned above, approaches to dialect identification (Scherrer and Owen, 2010), morphology generation (Scherrer, 2013) and dialect machine translation (Scherrer, 2012) have been published. The latter presents work on a rule-based system which accounts for the differences between Standard German and the Swiss dialects.

While NLP applications in the past mostly focused on standardised language, a growing interest in NLP for variational linguistics has emerged recently. The *EACL* conference (*European Chapter of the Association for Computational Linguistics*)[17] 2017 hosted the *VarDial 2017 - Fourth Workshop on NMLP for Similar Languages, Varieties and Dialects* offering shared tasks including Arabic and German dialect identification among others (Zampieri et al., 2017).

Related to parsing, Klaper (2014) trained a dependency parser for Swiss German dialects, based on the *NOAH* corpus (version 1). This work was conducted as a term project and one of the conditions was to use manually tagged data (in this case 10,000 tokens). Hence, this supervised parser was trained on 500 sentences which is a very limited amount of data. The treebank was created by the author and consists of unlabelled dependency relations only. The reported accuracy of 56% for supervised learning on blog texts, and 61% with additional unsupervised information as well as the performance of around 75% for edited text like *Wikipedia* articles further show the limitations of this approach.

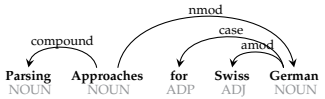
[13]https://www.linguistik.uzh.ch/en/forschung/agora.html

[14]http://www.ksds.uzh.ch/de.html

[15]http://www.dialektsyntax.uzh.ch/de/projekt.html

[16]https://www.idiotikon.ch/

[17]http://eacl2017.org/



Due to the lack of a sufficient amount of gold standard data to use for supervised approaches, in this thesis I will take advantage of the fact that there are resources available for the closely related Standard German. Making use of available treebanks and parsers for Standard German, I will train a parser on the base of projected annotations and hence aim to overcome the resource limitations. Furthermore, the sentences will be syntactically annotated with dependency labels in order to produce a *UD*-conform treebank for Swiss German dialects.

3.1.1. POS tagging for Swiss German

We have worked on POS tagging for Swiss German and provided resources in the form of the *NOAH* corpus[18] as well as pre-trained models (Hollenstein and Aepli, 2014, 2015). This work has been going on since and we are correcting and expanding it. Furthermore, we changed the POS tagger from *BTagger* (Gesmundo and Samardžić, 2012) to *Wapiti* (Lavergne et al., 2010), with which the results are better. The *Wapiti* model has an accuracy of 92.25% on the *NOAH* corpus, measured by 10-fold crossvalidation, while *BTagger* reaches 90.62% (Hollenstein and Aepli, 2014).

The POS annotations in the *NOAH* corpus are generally based on the German guidelines, namely the *Stuttgart-Tübingen-TagSet* (*STTS*) (see Appendix A) and some changes according to the *TIGER* annotation scheme. Furthermore, dealing with Swiss German, there is the need for an additional POS tag `PTKINF`, not present in the *STTS* tagset (Schiller et al., 1999), as well as for the "meta tag" `TAG+`. `PTKINF` is an infinitive particle Glaser (2003) which does not exist in Standard German but is frequently used in dialects. It comes in the form of *go*, *cho*, *goge*, *lo* to name a few, as in *Si gönd* ***go*** *poschte.* (*They go shopping.*) In the Standard German translation, *Sie gehen einkaufen.*, we can see that there is no equivalent. `TAG+` is used for merged words. In the *STTS* there is one tag like this: the `APPRART`, used for combinations of articles and prepositions like *im* consisting of *in* + *dem* (*in the*). However, in Swiss German these kind of merges are performed with any kind of words and just merging the tags would result in a big tagset. Therefore we decided to use the "head" of the word or the first word as tag and simply add a plus to show that this word incorporates another one (Hollenstein and Aepli, 2014). Like this, they can easily be found and, if needed, manually expanded. Frequent examples of such words include *hemmer* (*haben* + *wir*), *häts* (*hat* + *es*), and *sinz* (*sind* + *sie*), for *we have, it has* and *they are*.

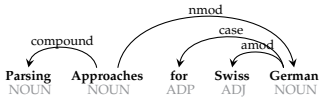
3.2. Universal Labels

Research in dependency parsing has increased significantly since a collection of dependency treebanks has become available, in particular through the *CoNLL* shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007a; Zeman et al., 2017) which have provided many data sets. In order to facilitate cross-lingual research on syntactic structure and to standardise best-practices, *Universal POS* (*UPOS*) Tags (Section 3.2.1) as well as *Universal Dependencies* (Section 3.2.2) have been introduced.

3.2.1. Universal POS Tags

With the idea that a set of syntactic POS categories exists in a similar form across languages, Petrov et al. (2012) have come up with a universal POS (*UPOS*) tagset. The purpose of this tagset is to facilitate future research in cross-lingual tagging approaches.

[18] https://github.com/noe-eva/NOAH-Corpus



ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Figure 3.1: *UPOS* tags

Originally, Petrov et al. (2012) came up with 12 *UPOS* tags, which in *Universal Dependencies version 2* (*UD v2*) extended to 16, as listed in Figure 3.1.

Furthermore, Petrov et al. (2012) provide the mapping from other tagsets to the *UPOS* categories. The mapping for *UD version 2* including all 16 *UPOS* tags is available on the *UD* website[19] and provided in Appendix B.

3.2.2. Universal Dependencies

The goal of the project *Universal Dependencies* (*UD*)[20] by Nivre et al. (2016) is to develop cross-linguistically consistent annotated treebank in order to facilitate multilingual parsing research. The annotation scheme is originally based on *Stanford dependencies* (de Marneffe et al., 2006; de Marneffe and Manning, 2008; de Marneffe et al., 2014). McDonald et al. (2013) present the first collection of six treebanks with homogenous syntactic dependency annotation, which has continually been expanded since.

The relations originally described by de Marneffe et al. (2014) have been revised in *UD v2*. The resulting 37 universal syntactic relations used in *UD v2* are listed in Appendix C.

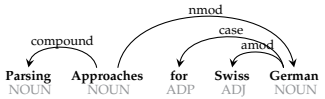
The changes from *UD v1* to *v2* of the universal guidelines are described in detail on the *UD* website[21] including the changes of the *UPOS* tagset. Changes include for example the differentiation between nominals modifying nominals (`nmod`) and oblique dependents of predicates (`obl`). Also, coordinating conjunctions and punctuation inside coordinated structures are attached to the immediately succeeding conjunct instead of the first conjunct.

CoNLL Format: The *10th Conference on Computational Natural Language Learning (CoNLL-X)* featured a

[19]http://universaldependencies.org/tagset-conversion/de-stts-uposf.html

[20]http://universaldependencies.org/

[21]http://universaldependencies.org/v2/summary.html



shared task on *Multilingual Dependency Parsing*, where the data was provided in `CoNLL-X` format (Buchholz and Marsi, 2006). Sentences are separated by a blank line and each token of a sentence is on a separate line, consisting of 10 fields. *UD v2* uses a revised version of the `CoNLL-X` format called `CoNLL-U`, as described in Figure 3.2. If a field is not available, it is filled with an underscore as placeholder.

ID	word index, integer starting at 1 for each new sentence
FORM	word form or punctuation symbol
LEMMA	lemma or stem of word form
UPOSTAG	universal part-of-speech tag
XPOSTAG	language-specific part-of-speech tag
FEATS	list of morphological features from the universal feature inventory or from a defined language-specific extension
HEAD	head of the current word, which is either a value of ID or zero.
DEPREL	universal dependency relation to the HEAD (root iff HEAD = 0)
DEPS	enhanced dependency graph in the form of a list of head-deprel pairs
MISC	any other annotation

Figure 3.2: `CoNLL-U` format: description of the 10 fields each token consists of.

Apart from some changes of the fields themselves as well as added constraints, the `CoNLL-U` format contains comments (lines starting with hash `#`). Furthermore, multiword tokens are split in the extension of `CoNLL-X` format. This results in an expansion of one multiword token over several lines: the multiword token itself on one line indexed with an integer range. The parts in which the multiword token is split follow each on a separate line with the respective annotations. In German, this happens with `APPRART` tokens, i.e. words consisting of a preposition `APPR` and an article `ART`. Figure 3.3 presents an example of such a case.

```
# sent_id = train-s36
# text = Terminfestlegung am Vortag war kein Problem; keine Wartezeiten.
1 Terminfestlegung Terminfestlegung NOUN NN Case=Nom|Gender=Fem|Number=Sing 7 nsubj _ _
2-3 am _ _ _ _ _ _ _ _
2 an an ADP APPR _ 4 case _ _
3 dem der DET ART Case=Dat|Definite=Def|Gender=Masc,Neut|Number=Sing|PronType=Art 4 det _ _
```

Figure 3.3: Beginning of sentence 36 of the German *Universal Dependency* treebank's training set containing a multiword token (`2-3 am`).

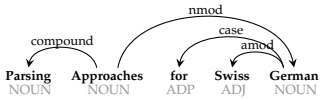
3.3. Cross-lingual Parsing

Cross-lingual learning is usually motivated by a low-resource scenario, which is common. The majority of languages suffers from a lack of resources and tools and the development is too time-consuming and expensive. The idea of cross-lingual methods is to make use of existing tools and data of resource-rich languages in order to create such tools for lower-resourced languages or varieties.

There are two main approaches to cross-lingual syntactic parsing: delexicalized *model transfer* (see Section 3.3.1) and lexicalised *annotation projection* (see Section 3.3.2), the latter including *treebank translation*.

3.3.1. Model Transfer

The goal of the model transfer approach is to abstract away from language-specific parameters, i.e. train delexicalised parsers. The idea is based on universal features and model parameters that can be



transferred between related languages. Hence, this method assumes a common feature representation across languages. The advantage of the model transfer approach is that no parallel data is needed.

Zeman and Resnik (2008) train a basic delexicalised parser relying on POS tags only, as the following paragraph presents. McDonald et al. (2013); Petrov et al. (2012) and Naseem et al. (2010) rely on universal features while Täckström et al. (2013) adapt model parameters to the target language in order to cross-linguistically transfer syntactic dependency parses.

Cross-Language Parser Adaptation between Related Languages: Zeman and Resnik (2008) conceptualise two related languages (or two dialects of one language) as two domains of one "super-language" with different vocabulary but shared morphological and syntactic properties. In their paper, the authors use Danish and Swedish for the experiments. Their delexicalisation approach is based on the hypothesis that the interaction between morphology and syntax in the two languages are similar. As basic approach, the authors replace the Danish words in the training data as well as the Swedish words in the test data with their respective POS tags (the delexicalisation process). Crucial here is that Danish and Swedish use the same tagset. Then, the parser is trained on the delexicalised Danish and run over the delexicalised Swedish data. Afterwards the resulting trees are re-lexicalised with the original Swedish words. With the delexicalised approach, the authors reached an F-score of 66.4% in their setting. In order to assess the success of their result, Zeman and Resnik (2008) compare it to the learning curve of the Swedish treebank and find that they would have needed more than 1500 Swedish parse trees for training in order to achieve the same result.

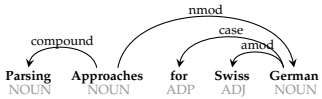
3.3.2. Annotation Projection

The main idea of the annotation projection approach is the mapping of labels across languages using parallel corpora and automatic alignment. It includes projection heuristics and usually post projection rules. The main drawback of this approach is that it relies on sentence aligned parallel corpora. In order to deal with this restriction treebank translation has emerged where the training data is automatically translated with a machine translation system.

The central point of this lexicalised data transfer method is the alignment along which the annotations are mapped from one language to the other. Automatic word alignment has already been used by Yarowsky et al. (2001); Aepli et al. (2014) and Snyder et al. (2008) for improving resources and tools for POS tagging of supervised and unsupervised learning respectively. Hwa et al. (2005), Tiedemann (2014) and Tiedemann (2015) use annotation projection approaches for parsing, and Tiedemann et al. (2014) as well as Rosa et al. (2017) use machine translation in addition instead of relying on parallel corpora.

Bootstrapping Parsers via Syntactic Projection across Parallel Texts: In their article, Hwa et al. (2005) explore the use of parallel text to circumvent the laborious and expensive treebanking process. The idea is to automatically annotate the English side of a parallel corpus, project the analysis to the other language and train a parser on the resulting noisy annotations.

First, the authors formalise the assumption that the syntactic dependencies in source language sentences induce corresponding syntactic dependencies in the target language: The Direct Correspondence Assumption (DCA) "amounts to an assumption that the cross-language alignment resembles a homomorphism relating the syntactic graph of E to the syntactic graph of F", where E and F are translations of each other. Hence, the assumption is that for two sentences in parallel translation, the syntactic relationships in one language directly map to the syntactic relationships in the other. According to this algorithm, the authors use a projection procedure (see Figure 3.4, taken from Hwa et al. (2005)) which



projects the dependencies in a source language sentence to the sentence's translation across word-level alignments.

Given sentence pair (E, F) and a set of syntactic relations for E, where $E = e_1, \ldots, e_n$ is an English sentence and $F = f_1, \ldots, f_m$ is its non-English parallel, syntactic relations (denoted as $R(x, y)$) are projected from English for the following situations:

- **one-to-one** if e_i is aligned with a unique f_x and e_j is aligned with a unique f_y, if $R(e_i, e_j)$,conclude $R(f_x, f_y)$.
- **unaligned (English)** if e_j is not aligned with any word in F, then create a new empty word f_y such that for any e_i aligned with a unique f_x, $R(e_i, e_j) \Rightarrow R(f_x, f_y)$ and $R(e_j, e_i) \Rightarrow R(f_y, f_x)$.
- **one-to-many** if $e_i$is aligned with $f_x, \ldots, f_y$, then create a new empty word f_z such that f_z is the parent of $f_x, \ldots, f_y$ and set e_i to align to f_z instead. We called this a *Multiply-Aligned Component, or (MAC).*
- **many-to-one** if $e_i, \ldots, e_j$ are all uniquely aligned to f_x, then delete all alignments between $e_k (i \leq k \leq j)$ and f_x except for the head of $e_i, \ldots, e_j$.
- **many-to-many** decomposed into a two-step process: first perform one-to-many, then perform many-to-one.

Leave unaligned words in F out of the projected syntactic tree.

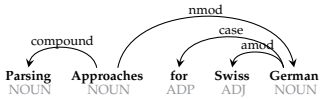
Figure 3.4: The Direct Projection Algorithm (Hwa et al., 2005).

In an earlier paper, Hwa et al. (2002) evaluated the DCA using annotation projection and found that the direct mapping between the syntactic dependencies of two languages cannot be safely assumed. Hence, the DCA itself is not enough, as second-language parses require some monolingual knowledge of the target or rather projected-to language. Therefore, the authors incorporated a set of post-projection correction rules into their projection approach. The rules are motivated by general linguistic properties of the projected-to language and perform local transformations of projected analysis on the target language side.

The final framework for bootstrapping parsers consists of a parallel corpus where the English dependency structures are projected across the word alignment to the non-English side (according to the aforementioned DCA) before the language-specific post-projection transformation rules are applied. In order to solve the problem of propagating English parsing errors as well as word alignment errors, the authors apply a filtering strategy. The remaining projected trees become the treebank which is used to train a new dependency parser for the target language.

Hwa et al. (2002) present two studies where they created a Spanish and a Chinese parser using this framework. The additional language-specific knowledge used in the experiments was created by linguists within a few weeks. The experiments showed that projection of syntactic dependencies across a parallel corpus yields performances that are comparable with a state-of-the-art rule-based Spanish system and a Chinese parser trained on over 2000 noise-free dependency trees respectively, both of which presumably took considerably longer to construct.

Parsing Arabic Dialects: The lack of a standard orthography for the Arabic dialect(s) makes the situation quite similar to the one in the German speaking part of Switzerland. In their paper, Chiang et al. (2006) use Modern Standard Arabic (MSA) tools to parse Levantine Arabic (LA), a spoken dialect.



They assume neither the existence of an annotated LA corpus nor a parallel LA-MSA corpus but instead only use explicit knowledge about the relation between LA and MSA. However, they do have a lexicon relating LA lexemes to MSA lexemes and knowledge about the morphological and syntactic differences between LA and MSA.

Chiang et al. (2006) built three frameworks using MSA corpora for LA parsing: *sentence transduction, treebank transduction and grammar transduction*. In the sentence transduction approach, each LA word is automatically translated into its possible MSA words producing a lattice. The best scoring lattice path is then parsed and the MSA words in the resulting parse structure are replaced by the original LA words. Their implementation of this approach does not handle word order changes between MSA and LA. The idea of the second approach, treebank translation, is to use linguistic knowledge of variations in order to convert the MSA treebank into an LA-like treebank and train a parser on this transduced treebank. The grammar transduction approach uses synchronous grammars, i.e. paired elementary trees, to relate MSA and LA. They use handwritten rules to transform MSA elementary trees into LA elementary trees, resulting in an MSA-LA synchronous grammar which can be used to parse new LA sentences using probabilities of the MSA data.

With these approaches, the authors achieve improvements in parsing quality and stress that they could further improve by using data of better matching domain and genre as well as improved language models among others.

Cross-Lingual Dependency Parsing with Universal Dependencies and Predicted POS Labels: After arguing that annotation projection approaches are more promising than previous results indicate, (Tiedemann, 2014, 2015) presents monolingual and cross-lingual baseline models and discusses the impact of POS tags for parsing.

Comparing gold standard vs. predicted POS labels, the author shows the significant drop of accuracy when using predicted labels in both the monolingual and cross-lingual settings. Tiedemann (2015) shows that delexicalised models are quite robust across closely related languages but useless for distant languages and small training data sets. He concludes that reliable POS tagging is essential for dependency parsing, especially across languages.

Furthermore, Tiedemann (2015) describes dependency parsing experiments in two setups; the classical annotation projection setup as well as treebank translation. Figure 6.8 taken from Tiedemann (2015) illustrates the two approaches and their differences.

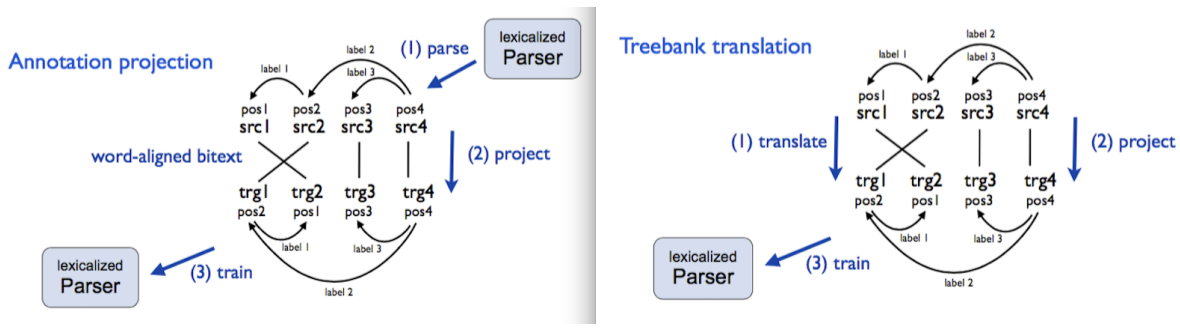
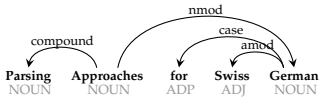


Figure 3.5: The two approaches: *annotation projection* and Treebank Translation (Tiedemann, 2015).

The classical approach to parse source language data with a monolingually trained parser and transferring the automatic annotation to the target language through word alignment comes with certain issues: This approach relies on noisy annotations of the source languages, it requires accurate word



alignments and it needs defined heuristics to treat ambiguous alignments which cannot support one-to-one annotation projection. In this setup, Tiedemann (2015) follows the strategies described in Tiedemann et al. (2014), which are based on the direct projection algorithm of Hwa et al. (2005) with the difference that Tiedemann et al. (2014) makes use of additional information provided by Statistical Machine Translation (SMT) to avoid dummy-nodes.

The treebank translation approach, as presented by Tiedemann et al. (2014), can be seen as creating synthetic parallel corpora. Treebank translation is based on automatically translating training data to a new language in order to create annotated resources directly from the original source. In this setting, the advantages are that the source language annotation is given and the word alignment is provided as an integral part of SMT. Also, the SMT output is closer to the input than manual translations because of the human tendency towards literal translations. The experiments by Tiedemann (2015) show that treebank translation is an alternative to the classical annotation projection generating comparable results.

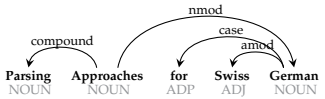
Slavic Forest, Norwegian Wood: Rosa et al. (2017) won the Cross-lingual Dependency Parsing shared task at VarDial 2017 (Zampieri et al., 2017). Their method relies on an automatically word-by-word translated treebank, by which they switch from a cross-lingual to a pseudo-monolingual setting. The core of the switch approach comprises three steps: word-alignment, extraction of a translation table and the treebank translation. For word-alignment they do not use *GIZA++* (Och and Ney, 2003) but the heuristic *Monolingual Greedy Aligner*[22] because it uses word, lemma and tag similarity which makes sense in a setting focusing on similar languages. The authors find their initial hypothesis "that for very close languages, much of the gap between the baseline and the supervised parser can be bridged by appropriate lexicalization" confirmed. Therefore they identify the most important component of their system as the translation of word forms, which leads to an improvement of +5 to +7 LAS.

3.4. Summary

Even though there have been several projects involving Swiss German, resources for NLP applications are still rare. As it is often the case with dialects, data for Swiss German is sparse. Therefore, the approach is to use tools and data of related resource-rich languages and apply transfer methods. In order to facilitate that, the *Universal Dependencies* project provides guidelines as well as resources which are applicable across languages and consistent. Using these resources and following their guidelines will result in an expansion of the resources, facilitate cross-lingual approaches, and hence support resource-poor languages.

Model transfer and annotation projection are the two main approaches to cross-lingual syntactic dependency parsing. The latter includes treebank translation, which is not viable for this project because of sparse data and the lack of a Machine Translation (MT) system for Swiss German. This is due to the huge variety of dialects and spelling approaches, which do not allow for the resources needed to build an MT system. Hence, in this thesis I apply annotation projection as lexicalised approach and model transfer as delexicalised approach.

[22]https://github.com/ufal/treex/blob/master/lib/Treex/Tool/Align/MonolingualGreedy.pm



4. Materials

The first section of this chapter describes the data used for this work: the *German Universal Dependency treebank* and the crowdsourced parallel corpus Swiss German – Standard German (GSW/DE). Section 4.2 describes how I created the gold standard for the evaluation as well as the issues I encountered during the process. Furthermore, I provide some comparisons between the annotated Swiss German and Standard German data sets in Section 4.3.

4.1. Data

The data I worked with consists of a publicly available German treebank (Section 4.1.1) as well as data I collected via crowdsourcing (Section 4.1.2), both of which are described in detail in the following.

4.1.1. German Universal Dependency Treebank

I worked with the *German Universal Dependency treebank* available via the website of the *Universal Dependencies* project[23] or directly from the git repository on *GitHub*[24]. The training set consists of 13,814 sentences. In addition the treebank contains 799 sentences in the development set and 977 in the test set. The treebank was originally converted from the content-head version of the *Universal Dependency treebank v2.0*[25], a set of treebanks annotated in basic Stanford-style dependencies (McDonald et al., 2013). The treebank is annotated according to the *UD* guidelines[26].

The treebank comes in `CoNLL-U` format (see Section 3.2.2), as shown in Figure 4.1, an example sentence taken from the training set of the corpus.

```
# sent_id = train-s2
# text = Die Kosten sind definitiv auch im Rahmen.
1 Die der DET ART Case=Nom|Definite=Def|Gender=Fem|Number=Sing|PronType=Art 2 det _ _
2 Kosten Kosten NOUN NN Case=Nom|Gender=Fem|Number=Sing 3 nsubj:pass _ _
3 sind sein VERB VAFIN Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin 0 root _ _
4 definitiv definitiv ADV ADJD _ 3 advmod _ _
5 auch auch ADV ADV _ 3 advmod _ _
6-7 im _ _ _ _ _ _ _ _
6 in in ADP APPR _ 8 case _ _
7 dem der DET ART Case=Dat|Definite=Def|Gender=Masc,Neut|Number=Sing|PronType=Art 8 det _ _
8 Rahmen Rahmen NOUN NN Case=Dat|Gender=Masc,Neut|Number=Sing 3 obl _ SpaceAfter=No
9 . . PUNCT $. _ 3 punct _ _
```

Figure 4.1: Sentence 2 of the *German Universal Dependency treebank*'s training set. Translation: *The company is located exactly at the entrance to the town.*

As some tools cannot handle `CoNLL-U` format, I converted it to `CoNLL-X` using the conversion scripts on *GitLab*[27]. In the conversion, the lines starting with hashtag (`#`) are removed and the splitting of `APPRART` as described in Section 3.2.2 is undone, i.e. the three lines with IDs `6-7`, `6` and `7` in Figure 4.1

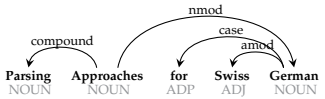
[23] http://universaldependencies.org/

[24] https://github.com/UniversalDependencies/UD_German

[25] https://github.com/ryanmcd/uni-dep-tb

[26] http://universaldependencies.org/guidelines.html

[27] https://gitlab.cl.uzh.ch/siclemat/de-ud-conllu2conllx/



become one line: `6 im in_dem ADP APPRART _ 8 case _ _` and the following word IDs are corrected accordingly.

4.1.2. Crowdsourced Data

The Swiss German data consists of 6,197 sentences, which we presented to the users on the *AGORA* project[28] webpage *dindialaekt.ch* (see Section 2.3) in order to translate them to Standard German. 5,801 sentences are taken from the *NOAH* corpus, 200 from Pedro Lenz' novel *Di Schöni Fanny* and 196 from Renato Kaiser's novel *Uufpassä, nöd aapassä*. *NOAH* corpus contains data from *Wikipedia* (Wikipedia, The Free Encyclopedia, 2011) and the *Swatch* annual report (The Swatch Group AG, 2012), extracts of novels by Viktor Schobinger (Schobinger, 2013), one edition of the newspaper *Black am Abend* (Ringier AG, 2013) and some blog posts from *BlogSpot*. By the end of November 2017, the users generated 41,670 translations. Figure 4.2 shows an example of user-provided translations for the same Swiss German sentence.

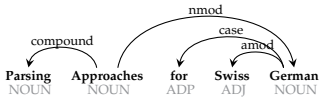
sie wurden abgesetzt, gerade oben auf dem Berg.
sie hatten es abgesetzt auf dem Berg oben
sie setzten es ab, oben auf dem Berg
es wurde auf oben auf dem Berg abgesetzt
Sie haben es gerade oben auf dem Berg abgesetzt.
sie stellten es zuoberst auf dem Berg auf den Boden.
Sie haben es direkt oben auf dem Berg platziert.

Figure 4.2: Different German translations for the Swiss German sentence *si hei s abgsezt gad uf em bäärg obe* (*they put it down right on the top of the mountain*).

In order to work with the cleanest data possible, I filtered out sentences containing certain symbols as well as translations which differ too much in length or Levenshtein edit distance[29] from the Swiss German source sentence. The symbols I filtered out were parentheses, slashes (/) and asterisks (∗) which were used by certain users to give translation options or placeholders for untranslated words for example. As threshold for the Levenshtein edit distance between source sentence and translation, I use the rounded mean of the relative Levenshtein edit distance of all the sentences. This means that I divide the edit distance by the length of the Swiss German sentence in order to avoid penalising long sentences, obtaining a relative measure of 0.5 as threshold. Concerning the length of a sentence, I filtered out all German translations which were more than 30 characters longer or shorter than the original Swiss German sentence. 744 sentences were filtered out because of certain symbols, 1,214 due to the difference in length and 13,697 because the Levenshtein edit distance was too big. The remaining 26,015 sentences – which passed my filter – represent the parallel GSW/DE corpus.

[28] https://www.linguistik.uzh.ch/de/forschung/agora.html

[29] The Levenshtein distance (Levenshtein, 1966) measures the difference between two sequences of characters. Hence, the minimal edit distance between two words is the minimum number of characters to be changed (i.e. inserted, deleted or substituted), in order to change one word into the other.



4.2. Gold Standard

With the purpose of comparing and evaluating the different approaches to parsing Swiss German, a test set is required. In order to be able to draw a conclusion from the evaluation scores, a large test set would be preferable. However, as this is manual annotation work, a big test set is not feasible within this thesis.

The gold standard consists of 100 Swiss German sentences out of which 25 are taken each from Kaiser's, and Lenz's books and 50 sentences from the *NOAH* corpus (10 from each original source). After tokenisation with the German *cutter*[30], I automatically POS tagged them with the current *Wapiti* model trained on the Release 2.2 of the *NOAH* corpus, where average accuracy in 10-fold crossvalidation is 92.25% (see Section 3.1.1). Note that some of the 50 gold standard sentences taken from the *NOAH* corpus are used in the training of the *Wapiti* model for POS tagging. I consider this in the evaluation.

I corrected the automatically annotated POS tags following the *NOAH* corpus guidelines, which are built upon the *STTS* (Schiller et al., 1999) and *TIGER* (TIGER Project, 2003) guidelines. Subsequently, I added *Universal Dependency* POS (*UPOS*) tags[31] according to the mapping provided by the *Universal Dependency* project[32] with the addition of `PTKINF` as `PART`. In cases of `TAG+`, I discarded the plus sign in order to map the *STTS* tag to *UPOS* tags.

In a next step I annotated the sentences manually with universal dependencies using the tool *WebAnno* (Yimam et al., 2013) following the *Universal Dependency* guidelines for the syntax parses. Additional sources to support difficult linguistic decisions were: *Duden*[33], *grammis*[34], *STTS* (Schiller et al., 1999) and *TIGER* (TIGER Project, 2003) guidelines as well as the manually annotated dependency trees of the English *UD* treebank[35].

4.2.1. POS Tagging problems

During the correction of the POS tags one sometimes gets stuck with unexpected special cases which are not trivial to tag. Below are examples of interesting words that made for difficult decisions.

- **z'trinke** as in *als hättisch z'trinken übercho* (eng. *as if you had got something to drink*): The appearance of a *z* in combination with a verb is most likely a *zu*-particle in front of an infinitive (`PTKZU`). However, the *trinken* here is to be tagged as a noun according to me. Thus, the most likely tag for a *z* in front of a noun would be a preposition (`APPR`). Nevertheless, I would argue it is actually a *ds*, written shortly as a *z* because the sound is more or less the same in Swiss German. In that case, the *z* can be tagged as an article (`ART`) and *trinken* as noun (`NN`), which is the most straightforward interpretation in this context.
- **Tschuld** as in *är isch Tschuld* for the Standard German sentence *er ist schuld* (eng. *it's his fault*). At first it seems straightforward: *tschuld* (or *gschuld* in the Zurich region) is an adjective (`ADJD`: adverbial or predicative adjective). However, the author used a capital letter, which poses the question if he had reasons for it. Another version I have come across is *d'schuld*, which would suggest an analysis as article `ART` and noun `NN`. This could be interpreted as a short form of something like *die Schuld*

[30]https://gitlab.cl.uzh.ch/graen/cutter

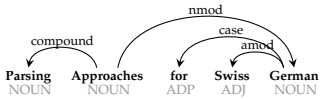
[31]http://universaldependencies.org/u/pos/all.html#al-u-pos

[32]http://universaldependencies.org/tagset-conversion/de-stts-uposf.html

[33]https://www.duden.de/

[34]http://hypermedia.ids-mannheim.de/

[35]http://bionlp-www.utu.fi/dep_search/



haben (**have the fault*). This is also suggested by the *Idiotikon* entry for *d'Schuld*[36]. However, it could also just be an assimilation problem resulting in the added *t* or *g* respectively was added, and the capitalisation was randomly produced. On these grounds, I tagged the word as an adjective..

- **verschwunde** as in *dür (...) sy aber die Ungerschide verschwunde* (eng. *but through (...), these differences vanished*). This is a frequently appearing issue whereby a past participle (`VVPP`) or adjective (`ADJD`) also has an influence on the dependency structure. If it is tagged as `VVPP`, the dependency is auxiliary (`aux`); if tagged as `ADJD`, it is a copula construction (`cop`).

4.2.2. Complex Structures

Even though Swiss German is an oral language, its syntax is not trivial. Usually this fact is exemplified by the Shieber's (1985) sentence (see Section 2.2.4), which is a constructed and therefore disputed sentence. In this section I provide empirical data in order to illustrate the existing complexity.

During the manual annotation process, I found that Swiss German sentences contain surprisingly complex constructions, especially verbal ones. This is partly due to the fact that there is no simple past tense (as mentioned in Section 2.1). Furthermore, there are many ellipses and copula constructions, quite a few coordinated phrases and sentences, crossing edges, incorporated subjects and even sentences with exclusively auxiliary and adverbial modifier dependency edges. In the following, I present some interesting examples appearing in the gold standard.

Looking at the following examples of Swiss German dependency trees, the parses might not seem especially complex for the reader.

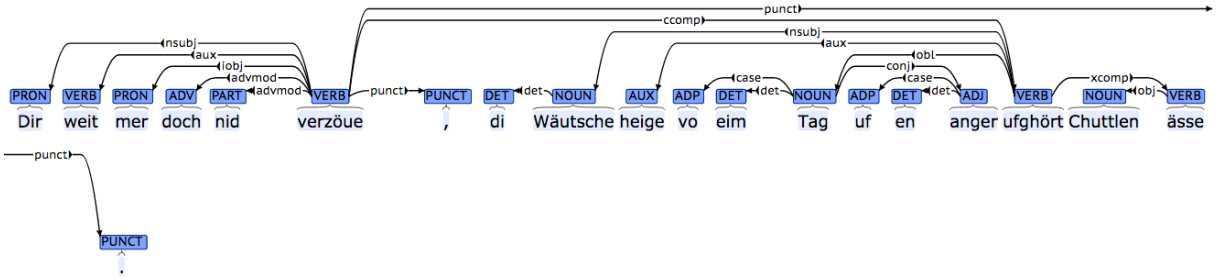
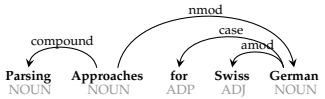


Figure 4.3: *You don't mean to tell me the Welsh stopped eating tripe from one day to the next.*

As mentioned in Section 2.2, the finite verb is usually not the head in content-head dependency structures. Illustrating this point, the head of the main sentence in Figure 4.3 is the infinitive *verzöue* (*tell*) with the finite word *weit* (*want*) as dependent. This is not intuitive as the subject *Dir* (*you*) has to be attached to an infinitive. Furthermore, the structure of this sentence is not trivial: The head of the clausal complement is the past participle *ufghört* (*stopped*) with the infinitive *ässe* (*eat*) as clausal complement, which in turn is the head of an object, namely *Chuttlen* (*tripe*).

[36]https://digital.idiotikon.ch/idtkn/id8.htm#!page/80647/mode/2up



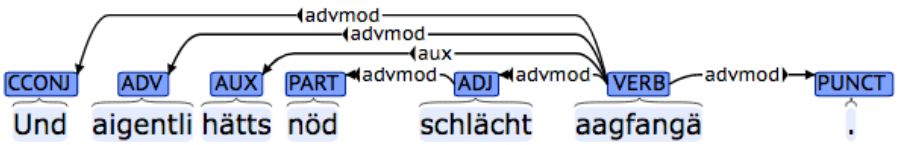


Figure 4.4: *And actually it did not start off badly.*

Figure 4.4 shows the structure of a typical Swiss German sentence which often consist of nothing but several adverbial modifiers and an auxiliary dependency. The subject is incorporated as the *s* of *hätts* (*it has*) and therefore not visible in the dependency structure.

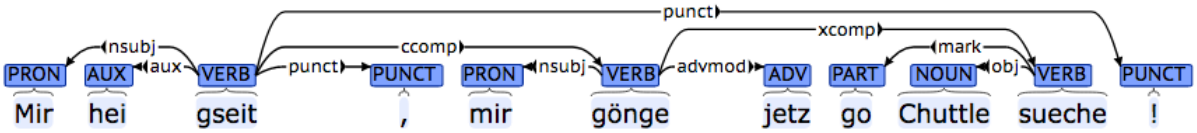


Figure 4.5: *We said we would go looking for tripe now.*

The sentence shown in Figure 4.5 contains the typical Swiss infinitive particle *go* (see Section 3.1.1). I decided to treat it as a marker "introducing a finite clause subordinate to another clause"[37] (`mark`) because they usually appear in *um ... zu* (*in order to*) constructions in German. Although quite short, this sentence contains the clausal complement *mir gönge jetz* (*we would now go*) which in turn contains the open clausal complement *go Chuttle sueche* (*look for tripe*).

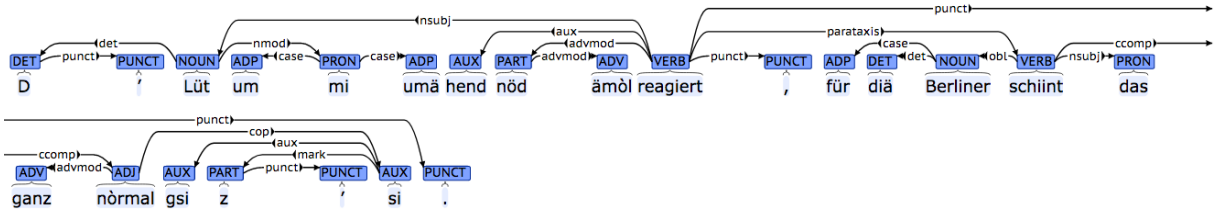
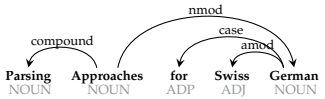


Figure 4.6: *The people around me did not even react; for these people from Berlin this seems to have been perfectly normal.*

The sentence in Figure 4.6 turned out to be a tough nut to crack. It actually consists of two full-fledged separate sentences, connected by a `parataxis` label. The first sentence is quite regular with only a tricky decision concerning the word *umä* (second part of *um ... umä* meaning *around*). However, the second sentence contains a copula construction as clausal complement *das ganz nòrmal gsi z'si* (*this (seems) to have been perfectly normal*).

[37] http://universaldependencies.org/u/dep/mark.html



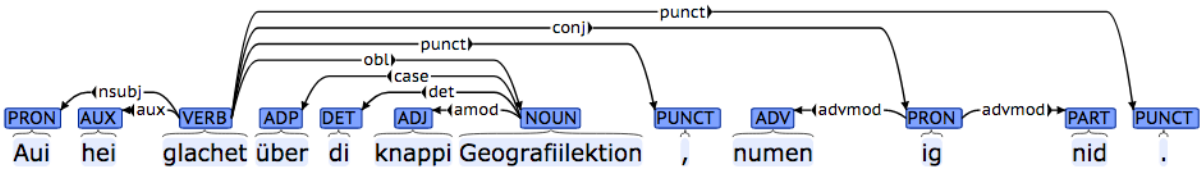


Figure 4.7: *Everybody laughed about the short geography lesson, but I didn't.*

The sentence in Figure 4.7 is a coordination of two sentences connected by the comma. However, the second sentence only contains the subject, an adverb and a particle. Everything else was left out because it would be the same as in the first sentence. The *UD* guidelines specify an approach to ellipsis in clauses in which the main predicate is elided: If there is an `aux` or `cop` (or `mark`), it is promoted to take the role of the head. If no `aux` or `cop` is present, the dependents are promoted in the order `nsubj > obj > iobj > obl > advmod > csubj > xcomp > ccomp > advcl`[38]. For the present sentence, this means that the two heads connected by the conjunction dependency are the verb *glached* (*laughed*) and the pronoun *ig* (*I*) because the predicate of the second sentence is elided and therefore the subject is promoted to take the role of the head.

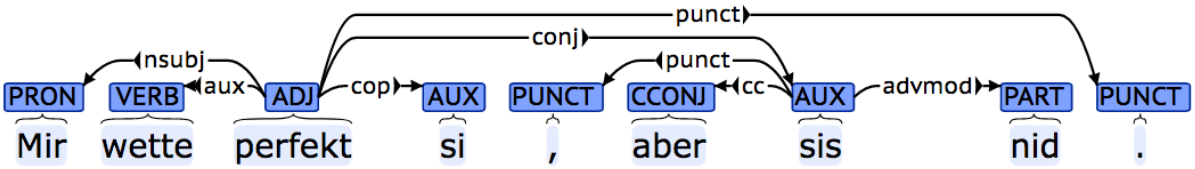


Figure 4.8: *We want to be perfect but we're not.*

Figure 4.8 shows a sentence entirely made up of coordinated copula constructions with ellipses. The auxiliary *sis* (*are*), containing the incorporated object *s* standing for *es* (*it*), is promoted to take the role of the head as the main predicate is elided. Because the `aux` is promoted to the predicate, the negation *nid* (*not*) is dependent on this (second) copula.

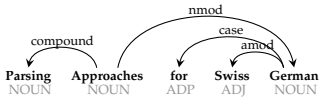
4.3. Quantitative Comparison

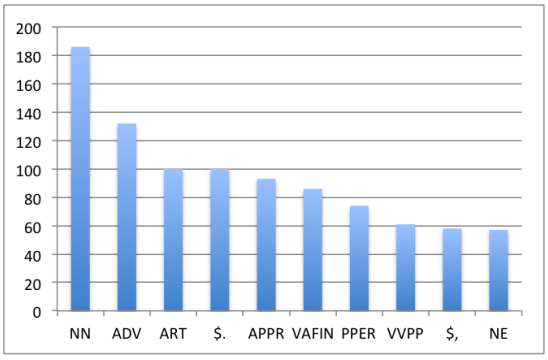
Comparing the (relative) frequencies of POS tags and dependency labels between Standard German and Swiss German, we can observe some peculiar differences. Keep in mind the different sizes of the corpora: the Swiss German gold standard contains 1444 tokens, while the training set of the German *Universal Dependency* treebank has 264,905.

4.3.1. Differences in POS Distributions

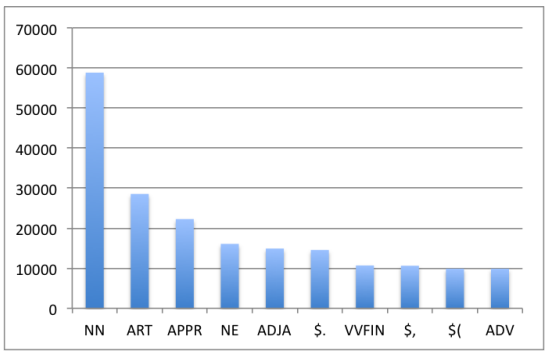
Figure 4.9 shows the frequencies of the ten most frequent *STTS* POS tags in the Swiss German and the German corpora, with the actual frequency counts on the y-axis.

[38] http://universaldependencies.org/u/overview/specific-syntax.html





(a) Swiss German gold standard



(b) German *UD* treebank

Figure 4.9: Frequency histograms of the distribution of the 10 most frequent *STTS* POS tags.

The most frequent part of speech is the noun in both corpora, but in the German corpus 22.21% of the tokens are labelled as nouns while in the Swiss German corpus there are only 12.88%. As expected, in the German corpus the second most frequent POS tag is determiner (`ART`) with 10.78%. However, in the Swiss German corpus, the adverb (`ADV`) is ranked second with 9.14%. This is a striking difference as in the German corpus adverbs occur with a frequency of only 3.73% and are the tenth most frequent POS tag. One hypothesis behind this striking difference in adverb usage could be the discrepancy in oral vs. written language.

The meta POS tag `TAG+` appears 24 times, i.e. with a frequency of 1.7% in these 100 gold standard sentences.

Figure 4.10 illustrates the differences in distribution of the *UPOS* tags over the Swiss German and the German corpora.

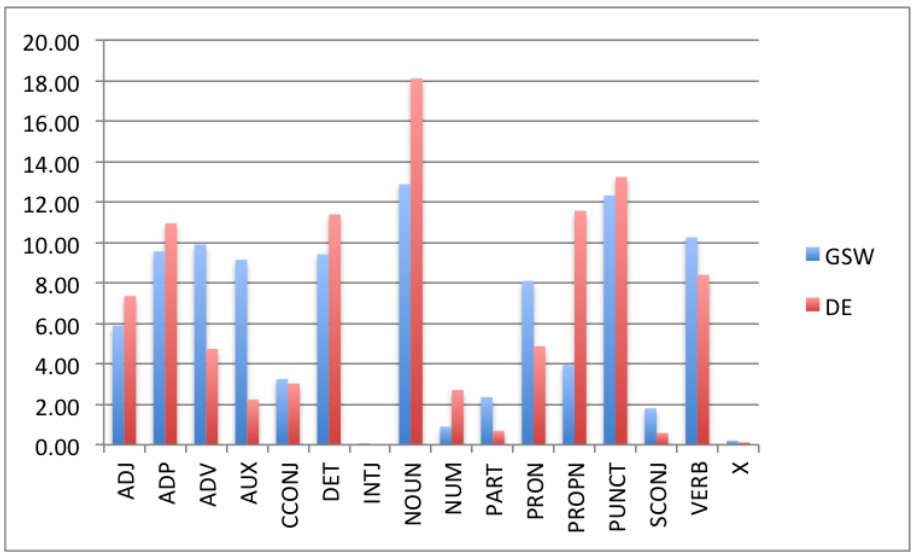
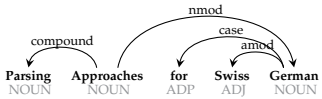


Figure 4.10: Relative frequency (% on y-axis) of *UPOS* tags in the Swiss German and German corpora.

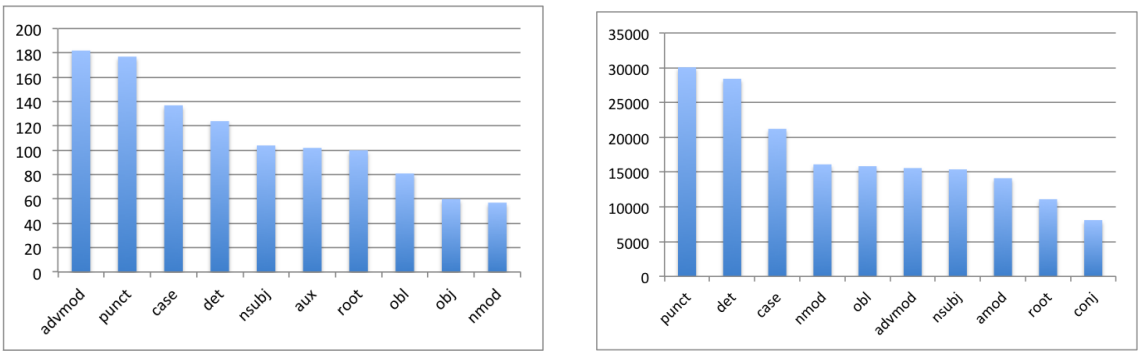
The y-axis in Figure 4.10 shows the percentage of occurrence of the respective *UPOS* tag. This histogram shows that, contrary to the noun and adverb frequencies, ajdectives (`ADJ`), prepositions (`ADP`), coordinating conjunctions (`CCONJ`), determiners (`DET`) and punctuations (`PUNCT`) have quite similar distributions. Furthermore, this histogram shows another interesting fact concerning verbs. The percent-



age of auxiliaries (`AUX`) is almost 7% higher, meaning that 4 times more auxiliaries are used in Swiss German. In addition, almost 2% more verbs (`VERB`) occur in Swiss German texts. This comparison shows that the verbal constructions are more complex in Swiss German. The high amount of auxiliaries is due to the use of perfect tense instead of the German simple past tense as mentioned in Section 2.1.

4.3.2. Differences in Dependency Label Distributions

The frequency distribution of the 10 most frequent universal dependency labels in the Swiss German gold standard and the German *Universal Dependency* treebank are illustrated in Figure 4.11.

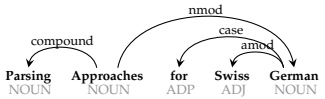


(a) Swiss German gold standard

(b) German *UD* treebank

Figure 4.11: Frequency distribution of the 10 most frequent dependency labels.

In accordance with the frequency distribution of the POS tags, the most frequent dependency label in the Swiss German corpus is the adverbial modifier (`advmod`) occurring 12.6% of the times as shown in Figure 4.11. As a comparison, in the German treebank `advmod` has an occurrence rate of 5.89%. Furthermore, the different usage of tenses is also shown here: in Swiss German, auxiliary verbs make up 7.06% of the dependencies, while in the German corpus it is only 1%. In contrast, punctuation (`punct`), prepositions (`case`) and determiners (`det`) indicate similar frequency distributions.



5. Methods

I worked with the two classical parsing approaches already discussed in detail in Section 3.3: *model transfer* with a delexicalised parser (Section 5.2) as well as *annotation projection* with the help of crowdsourcing to get the required parallel data (Section 5.3). Section 5.1 explains the work of a statistical dependency parser including a short description of the parsers I used. The pre- and postprocessing steps specific to a parsing approach are treated in the sections of the respective parser requiring the information, i.e. POS tagging in Section 5.2.1, word alignment in Section 5.3.1 and transfer of the parse in Section 5.3.2.

5.1. Statistical Dependency Parsing

Unlike grammar-based parsers which build a tree structure according to specified grammar rules, data-driven statistical dependency parsers learn to produce dependency graphs from a given treebank. A sentence's dependency graph represents each word and its syntactic functions through labelled directed arcs (Mcdonald, 2007). In the training phase, statistical parsing frameworks learn a model, which can then be used to parse new input in the application phase. Many such tools, featuring different architectures, are available nowadays.

Depending on the languages and applications, different parsers can be exploited to make use of their strengths and weaknesses. There are mainly two approaches to data-driven dependency parsing: *graph-based* and *transfer-based*.

The former is an algorithm that learns a model from scoring possible dependency graphs for a given sentence. It then performs a near-exhaustive search over a dense graphical representation of the sentence in order to find the dependency graph which maximises the score (Mcdonald, 2007). The Maximum Spanning Trees (MST) algorithm exhibits this architecture (Mcdonald et al., 2005).

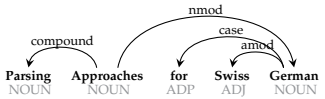
In contrast, transition-based approaches learn a model for scoring parsing transition actions (e.g. shift, reduce) from one state to the next, conditioned on the history and current input. They feature a greedy inference algorithm by taking the highest-scoring transition of every state. These parsers define features over a rich history of parsing decisions. *MaltParser* is an example of such a system (Mcdonald, 2007).

One important aspect of dependency trees is projectivity, especially for languages with more flexible word order where non-projective dependencies are more frequent (Mcdonald et al., 2005). Projectivity can easiest be explained by means of the visualisation of a dependency tree (see Section 2.2.1), as Mcdonald et al. (2005) do: "If we put the words in their linear order, preceded by the root, the edges can be drawn above the words without crossings." According to Buchholz and Marsi (2006), the German *CoNLL-X shared task* training data for example contains 2.3% non-projective dependencies and 27.8% non-projective sentences and thus shows a need for a strategy to handle non-projective trees.

The less flexible transition-based approaches can only produce projective trees (Jurafsky and Martin, 2017) while graph-based approaches can handle more complex non-projective parse trees. However, Nivre and Nilsson (2005) describe ways to overcome the restrictions by using graph transformation techniques in pre- or postprocessing steps.

In this work I test two parsing frameworks; the *MaltParser* (Nivre et al., 2007b) and the more recent *UDPipe* (Straka and Straková, 2017), described in the following subsections. Both parsers are provided with tokenised input, generated by the tokeniser *cutter*[39] at test time.

[39]https://gitlab.cl.uzh.ch/graen/cutter



5.1.1. MaltParser

MaltParser (Nivre et al., 2007b) is based on the transition-based approach to dependency parsing and induces a model from the provided treebank during training. The parser is a complex system which requires optimisation for many parameters in order to work well. To find the best settings, *MaltOptimizer* (Ballesteros and Nivre, 2012) can be applied which searches the best combination for the parsing and learning algorithm as well as the feature model.

MaltParser takes as input files in `CoNLL` format, where POS tags and optionally also morphological information and lemmas are given. Missing or unspecified information is represented by underscores as place holders. Dependency information is given in two columns: the id of the word's head and its corresponding dependency label.

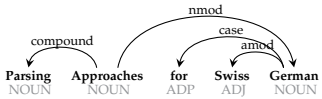
5.1.2. UDPipe

UDPipe (Straka and Straková, 2017) is a pipeline for tokenisation, tagging, lemmatisation and dependency parsing and can be trained by giving it annotated data in `CoNLL-U` format. The parsing step itself is performed with *Parsito* (Straka et al., 2015), a transition-based parser using a neural network classifier. *UDPipe* was used as baseline in the *CoNLL 2017 Shared Task Multilingual Parsing from Raw Text to Universal Dependencies* (Zeman et al., 2017) and as such on the 13th place out of 33 competing systems. The best-scoring systems of the shared task are not available for use.

UDPipe takes as input plain or tokenised text and produces `CoNLL-U` format as output. Using the whole pipeline, POS tags cannot be provided, all columns except the first one of a tokenised input are ignored by the *UDPipe*.

5.2. Delexicalised Model Transfer Approach

As visualised in Figure 5.1, the delexicalised approach is straightforward; the parser works on the basis of POS tags only. The words in the German training corpus are replaced with their *STTS* POS tags before training. Accordingly, the Swiss German words are replaced by their *STTS* POS tags at test time before parsing and re-inserted afterwards.



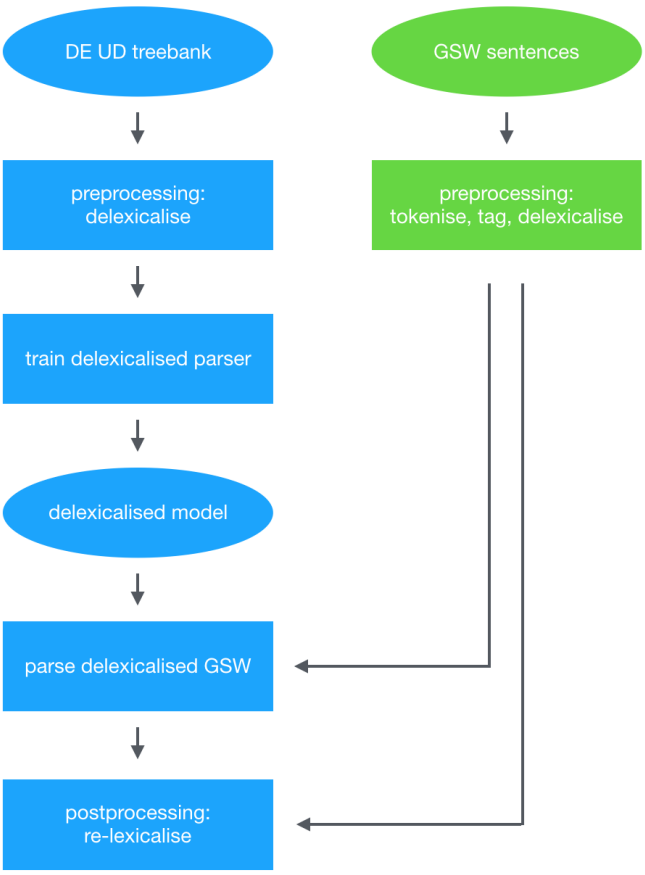


Figure 5.1: Workflow of the delexicalised approach.

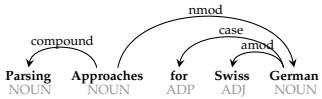
5.2.1. Part-of-Speech Tagging

Part-of-speech tagging is the process of labeling tokens, i.e. words and punctuation, according to a label set and its corresponding guidelines. In this process, the frequency distribution of the POS tags of a word, which were observed during training, as well as its context, are considered in order to decide upon its label. The context of the word helps in disambiguating word forms as they are often ambiguous (Schmid, 1994).

POS tagging is an important step prior to parsing because the syntactic structure builds upon the POS information. Obviously, when training delexicalised parsers, this step is crucial as the tags are the only information available to the parser.

For POS tagging Swiss German sentences I used the *Wapiti* model described in Section 4.2. In order to provide the *STTS* POS tags for the Standard German sentences of the parallel corpus, I used the pre-trained *TreeTagger* (Schmid, 1994) model for Standard German. The *TreeTagger* tags sentences according to the *STTS* guidelines (Schiller et al., 1999) (see Appendix A).

The `CoNLL`-format includes *UPOS* tags in addition to the language specific POS tags (*STTS* in the case of German and Swiss German). As mentioned in chapter 3.2.1 I used the mapping provided by the *UD* project (see Appendix B) in order to infer the *UPOS* tags from the given *STTS* tags.



5.3. Annotation Projection

Annotation projection is not only more complex in processing but also requires more resources. Most importantly, annotation projection needs a word-aligned parallel corpus. As described in Sections 2.3 and 4.1.2, we created a parallel corpus with the help of crowdsourcing within the *AGORA* project[40]. In addition, the corpus has to be word-aligned. This entails that for every word in the source sentence, the information which word of the target sentence (translation) it corresponds to has to be presented. This is the task of a word aligner, which can be accomplished using different approaches. As annotation projection crucially depends on the quality of word alignment, I tested three different word aligners (*GIZA++*, *FastAlign* and *Monolingual Greedy Aligner*) described in the subsequent section.

Figure 5.2 shows an overview of the processes I applied in the annotation projection approach for dependency parsing.

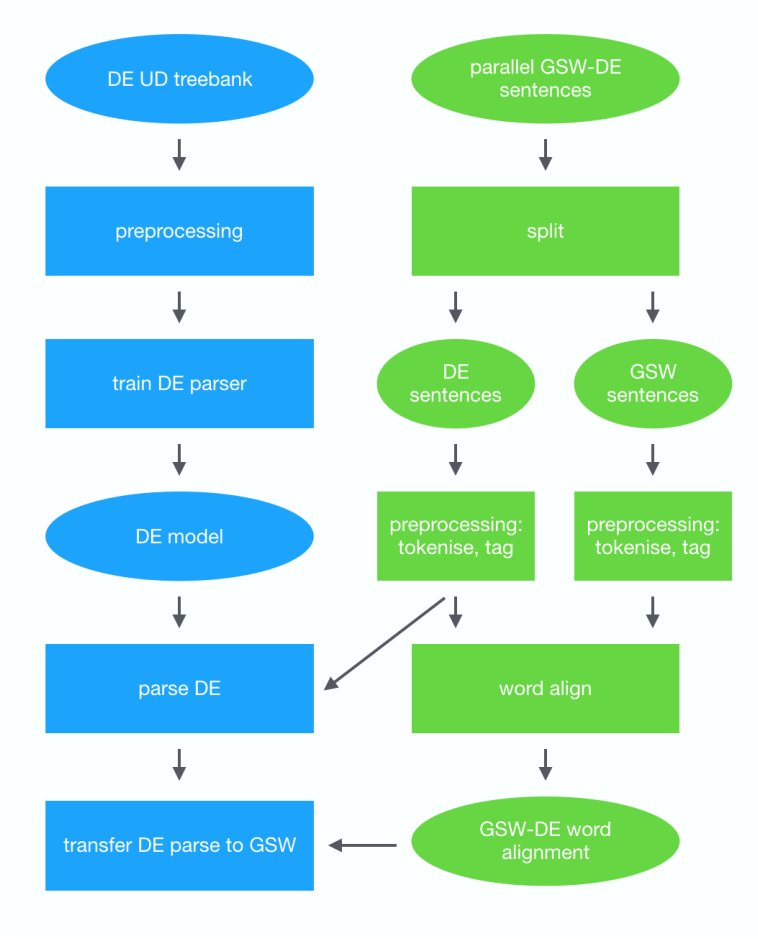
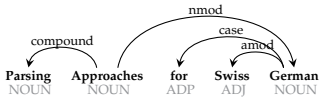


Figure 5.2: Workflow of the annotation projection approach.

The idea of the annotation projection process is to use the tool (here: parser) of a resource-rich language on that language (here: German) and then project the generated information (here: universal dependency structures) along the word alignment to the target language (here: Swiss German). In practice, this means I trained the parsers on the Standard German treebank (see Section 4.1.1) and

[40]https://www.linguistik.uzh.ch/de/forschung/agora.html



parsed the Standard German translations of the Swiss German original sentences. Then I projected the resulting parse structure along the word alignments from the German word to the corresponding Swiss German word. This transfer process is described in detail in Section 5.3.2.

5.3.1. Word Alignment

A word aligner works on a sentence level; it takes a sentence and its translation as input and computes the most probable word alignments, i.e. the information about which word of the (Swiss German) source sentence corresponds to which word of the target sentence, i.e. the translation. There are many tools for this as it is a basic step also in machine translation systems. I worked with three of them, each of which expecting a different input format and producing one of two different output formats.

GIZA++, the most popular word aligner, expects two text documents as input, with one tokenised sentence per line and generates a *GIZA++*-specific output format (see Figure 5.3).

```
# Sentence pair (7) source length 6 target length 5 alignment score : 0.0140654
Das war echt beeindruckend .
NULL ({ }) da ({ 1 }) isch ({ 2 }) echt ({ 3 }) beidruckend ({ 4 }) gsi ({ }) . ({ 5 })
```

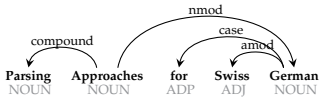
Figure 5.3: Example output of the word aligner *GIZA++*. Translation: *That was really impressive.*

FastAlign is a simple log-linear reparameterization of *IBM Model 2* (Dyer et al., 2013). This aligner can be trained, which I did on 26,015 parallel GSW/DE sentences (see Section 4.1.2). However, the annotation projection results based on the output of the trained aligner are, counterintuitively so, worse than the results based on the output of the untrained aligner.

FastAlign takes one text file as input, with one source sentence and its translation per line, separated by three pipes with leading and trailing white spaces. The output it produces is the widely-used `i-j` format meaning the `i`th word of the source language is aligned to the `j`th word of the target language. The *FastAlign* output for the sentences in Figure 5.3 is `0-0 1-1 2-2 3-3 4-3 5-4` (where the first number stands for the `n`th word of the GSW sentence, the second for the `n`th word of the DE sentence `GSW-DE`).

Monolingual Greedy Aligner (MGA) was used by Rosa et al. (2017) in the *VarDial* 2017 shared task *Cross-lingual Dependency Parsing* (Zampieri et al., 2017). While the majority of the other participants used the classical *GIZA++*, the authors decided to use *MGA* because it does not ignore word similarity, as most other aligners do. In this context, word similarity is not about semantic similarity but about similar letters and length, which *MGA* measures by means of the Jaro-Winkler distance (**?**). *MGA* operates with the information about the word form or lemma, the morphological tag similarity, the similarity of the relative position in the sentence, and an indication whether the source context words were already aligned to the target context words (Rosa et al., 2012). Furthermore, this aligner enforces one-to-one alignments, i.e. it never aligns several target words to one source word `1:n`, but `n>1` as well as zero alignments exist.

The *Monolingual Greedy Aligner* exploits lemma and POS information for the alignment and therefore requires sentence-aligned texts as input, where every token of the sentence consists of word, POS tag and lemma, separated by slashes. The output format is `i-j` format where the `i`th word of the source language is aligned to the `j`th word of the target language. *MGA* produces the alignment `0-0 1-1 2-2 3-3 4-5` (where the numbers represent the words' indices in the respective sentences `DE-GSW`) given the sentences in Figure 5.3.



For easier reading, I convert the output formats in a two column format with the Swiss German word in the first column and the aligned Standard German word(s) in the second column. Figure 5.4 shows the converted alignment of the aforementioned example sentence, where the alignment *gsi - beeindruckend* is only produced by *FastAlign* (check original aligner outputs above for comparison).

da	Das
isch	war
echt	echt
beidruckend	beeindruckend
gsi	beeindruckend
.	.

Figure 5.4: *FastAlign* word alignments for the sentence: *that was really impressive.*

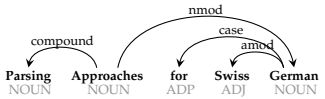
Figure 5.5 shows another example sentence taken from the gold standard along with the output of each aligner, which differ considerably. For `1:n` alignments, I used `£$` as a separator in the "human readable" alignment format.

GSW Sentence	**GIZA++**	**MGA**	**FastAlign**
;)	;)	;)	;)
Am	Am	Am	Am
Obig	Abend		Abend
simer	gingen £$ wir		gingen
den	dann	dann	gingen £$ wir
no	noch	noch	dann £$ noch
e	eine	eine	eine
Wassershow	Wassershow	Wassershow	Wassershow
vo		vom	
dem	vom		Wassershow
riesige	riesigen	riesigen	vom
Brunne	Brunnen	Brunnen	riesigen £$ Brunnen
vom	vom	vom	vom
Bellagio	Bellagio	Bellagio	vom £$ Bellagio
go			Bellagio
luege	schauen		schauen
.	.	.	.

Figure 5.5: Alignments of the three aligners. Translation: *;) In the evening we went to see a water show of this huge fountain of the Bellagio.*

5.3.2. Transfer of the Annotation

The transfer is the core component of annotation projection. The parse of the Standard German translation is projected along the word alignment to its Swiss German correspondent. The input consists of the Standard German parse and the alignment between the Standard German sentence and its Swiss



German version (`GSW:DE`). The following algorithm describes the projection process:

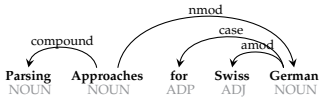
Data: DE parse & alignment `GSW:DE`
Result: DE parse transferred to GSW
for *word alignment in sentence* **do**
  if *`1:1` alignment* **then**
    transfer parse of DE
  else if *`1:0` alignment (i.e. no DE word aligned)* **then**
    attach GSW word to `root` as POS tag `ADV` and dependency label `advmod`
  else *`1:n` alignment (i.e. several DE words aligned)*
    transfer parse of aligned DE word with smallest edit (Levenshtein) distance
  end
end

The case of `1:1` alignment where exactly one German word is aligned to the Swiss German word is easy; the only thing to do is projecting the dependency of the German word to the Swiss German word. If, however, there are several German words aligned to one Swiss German word (`1:n`), the algorithm has to decide which parse to transfer. In order to take this decision, the algorithm computes the Levenshtein distances (Levenshtein, 1966) between the Swiss German word and every aligned German word and takes the one with the smallest edit distance. The most challenging case is when no German word is aligned to the Swiss German token. A simple baseline approach is to attach the corresponding Swiss German word as adverbial modifier to the root of the sentence.

The decision to regard every unaligned Swiss German word as an adverb is taken on the basis of the frequency distribution of POS tags (see Section 4.3). However, taking into consideration the word itself, some more sophisticated rules can be elaborated. Considering the differences between Standard German and Swiss German as explained in detail in Section 2.1.4, we can expect some words like infinitive particles (`PTKINF`) (e.g. *go*) or the past participle *gsi* (*been*) to remain unaligned. The former because these words do not exist in Standard German, the latter because Standard German simple past tense is expressed by perfect tense in Swiss German, typically resulting in a "spare" past participle in the alignment. Furthermore, there are unaligned articles because Swiss German requires articles in front of proper names. Also punctuation including the apostrophe is a source of errors which can easily be corrected. The application of these more elaborate rules have an impact of around 2% on the evaluation scores, which are discussed in more detail in Section 6.

As the algorithm above transfers the German parses as they are, the numbering of the token IDs is mixed up. Correcting the token IDs to be in ascending order (from 1 to the length of the sentence) requires the corresponding adjustment of the head references. Furthermore, one needs to make sure that there is exactly one root in a sentence.

Data: transferred DE parse to GSW words
Result: valid GSW parse
for *sentence in parse* **do**
  if *DE root was not projected to GSW parse* **then**
    take 1st VERB as root, else 1st NOUN
  else if *head of a projected word was not projected to GSW parse* **then**
    attach it to the root
  end
end



For this step I use the `conll_reader_utils` library[41] in order to manipulate the parses. The algorithm goes through every sentence of the input file and first makes sure that there is one root for the sentence. If the root of the Standard German parse has not been transferred to the Swiss German sentence (due to lacking word alignment), the first verb (*UPOS* `VERB`) is taken as root and if there is no `VERB` in the sentence, the first `NOUN` is considered the root. Furthermore, as the token IDs are changed to be in ascending order (from 1 to the length of the sentence), the head pointer of every word is changed accordingly, so the dependency arrow is still pointing to the same word, even though its token ID changed.

5.4. Optimisation

The parser was trained on the German *Universal Dependency* treebank and then used for Swiss German, which introduces some errors due to the differences between the two languages. In addition, the training set, being transformed to *UD* automatically as described above (see Section 4.1.1) is not free from errors. In this section, I present two approaches for optimisation. One of them was applied to the training set before the training (5.4.1), the other one applied as postprocessing on the parser's output (5.4.2).

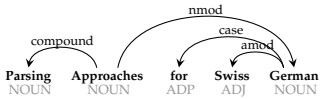
In order to get statistics about the errors and their frequencies, I randomly split the gold standard (see Section 4.2) into a development set of 30 sentences and a test set of 70 sentences. Then I used the development set to analyse the mistakes of the different parsing approaches I tested and came up with rules on the basis of the error analysis. I then applied these rules as postprocessing step to the 70 test sentences.

5.4.1. Preprocessing of the Training Set

One frequent mistake mostly observed in the delexicalised approach is the wrong assignment of passive dependency labels, i.e. `aux:pass` and `nsubj:pass` instead of `aux` and `nsubj`. The passive construction in Standard German is built with the auxiliary *werden*, which can, however, also be used in non-passive constructions. The combination of *werden* and a perfect participle (`VVPP`) is very frequent in Swiss German, however, it is usually not a passive construction but rather a perfect tense. In the training set, 2867 occurrences of *werden* as `AUX` (out of 3003 in total) are labelled as `aux:pass` and only 117 as `aux`. Therefore, a simple but effective solution is the introduction of a new "set" of POS tags in the German *UD* training set: `VWFIN`, `VWINF`, `VWPP` for finite verbs, infinitives and participles respectively of the verb *werden*. This means, all the occurrences of the lemma *werden* as an auxiliary (i.e. *UPOS*: `AUX` and *STTS*: `VA{INF|PP}`) are replaced by `VW{INF|PP}`. This was the case for 2,528 occurrences of *werden* as `VAFIN`, 377 as `VAINF` and 97 as `VAPP`. The other 355 times *werden* occurred, it was used as a full verb (*UPOS*: `VERB`) and therefore not changed; 298 times as `VAFIN`, 33 times as `VAINF` and 24 times as `VAPP`[42]. In this way, the system learned to discriminate between the usage of *werden* as auxiliary versus the usage as full verb and, most of all, it learned to differentiate between the auxiliary *werden* and the other auxiliaries *haben* (*to have*) and *sein* (*to be*). Hence, the number of wrongly assigned passive dependency labels decreased, which leads to an improvement of around 2.5% to 3.5% LAS as presented in Section 6.

[41] https://gitlab.cl.uzh.ch/mamsler/conll_reader_utils

[42] Unlike *UPOS* guidelines, the *STTS* guidelines do not differentiate between the functions of the verbs *haben*, *sein* and *werden*. Instead, they are all tagged as auxiliary (`VAXXX`) independent of their effective function in the sentence.



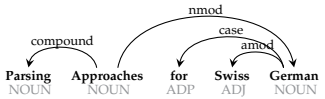
5.4.2. Postprocessing Rules

Some of the errors can easily be corrected with simple rules in a postprocessing step. The parser assigned many `det:poss` instead of `nmod:poss` for possessive nominal modifier. The former label does not exist anymore in *UD version 2*, so these must be some remnants of the 1st version. Another remainder concerns the frequent confusion of the two labels oblique nominal (`obl`) and nominal modifier (`nmod`) because the latter was used to modify nominals and predicates in *UD v1*. However, in *UD v2*, `obl` is used for a nominal functioning as an oblique argument, while `nmod` is used for nominal dependents of another noun (phrase) only. This means, if the head is a verb, adjective or adverb, the dependency label has to be `obl`. If, instead, the head is a noun, pronoun, name or number, the dependency label is `nmod`. Furthermore, the adverbial modifier (`advmod`) becomes an adjectival modifier (`amod`) if the word itself is an adjective (`ADJ`) and its head is a noun (`NOUN`, `PRON`, `PROPN` or `NUM`). Another straightforward rule can be applied if the expletive nominal *es* is parsed as (passive) subject (`nsubj(:pass)`) instead of `expl`.

In contrast to these simple improvements, there are also many confusions which cannot easily be fixed with postprocessing rules. Subject instead of (indirect) object assignments for example, or copula instead of auxiliary and the other way around; these are decisions that require more structural information of the whole sentence, which is the task of the parser itself. Furthermore, most of the mistakes are made only once in our test set and are therefore not worth treating with special postprocessing rules.

5.5. Creation of a Silver Treebank

Using the best scoring parsing approach, direct cross-lingual parsing (see Section 6), I automatically parse 6,155 Swiss German sentences (see Section 4.1.2; excluding gold standard sentences) in order to create a *silver treebank*. A silver standard treebank, as opposed to a gold standard treebank which is assumed to be correctly annotated, is automatically annotated and may therefore be faulty. Then, I use this silver treebank to train a monolingual Swiss German parser (after removing all the gold standard sentences) and hence, create a first monolingual Swiss German dependency parsing model. The advantage of using a silver treebank is the fact that it becomes a monolingual task. However, this comes with the price of a faulty training set, which is not the best resource to build a parser. The evaluation of the monolingual trained parsing model on the gold standard can be found in Section 6.



6. Results & Discussion

This section presents the different settings and combinations of aforementioned resources, approaches and tools. For the evaluation, I used the 100 gold standard sentences described in Section 4.2. I evaluated the approaches according to Labelled Attachment Score (LAS) and Unlabelled Attachment Score (UAS), not excluding punctuation. UAS is the percentage of tokens with the correct syntactic head, LAS the percentage of tokens assigned the correct syntactic head as well as the correct dependency label (see Section 2.2.3). The results I present here are macro accuracy scores, i.e. the scores are computed for each sentence separately and then averaged for all the sentences (as opposed to the word-based micro scores, where the true positives are summed up over the whole treebank and divided by the total number of words in the end). Note that the test set containing 100 gold standard sentences is small and therefore these results have to be taken with a grain of salt.

6.1. German Parser Accuracy

In order to put the results into context, I checked the performance of the parsers on the German *UD v2* treebank (see Section 4.1.1). I used the 13,814 sentences of the training set for training and tested on the test set consisting of 977 sentences. In this setting, I left all the available information for the parser to use, including morphology and lemmas. The `APPRART` splitting is undone for the `CoNLL-X` *MaltParser* input, not so for the *UDPipe* which takes `CoNLL-U` as input format (and performs worse with the *MaltParser*-`CoNLL-X` input).

Parser	LAS	UAS
MaltParser	79.71	83.19
UDPipe	70.31	77.06

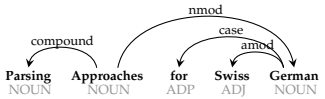
Table 6.1: Parser accuracy for the German *UD* treebank training and test set.

In the *CoNLL 2017* shared task (Zeman et al., 2017), the official *UDPipe* baseline for German has 73.11% LAS and 78.21% UAS[43]. Using the standard settings of *UDPipe* and *UD v2*, I could not reproduce these numbers.

6.2. Direct Cross-lingual Parsing

As a comparison to my main approaches, I applied Standard German parsers directly to Swiss German. This means, I used the training set of the German *UD* treebank to train the *MaltParser* (using *MaltOptimizer* to get the best hyperparameter settings) and *UDPipe*. Before training, I removed the morphology and lemma information because this information is not available in the Swiss German test set and therefore the parsers cannot rely on it. Furthermore, for the *MaltParser* I converted the training set from `CoNLL-U` to `CoNLL-X` format because, *MaltOptimizer* cannot handle the former. This conversion also undoes the splitting of `APPRART` (see Section 4.1.1).

[43]http://universaldependencies.org/conll17/baseline.html



Parser	Tagger	LAS	UAS
MaltParser	Gold Standard Tags	65.37	76.33
	Wapiti	55.28	69.51
	Wapiti + Preprocessing	**57.79**	70.59
UDPipe	UDPipe	21.19	35.28

Table 6.2: Evaluation of the direct cross-lingual parsing approach.

Testing the model on the gold standard with automatically assigned POS tags by *Wapiti* results in an LAS of 55.28%. The test setting with gold standard tags shows the upper bound of 65.37% LAS. The performance of the *UDPipe* is very low in this setting, not even reaching half the *MaltParser*'s accuracy. One reason for this low accuracy could be that *UDPipe* relies on word embedding information (Straka and Straková, 2017), which results in a low recall when applying a model trained on German to Swiss German. With the preprocessed training set, i.e. differentiating the auxiliary *werden* vs. the auxiliaries *haben* (*to have*) and *sein* (*to have*) (see Section 5.4.1), *MaltParser* improves by 2.5 percentage points to 57.79% LAS.

6.3. Delexicalised Model Transfer

Instead of giving the parser the Standard German words as input like in the direct cross-lingual approach, in the delexicalised approach I provide the parser with part of speech information only. This means, the words are replaced by *STTS* POS tags while all the other columns stay the same.

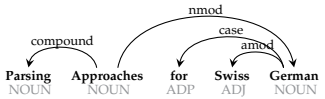
Parser	Tagger	LAS	UAS
MaltParser	Gold Standard Tags	63.84	79.92
	Wapiti	53.69	69.58
	Wapiti + Preprocessing	57.29	70.66
UDPipe	UDPipe	55.77	70.25
	UDPipe + Preprocessing	**57.78**	72.26

Table 6.3: Evaluation of the delexicalised parsing approach.

Given the small evaluation set and the negligible difference in the results, the two parsers' performance is the same: ~57% LAS for both when trained on the preprocessed training set, i.e. differentiating the auxiliary *werden* vs. the auxiliaries *haben* (*to have*) and *sein* (*to be*) (see Section 5.4.1). The preprocessing step brings an LAS increase of 2 percentage points in the case of *UDPipe* and 3.6 percentage points for *MaltParser* respectively.

6.4. Annotation Projection

The results shown in Tables 6.4 are reached with the baseline transfer rules where unaligned words are simply attached to the root as adverbs.



Parser	Aligner	LAS	UAS
MaltParser	GIZA++	49.17	59.39
	FastAlign	46.45	56.82
	MGA	51.37	62.74

Parser	Aligner	LAS	UAS
UDPipe	GIZA++	51.57	62.57
	FastAlign	48.15	58.89
	MGA	53.53	65.44
	MGA + better transfer rules	**55.65**	**66.54**

Table 6.4: Evaluation of the annotation projection parsing approach.

The combination of *UDPipe* and *Monolingual Greedy Aligner* reaches the best result in the annotation projection approach with 53.39% LAS. Applying more elaborate transfer rules as explained in Section 5.3.2 results in an improvement of 2.09 percentage points to 55.65% LAS and also an increase of the *UPOS* accuracy of 72.30% to 75.35%. The preprocessing step does not improve the results in this approach.

These results show that the *Monolingual Greedy Aligner* performs best in the task of Standard German - Swiss German alignment. As described in Section 5.3.1, *MGA* takes character-based word similarity into account. Intuitively, it makes sense that the information about similar letters is valuable information when dealing with closely related languages such as Standard German and Swiss German and therefore it is worthwhile not to ignore character-based similarity as the other two tested aligners *GIZA++* and *FastAlign* do.

6.5. Postprocessing

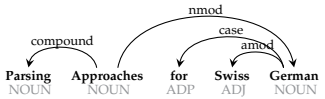
For further improvement, I apply the postprocessing rules (see Section 5.4.2) to the best scoring setting of the results described above.

Table 6.5 shows the ten most frequent confusions of the direct cross-lingual approach before and after applying the postprocessing rules to the output of the direct cross-lingual approach with *MaltParser*. The confusion matrices for the delexicalised model transfer approach with *UDPipe* looks similar and are therefore omitted here. As in Section 5.4.2 explained, the postprocessing rules treat `nmod/obl`, `det:poss/nmod:poss`, `aux:pass/aux`, and `advmod/amod` confusions.

Before Postprocessing			
System	**Correct**	**Count**	**Frequency**
nmod	**obl**	**23**	1.59
nsubj	obj	16	1.11
det:poss	**nmod:poss**	**11**	0.76
obl	**nmod**	10	0.69
aux	cop	10	0.69
cc	advmod	9	0.62
aux:pass	**aux**	**9**	0.62
advmod	**amod**	**8**	0.55
root	advmod	8	0.55
case	mark	7	0.48

After Postprocessing			
System	**Correct**	**Count**	**Frequency**
nmod	**obl**	**16**	1.11
nsubj	obj	15	1.04
obl	**nmod**	**13**	0.9
aux	cop	11	0.76
cc	advmod	9	0.62
advmod	**amod**	**8**	0.55
root	advmod	8	0.55
case	mark	7	0.48
obl	nsubj	6	0.42
nsubj:pass	nsubj	6	0.42

Table 6.5: Confusion tables showing counts and frequencies for the ten most frequent confusions before and after applying postprocessing rules to the direct cross-lingual approach.



Comparing the after postprocessing confusions to the ones before, we can observe that the `det:poss/nmod:poss`, `aux:pass/aux` do not appear anymore, as expected. Contrary to expectations however, the `nmod/obl` confusions are still present. The reason is that the parser assigned wrong heads to many of those words and therefore the rule to correct the `nmod/obl` confusions does not work.

Nevertheless, the postprocessing rules have some impact on the results, as Table 6.6 shows.

		Standard		Postprocessing	
Parser	**Approach**	**LAS**	**UAS**	**LAS**	**UAS**
UDPipe	Annotation Projection	55.65	66.44	57.73	66.57
MaltParser	Cross-lingual	57.79	70.59	59.41	70.59
UDPipe	Model Transfer	57.78	72.26	**59.85**	72.26

Table 6.6: Evaluation of the postprocessing step.

The LAS scores improve by 1.62 percentage points for the cross-lingual *MaltParser* and 2.07 percentage points for delexicalised model transfer and annotation projection *UDPipe* approachs respectively, reaching nearly 60% LAS accuracy.

The corrections added in the more elaborate transfer rules of annotation projection (see Section 5.3.2) can also be applied to the other approaches as additional postprocessing rules (pprules2) resulting in some 1 percentage point further improvement in the model transfer approach, as Table 6.7 shows.

Approach	**Setting**	**LAS**	**UAS**
Direct Cross-lingual	MaltParser (+ Wapiti) + Pre- and postprocessing + pprules2	59.78	70.80
Model Transfer	UDPipe + Pre- and postprocessing + pprules2	**60.64**	**72.48**

Table 6.7: Evaluation scores after applying the more elaborate transfer rules of the annotation projection approach to the other approaches.

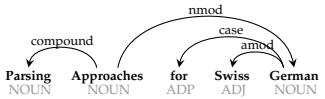
6.6. Discussion

To sum up, Table 6.8 shows the best results including the corresponding setting for every approach. The abbreviation `pprules2` as used in the previous Section 6.5 stands for more the elaborate transfer rules as explained in Section 5.3.2.

Approach	**Setting**	**LAS**	**UAS**
Annotation Projection	UDPipe + MGA + pprules2 + Postprocessing	57.73	66.57
Model Transfer	UDPipe + Pre- and postprocessing + pprules2	**60.64**	**72.48**
Direct Cross-lingual	MaltParser (+ Wapiti) + Pre- and postprocessing + pprules2	59.78	70.80

Table 6.8: Comparison of the best score of every approach.

As Table 6.8 illustrates, the best LAS results of all the applied approaches are very close, hence there is no clear answer to the question of which approach works best. Annotation projection is the most laborious among the three and as such not the first option to choose. Furthermore, the transfer of the annotation is strongly dependent on the performance of the aligner, which in turn benefits from big



parallel corpora to be trained on. However, such big parallel corpora do not exist yet for Swiss German dialects.

6.6.1. St. Gallen vs. Bern

As detailed in Section 4.1.2, besides 50 sentences taken from the *NOAH* corpus, the test set contains 25 Bern dialect sentences and 25 St. Gallen dialect sentences. These two dialects feature some differences (see Section 2.1) of which the word ordering is especially interesting for the parsing task. Table 6.9 shows a comparison of the LAS scores reached by the model transfer and the annotation projection approach for the two dialects as well as for *NOAH* sentences and the overall scores for comparison. The scores are based on the parsing output before the postprocessing rules were applied.

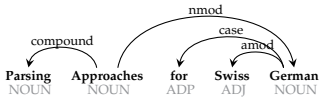
	Model Transfer	**Annotation Projection**
St. Gallen	48.20	44.48
Bern	59.12	49.76
NOAH	61.92	64.16
Overall	57.78	55.65

Table 6.9: LAS scores by dialect for the parsing approaches model transfer and annotation projection.

The LAS scores for the dialects St. Gallen and Bern in Table 6.9 are based on 25 sentences only, therefore these numbers have to be taken with a grain of salt. The model transfer approach reaches better results for both dialects but the difference for the Bern dialect is twice as big as the difference for the St. Gallen dialect. Interestingly, both approaches achieve better results on the Bern dialect. This is surprising because the word ordering for the St. Gallen dialect is closer to the Standard German word ordering while Bernese dialect speakers often change the order of the verbs (see Section 2.1). Due to these differences, I expected the model transfer approach to perform worse on the Bern dialect than annotation projection, where the word order changes should be handled by the aligner. Looking specifically at Bernese sentences, with "switched" word order (e.g. *ha aafo gränne* (*I started to cry*), *gfunge hei gha* (*have found*), *het übercho* (*have gotten*)), there is no significant difference between the two approaches.

6.6.2. Model Transfer vs. Annotation Projection

There are sentences where one approach works very well while the other fails, illustrated by the examples in Figures 6.1 and 6.2.



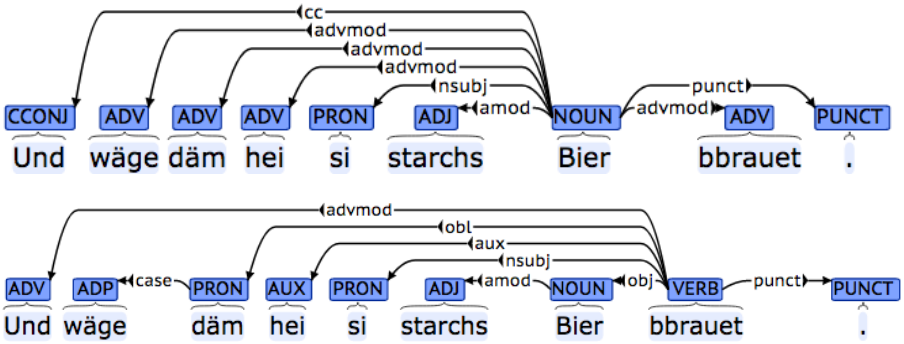


Figure 6.1: *And that's why they brewed strong beer.* Top: annotation projection (LAS 11%), bottom: model transfer (LAS 100%).

Figure 6.1 shows a sentence which was correctly parsed by the model transfer approach. The annotation projection approach failed in the alignment; the words *wäge* (*because*), *däm* (*this*), *hei* (*have*) and crucially, the verb *bbrauet* (*brewed*) were not aligned. The simple transfer method (see Section 5.3.2) attached them as adverbs to the root. As the root was not transferred from the German parse, the first noun is taken as a root, because there is no verb in the sentence either. This example illustrates the importance of the word alignment for this approach. In such situations, the POS information would help, as it would prevent every unaligned word from being treated as adverb.

Figure 6.2 shows an example where the annotation projection approach worked much better than the model transfer approach.

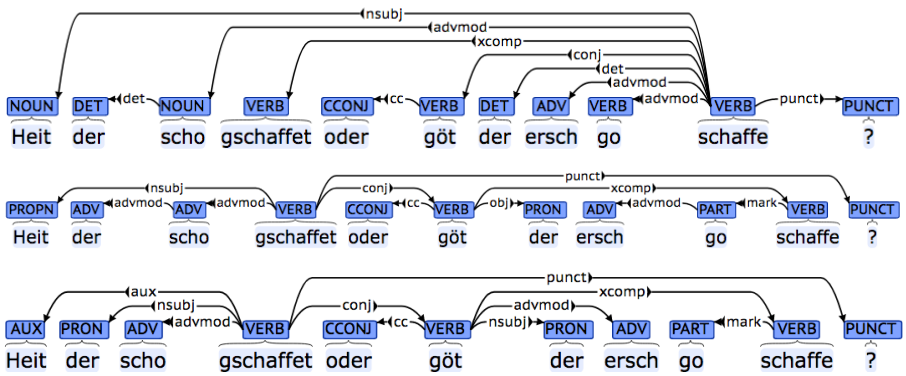
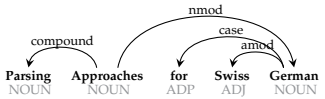


Figure 6.2: *Have you already been working or are you going to work later?* Top: model transfer (LAS 9%), middle: annotation projection (LAS 64%), bottom: gold standard.

In the example in Figure 6.2, both approaches did not parse the sentence correctly. However, while the annotation projection approach scores 64% LAS, model transfer fails with an LAS of 9%. The model transfer approach seems to be confused by the verbs as well as the *Heit* (*to have* in the second person plural used as polite form) at the beginning of a sentence, which is POS tagged as a noun and therefore considered the subject of the sentence. Annotation projection works well here, because all the words were aligned to their German translation and the word *go* was correctly integrated by the transfer rule (see Section 5.3.2).



6.6.3. POS Tagging Evaluation

As described in Section 4.2, parts of the test set were used in the training for the *Wapiti* tagger as half of the gold standard is taken from the *NOAH* corpus. This affects the results of the delexicalised and cross-lingual approaches where *MaltParser* and therefore *Wapiti* tagger is used, but not the *UDPipe* results. Hence, in Table 6.8, this concerns the direct cross-lingual approach. Table 6.10 shows the POS tagging accuracy for the three parsing settings in Table 6.8. Note that in difference to the parsing results, here I present micro accuracies (which deviate less than 0.5 percentage points from their macro counterpart).

Approach	**POS Accuracy**		
	NOAH	**Rest**	**Overall**
Annotation Projection	82.62	71.19	76.39
Delexicalised	94.05	85.03	89.13
Direct Cross-lingual	93.14	83.38	87.67

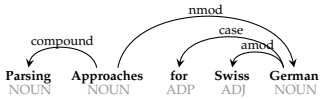
Table 6.10: POS tagging accuracy of the best scoring parsing approaches.

The evaluation of the POS tagging shows a difference in performance between the sentences taken from the *NOAH* corpus and the others: the POS tagging accuracy for the *NOAH* sentences is 93.14% while the POS tagging accuracy for the other sentences is only 83.38%. This seems to favour the delexicalised approach. However, evaluating the POS tagging accuracy for the winning delexicalised setting (*UDPipe*), the numbers show the same bias towards the *NOAH* sentences, which score 94.05% accuracy while the rest is only tagged with an accuracy of 85.03%. Hence, the *NOAH* sentences might be easier to tag than the others. Reasons for that could be the dialects; while *NOAH* includes different dialects (of which a big part is Zurich dialect), the rest only contains two dialects (St. Gallen and Bern). Furthermore, the text genre might have an influence, as the *Rest* contains only literary text with a tendency towards longer and more complex sentences. The average length of *NOAH* sentences is 13.12 while the average length of the other sentences is 15.76, i.e. 20% longer than an average *NOAH* sentence.

6.6.4. Swiss German Variability

The results presented here are not perfect and certainly require further improvement in order for a system to be used in real-life applications. Compared with the German parser accuracy for example, which reached almost 80% LAS on the German *UD v2* with standard settings of the parsers (see Section 6.1), there is room for improvement. However, these numbers have to be set in relation to the data I worked with. Even though I could make use of Swiss German books and crowdsourced data, it is still a small data set. Furthermore, the enormous variability in Swiss German dialect (writing) poses a serious challenge for all tools. Statistical tools work best if the observed events are not rare. However, they do not work with sparse data consisting of a vast amount of hapax legomena, i.e. word form which appears only once[44] (see Figure 6.3 for descriptive statistics).

[44]https://www.merriam-webster.com/dictionary/hapax%20legomenon



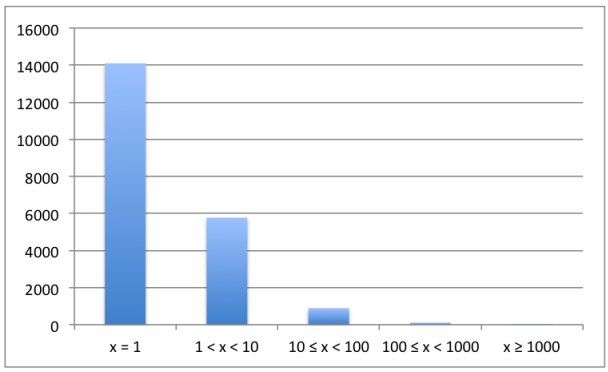


Figure 6.3: Frequencies of type frequencies (x) in a Swiss German text.

Figure 6.3 shows the frequencies (on y-axis) of type frequencies (x) in a Swiss German text. 6,155 Swiss German sentences which are made up of 105,692 tokens (tokenised by *cutter*) contain 20,882 types (different tokens). 14,099 types appear only once (i.e. hapax legomena), 19,874 less than 10 times and 20,767 less than 100 times.

The work presented here provides insights how well classical cross-lingual parsing approaches like annotation projection and model transfer work for the language pair Standard German – Swiss German.

6.7. Silver Treebank Parsing Model

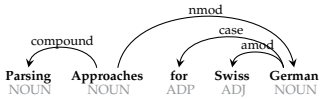
For training a first monolingual parser I used the *MaltParser* in the direct cross-lingual approach including preprocessing, which reached an accuracy of 57.79%. As Section 5.5 explains, I used this model to parse ~6000 Swiss German sentences, which results in a larger Swiss German *Silver Treebank*. These parses can in turn be used as input for training a first monolingual parsing model. I first expected that using an automatically parsed training set with an accuracy of ~57% LAS cannot possibly result in good LAS scores. However, the universal dependency parser trained on the automatically parsed Swiss German silver treebank does not perform as bad as expected.

Training Data	**LAS**	**UAS**
Silver treebank	57.10	68.88
Silver treebank + DE UD	55.46	67.13

Table 6.11: Evaluation of the parser performance trained on a Swiss German silver treebank.

The performance of a parser trained only on the silver treebank reaches the same result as the direct cross-lingual parsing approach itself, which was used to generate the silver treebank. Given that 6,000 sentences does not constitute a large training set for a statistical parser, a parser could maybe profit from additional related Standard German material. However, combining the two training sets, i.e. the German *Universal Dependency* treebank and the silver treebank, the parser performs slightly worse.

The silver treebank simplifies and reduces the manual annotation work as compared to annotating all the sentences from scratch, i.e. manually correcting the silver treebank could be the next step in order to get a substantial amount of monolingual training data for a statistical parser. Also, silver treebank could be a bit improved by only choosing sentences where all approaches reach a certain LAS and discard the others.



6.8. Future Work

For further improvement of the annotation projection approach, the crucial alignment information needs to be improved. In order to do so, the outputs of different aligners can be ensembled. This can be done via majority vote for instance, i.e. only choosing or relying upon information which was produced by several approaches. Thereby a word would only be aligned if several aligners have produced the same output.

Concerning all parsing approaches, more elaborate postprocessing rules as well as transfer rules in the annotation projection approach could be added. Especially for the latter case where the alignment did not work, rules could be added such that the unaligned words do not all become adverbs. Furthermore, decisions concerning the cases where the root was not transferred because of missing alignment could be improved.

Furthermore, given the enormous variability and the vast amount of hapax legomena, another source of improvement could be normalisation of the writing. Samardžić et al. (2015) have worked on normalisation for Swiss German for the *ArchiMob* and the morphological analysis tool of Baumgartner (2016) includes lemmatisation. These tools could be exploited in order to be able to map different spelling variants to one version of the word.

Also, the indicator + on the POS tags for expressing incorporated elements can be transferred not only to the *UPOS* tag but also to the dependency relation in order to indicate that there are incorporated words. Depending on the end-application of the parses, this might be interesting information as for example for a content analysis. Figure 6.4 shows a case of an incorporated subject as the *i* merged to the auxiliary *to be*; *bini* (*STTS*: `VAFIN+`), which is why there is no `nsubj` dependency.

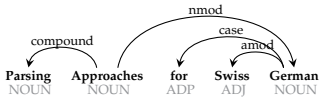
1	Aso	_	ADV	ADV	_	4	advmod	_	_
2	bini	_	AUX	VAFIN+	_	5	aux	_	_
3	rächt	_	ADV	ADV	_	4	advmod	_	_
4	uufgschmissä	_	ADJ	ADJD	_	0	root	_	_
5	gsi	_	AUX	VAPP	_	4	cop	_	_
6	und	_	CCONJ	KON	_	10	cc	_	_
7	dem	_	PRON	PDS	_	8	iobj	_	_
8	entschprächend	_	ADJ	ADJD	_	10	advmod	_	_
9	fascht	_	ADV	ADV	_	10	advmod	_	_
10	verzwiiflät	_	VERB	VVPP	_	4	conj	_	_
11	.	_	PUNCT	$.	_	4	punct	_	_

Figure 6.4: *So I was pretty much in a fix and therefore almost despaired.*

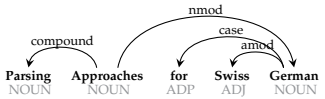
Transferring this information from the POS tag to the parse, a parser could learn more differentiated parse structures. However, the class of the dependency labels from which the parser chooses becomes larger, which might also decrease the performance.

Furthermore, the outputs of the three parsing approaches could be joined, for example via majority vote like for alignment as aforementioned, to get rid of the weaknesses of each approach. In cases where the word alignment of the annotation projection approach did not work for example, one could favour the parse of a different approach.

Once the data sparseness for Swiss German varieties is overcome, deep learning strategies are promising as shown for example in the work by Ammar et al. (2016). Ammar et al.'s approach is to train one multilingual model that can be used to parse sentences in several languages. In order to do so, they



use many resources including a bilingual dictionary for adding cross-lingual lexical information, and a monolingual corpus. The latter is used for training word embeddings which their Malopa parser makes use of. Such kind of approaches need a big amount of data of the language to be parsed, which is still a problem for Swiss German.



7. Conclusion

In this thesis, I experimented with a variety of approaches for parsing texts written in Swiss German. For statistically driven systems, languages with non-standardised orthography are a demanding task. Swiss German dialects feature challenging Natural Language Processing (NLP) problems with their lack of orthographic spelling rules and a huge pronunciation variety. This is a situation which leads to a high degree of data sparseness and with it, a lack of resources and tools for NLP.

Dependency parsing is a crucial step needed for numerous NLP applications like information extraction, sentiment analysis or question answering, to name but a few. In order to build a statistical parser, resources are needed in form of a treebank, i.e. a syntactically annotated text corpus.

In the case of low-resourced languages for which a resource-rich language is available, cross-lingual methods can be exploited. In my thesis, I apply this paradigm by using Standard German resources and applying them to Swiss German dialects in order to deal with NLP challenges for Swiss German.

I have tested a lexicalised annotation projection method as well as a delexicalised model transfer method. The annotation projection method requires parallel sentences in both the resource-rich and the low-resourced language. Then, the resource-rich language's parser is used to parse the sentences and the resulting syntactic analyses are transferred to the low-resourced language via word alignment. The delexicalised model transfer approach does not require a parallel corpus but a monolingual treebank of a closely related resource-rich language. A parser is then trained on the basis of part of speech tags only, without any information about words. The sentences of the resource-poor language are also delexicalised before parsing, such that a parse is built on the part-of-speech information. After parsing, the sentences are re-lexicalised resulting in a normal parse tree.

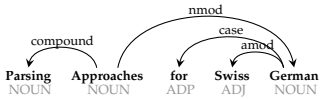
To obtain a parallel corpus in order to apply the annotation projection method, I used the *Agora* project platform to crowdsource Standard German translations for Swiss German sentences. After filtering and cleaning the crowdsourced data, I got 26,015 sentences as a parallel GSW/DE corpus, containing several Standard German translations of each of the 6,197 Swiss German sentences taken from the *NOAH* corpus and from two books by Pedro Lenz and Renato Kaiser.

The evaluation on a manually parsed gold standard consisting of 100 sentences shows a 60% Labelled Attachment Score (LAS) with negligible differences between the different parsing approaches. However, the annotation projection approach is more complex than a model transfer due to the transfer rules which have to be specified, and to the crucial word alignment process. Interestingly, the evaluation showed big differences in the LAS scores specific to the dialects: Unexpectedly, both approaches performed better for the Bern dialect, even though it differs more from Standard German than the St. Gallen dialect, for which the LAS scores were lower.

There are several opportunities for further improvements like joining information from different word aligners or adding further transfer and postprocessing rules. In addition, a spelling normalisation strategy could help to deal with the vast amount of hapax legomena due to the linguistic variability in Swiss German dialects. Furthermore, the *silver treebank* created could be manually corrected in order to generate a treebank which can be used as training set for a monolingual dependency parser for Swiss German.

This thesis provides a first substantial step towards closing a big gap in Natural Language Processing tools for Swiss German and provides data[45] to work on further improvements.

[45]https://github.com/noe-eva/SwissGermanUD



References

Aepli, N. and Allemann, A. (2016). Schwiizer{d | t}ütschi Vokä{u | l} – west vs. ost. Seminar Thesis.

Aepli, N., Samardžić, T., and von Waldenfels, R. (2014). Part-of-Speech Tag Disambiguation by Cross-Linguistic Majority Vote. In *First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects (VarDial)*, Dublin.

Ammar, W., Mulcaire, G., Ballesteros, M., Dyer, C., and Smith, N. A. (2016). Many Languages, One Parser. *Transactions of the Association for Computational Linguistics (ACL)*, 4:431–444.

Ballesteros, M. and Nivre, J. (2012). MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 58–62, Stroudsburg, PA, USA.

Baumgartner, R. (2016). Morphological analysis and lemmatization for Swiss German using weighted transducers. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, Bochum, Germany.

Buchholz, S. and Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of Conference on Computational Natural Language Learning (CoNLL 2006)*, pages 149–164.

Chiang, D., Diab, M. T., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing Arabic Dialects. In *Proceedings of European Chapter of the Association for Computational Linguistics (EACL)*, pages 369–376, Trento.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*.

de Marneffe, M.-C. and Manning, C. D. (2008). The Stanford Typed Dependencies Representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

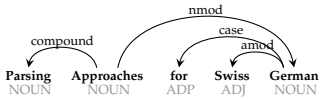
Dürscheid, C. and Stark, E. (2011). SMS4science: An International Corpus-based Texting Project and the Specific Challenges for Multilingual Switzerland. *Digital Discourse: Language in the New Media*, pages 299–320.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *HLT-NAACL*, pages 644–648. The Association for Computational Linguistics (ACL).

Eisner, J. (1996). Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 96)*, pages 340–345.

Foth, K. A. (2004). Eine umfassende Constraint-Dependenz-Grammatik des Deutschen. *http://nbn-resolving.de/urn:nbn:de:gbv:18-228-7-2048*.

Gesmundo, A. and Samardžić, T. (2012). Lemmatisation as a Tagging Task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–372.



Glaser, E. (2003). Schweizerdeutsche Syntax: Phänomene und Entwicklungen. In Dittli, B., Buhofe, A. H., and Haas, W., editors, *Gömmer MiGro?*, pages 39–66, Freiburg, Schweiz.

Hollenstein, N. and Aepli, N. (2014). Compilation of a Swiss German Dialect Corpus and its Application to PoS Tagging. *International Conference on Computational Linguistics (COLING) 2014*, page 85.

Hollenstein, N. and Aepli, N. (2015). A Resource for Natural Language Processing of Swiss German Dialects. *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL) 2015*.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*, 11(03):311–325.

Hwa, R., Resnik, P., Weinberg, A., and Kolak, O. (2002). Evaluating Translational Correspondence Using Annotation Projection. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 392–399, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2017). Dependency Parsing. *https://web.stanford.edu/~jurafsky/slp3/14.pdf*.

Kaiser, R. (2012). *UUFPASSÄ, NÖD AAPASSÄ!* Der gesunde Menschenversand.

Klaper, D. (2014). 11-712: NLP Lab Report: A Dependency Parser for Swiss German. Seminar Thesis.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical Very Large Scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513.

Lenz, P. (2013). *I bi meh aus eine*. Cosmos Verlag AG.

Levenshtein, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Loos, E., Anderson, S., Dwight, D., Jordan, P., and Wingate, D. (2004). Glossary of Linguistic Terms. *http://www-01.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsACliticGrammar.htm*.

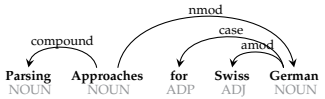
Luftseilbahn Jakobsbad-Kronberg AG (2016). Geschäftsbericht 2016. *http://www.kronberg.ch/de/aktionaere.html*.

Mcdonald, R. (2007). Characterizing the Errors of Data-driven Dependency Parsing Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*, pages 122–131, Prague.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., and Lee, J. (2013). Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 92–97.

Mcdonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective Dependency Parsing Using Spanning Tree Algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530.

Naseem, T., Chen, H., Barzilay, R., and Johnson, M. (2010). Using Universal Linguistic Knowledge to Guide Grammar Induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244.



Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007a). The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL) Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic. Association for Computational Linguistics (ACL).

Nivre, J., Hall, J., and Nilsson, J. (2004). Memory-Based Dependency Parsing. In Ng, H. T. and Riloff, E., editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL 2004)*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics (ACL).

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007b). MaltParser: A Language-independent System for Data-driven Dependency Parsing. *Natural Language Engineering*, 13(2):95–135.

Nivre, J. and Nilsson, J. (2005). Pseudo-projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 99–106, Stroudsburg, PA, USA.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Petrov, S., Das, D., and McDonald, R. (2012). A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2089–2096.

Ringier AG (2013). Blick am Abig. *http://epaper.blick.ch/webreader/baa/download/?doc=BAA280513ZH*.

Rosa, R., Dušek, O., Mareček, D., and Popel, M. (2012). Using Parallel Features in Parsing of Machine-Translated Sentences for Correction of Grammatical Errors. In *Proceedings of SSST-6, Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 39–48.

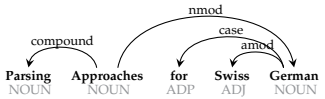
Rosa, R., Zeman, D., Mareček, D., and Žabokrtský, Z. (2017). Slavic Forest, Norwegian Wood. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 210–219, Valencia, Spain. Association for Computational Linguistics.

Samardžić, T., Scherrer, Y., and Glaser, E. (2015). Normalising orthographic and dialectal variants for the automatic processing of Swiss German. In *Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 294–298, Poznan, Poland.

Samardzic, T., Scherrer, Y., and Glaser, E. (2016). ArchiMob - A Corpus of Spoken Swiss German. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Scherrer, Y. (2007). Adaptive String Distance Measures for Bilingual Dialect Lexicon Induction. In *Proceedings of the ACL 2007 Student Research Workshop*. The Association for Computer Linguistics.

Scherrer, Y. (2012). Machine Translation into Multiple Dialects: The Example of Swiss German. *7th SIDG Congress - Dialect 2.0*.



Scherrer, Y. (2013). Continuous Variation in Computational Morphology - The Example of Swiss German. In *TheoreticAl and Computational MOrphology: New Trends and Synergies (TACMO)*, Genève, Suisse. 19th International Congress of Linguists.

Scherrer, Y. and Owen, R. (2010). Natural Language Processing for the Swiss German Dialect Area. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 93–102, Saarbrücken, Germany.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textkorpora mit STTS. *http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf*.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Schobinger, V. (2013). Viktor's züritü(ü)tsch. *http://www.zuerituetsch.ch/index.html*.

Schobinger, V. (2014). *Der Ääschmen und de Schtùùrzmord*. Schobinger-Verlaag.

Shieber, S. M. (1985). Evidence Against the Context-freeness of Natural Language. *Linguistics and Philosophy*, 8:333–343.

Siebenhaar, B. and Voegeli, W. (1997). 6 Mundart und Hochdeutsch im Vergleich. *http://home.uni-leipzig.de/siebenh/pdf/Siebenhaar_Voegeli_iPr.pdf*.

Siebenhaar, B. and Wyler, A. (1997). Dialekt und Hochsprache in der deutschsprachigen Schweiz. *http://home.uni-leipzig.de/siebenh/pdf/Siebenhaar_Wyler_97.pdf*.

Snyder, B., Naseem, T., Eisenstein, J., and Barzilay, R. (2008). Unsupervised Multilingual Learning for POS Tagging. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1041–1050, Honolulu, Hawaii. Association for Computational Linguistics (ACL).

Stark, E., Ueberwasser, S., and Göhrig, A. (2014–2018). Corpus "What's up, Switzerland?". *www.whatsup-switzerland.ch*.

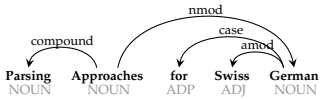
Straka, M., Hajič, J., Straková, J., and Hajič jr., J. (2015). Parsing Universal Dependency Treebanks using Neural Networks and Search-Based Oracle. In *Proceedings of Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14)*.

Straka, M. and Straková, J. (2017). Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL) 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics (ACL).

Stukenberg, T. (2015). Feindlich simma! Aber deppert? *http://www.spiegel.de/wissenschaft/mensch/image-von-dialekten-mia-san-nett-aber-deppert-a-1030038.html*.

Täckström, O., McDonald, R., and Nivre, J. (2013). Target Language Adaptation of Discriminative Transfer Parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.

The Swatch Group AG (2012). Swatch Group Geschäftsbericht 2012. *http://www.swatchgroup.com/de/investor_relations/jahres_und_halbjahresberichte/fruehere_jahres_und_halbjahresberichte*.



Tiedemann, J. (2014). Rediscovering Annotation Projection for Cross-Lingual Parser Induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1854–1864.

Tiedemann, J. (2015). Cross-Lingual Dependency Parsing with Universal Dependencies and Predicted PoS Labels. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 340–349.

Tiedemann, J., Agic, Z., and Nivre, J. (2014). Treebank Translation for Cross-Lingual Parser Induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, pages 130–140.

TIGER Project (2003). TIGER Annotationsschema. *https://www.linguistics.ruhr-uni-bochum.de/~dipper/pub/tiger_annot.pdf*.

von Ahn, L. (2006). Games with a Purpose. *Computer*, 39(6):92–94.

Wang, A., Hoang, C. D. V., and Kan, M.-Y. (2013). Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation (LREC 2013)*, 47(1):9–31.

Wikipedia, The Free Encyclopedia (2011). Alemannic Wikipedia. *http://als.wikipedia.org/wiki/Wikipedia:Houptsyte*.

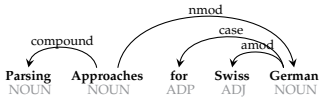
Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8.

Yimam, S. M., Gurevych, I., Eckart de Castilho, R., and Biemann, C. (2013). WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria.

Zampieri, M., Malmasi, S., Ljubešić, N., Nakov, P., Ali, A., Tiedemann, J., Scherrer, Y., and Aepli, N. (2017). Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.

Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinkova, S., Hajic jr., J., Hlavacova, J., Kettnerová, V., Uresova, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C. D., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.-C., Sanguinetti, M., Simi, M., Kanayama, H., dePaiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, c., Sulubacak, U., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Alcalde, H. F., Strnadová, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonca, G., Lando, T., Nitisaroj, R., and Li, J. (2017). CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL) 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics (ACL).

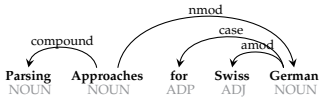
Zeman, D. and Resnik, P. (2008). Cross-language Parser Adaptation Between Related Languages. *NLP for Less Privileged Languages*, pages 35 – 35.



Appendices

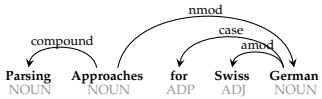
A. *Stuttgart-Tübingen-TagSet* (*STTS*) Part-of-Speech Tagset

ADJA	attributive adjective
ADJD	predicate adjective; adjective used adverbially
ADV	adverb
APPR	preposition left hand part of double preposition
APPRART	preposition with fused article
APPO	postposition
APZR	right hand part of double preposition
ART	article
CARD	cardinal number; also declined
FM	foreign words
ITJ	interjection
KON	co-ordinating conjunction
KOKOM	comparative conjunction or particle
KOUI	preposition used to introduce infinitive clause
KOUS	subordinating conjunction
NE	names and other proper nouns
NN	noun
PAV	pronominal adverb
PDAT	demonstrative determiner
PDS	demonstrative pronoun
PIAT	indefinite determiner
PIS	indefinite pronoun
PPER	personal pronoun
PRF	reflexive pronoun
PPOSS	possessive pronoun
PPOSAT	possessive determiner
PRELAT	relative depending on a noun
PRELS	relative pronoun
PTKA	particle with adjective or adverb
PTKANT	answer particle
PTKNEG	negative particle
PTKVZ	separable prefix
PTKZU	infinitive particle
PWS	interrogative pronoun
PWAT	interrogative determiner
PWAV	interrogative adverb
TRUNC	truncated form of compound
VAFIN	finite auxiliary verb
VAIMP	imperative of auxiliary
VAINF	infinitive of auxiliary
VAPP	past participle of auxiliary
VMFIN	finite modal verb
VMINF	infinitive of modal
VMPP	past participle of auxiliary
VVFIN	finite full verb
VVIMP	imperative of full verb
VVINF	infinitive of full verb
VVIZU	infinitive with incorporated "zu"
VVPP	past participle of full verb
XY	non-word, contains symbol
$,	comma
$.	sentence-final punctuation
$(	other punctuation, sentence internal
PTKINF	infinitive particle
TAG+	merged words



B. Mapping STTS to Universal Part-of-Speech Tagset

ADJ	adjective	ADJA, ADJD
ADP	adposition	APPO, APPR, APPRART, APZR
ADV	adverb	ADV, PAV, PWAV
AUX	auxiliary	VAFIN, VAIMP, VAINF, VAPP
CCONJ	coordinating conjunction	KOKOM, KON
DET	determiner	ART; PDAT, PIAT, PIDAT, PPOSAT, PRELAT, PWAT
INTJ	interjection	ITJ
NOUN	noun	NN
NUM	numeral	CARD
PART	particle	PTKA, PTKANT, PTKNEG, PTKVZ, PTKZU, PTKINF
PRON	pronoun	PDS, PIS, PPER, PPOSS, PRELS, PRF, PWS
PROPN	proper noun	NE
PUNCT	punctuation	$(, $., $,
SCONJ	subordinating conjunction	KOUI, KOUS
SYM	symbol	–
VERB	verb	VMFIN, VMINF, VMPP VVFIN, VVIMP, VVINF, VVIZU, VVPP
X	other	FM, TRUNC, XY



C. Universal Dependency Relations

acl	clausal modifier of noun (adjectival clause)
advcl	adverbial clause modifier
advmod	adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
case	case marking
cc	coordinating conjunction
ccomp	clausal complement
clf	classifier
compound	compound
conj	conjunct
cop	copula
csubj	clausal subject
dep	unspecified dependency
det	determiner
discourse	discourse element
dislocated	dislocated elements
expl	expletive
fixed	fixed multiword expression
flat	flat multiword expression
goeswith	goes with
iobj	indirect object
list	list
mark	marker
nmod	nominal modifier
nsubj	nominal subject
nummod	numeric modifier
obj	object
obl	oblique nominal
orphan	orphan
parataxis	parataxis
punct	punctuation
reparandum	overridden disfluency
root	root
vocative	vocative
xcomp	open clausal complement

Universität Zürich[UZH]

Philosophische Fakultät
Studiendekanat

Universität Zürich
Philosophische Fakultät
Studiendekanat
Rämistrasse 69
CH-8001 Zürich
www.phil.uzh.ch

Selbstständigkeitserklärung

Hiermit erkläre ich, dass die Masterarbeit von mir selbst ohne unerlaubte Beihilfe verfasst worden ist und ich die Grundsätze wissenschaftlicher Redlichkeit einhalte (vgl. dazu: http://www.uzh.ch/de/studies/teaching/plagiate.html).

.....

Ort und Datum Unterschrift