11-712: NLP Lab Report:
A Dependency Parser for Swiss German

David Klaper

April 25, 2014

Abstract

This lab report details the different stages in building the first Swiss German dependency parser. Swiss German is the main spoken 'language' in Switzerland. The parser is trained on 10'000 tokens manually annotated dependency data. The data comes partly from an existing Swiss German text collection, which includes Wikipedia texts, a corporate report, and a newspaper. The second part of the data comes from online blogs mainly of Swiss high school students doing an exchange year. The blog texts collected for this project amounts to over 300'000 tokens Swiss German text that was not edited for publication. The experiments show that the blog text is much harder to parse. The accuracy of the parser for the blog texts is about 56% using only supervised learning and 61% when using additional unsupervised information. For edited text, such as Wikipedia the tentative performance is above 75% with the caveat that the PoS tagger was trained on the same data.

Swiss German is one of the 'languages' that does not have a standardized written form. Yet millions of Swiss people use this language to write emails, blogs, short messages, and sometimes even publicized materials. It is used in daily life in Switzerland even in business and retail contexts. Moreover, it is used as the main language on social media. Thus, to provide Swiss German speakers with appropriate recommendations, and understanding them Swiss German language technology is needed. This parser is an early step into the direction of powerful Swiss German language technology.

1 Basic Information about Swiss German

Swiss German is a group of Germanic dialects spoken in Switzerland. In 2000, about 4.6 million people in Switzerland spoke Swiss German (Lewis et al., 2013). By now, this number probably has increased since the overall population of Switzerland increased from 7.2 to 8 million people since 2000 (Swiss Federal Statistical Office, 2013). In general, Swiss German is quite similar to Standard German there are specific syntactic, lexical and other differences between Standard German and the Swiss German dialects. There are also considerable differences with regard to these features between different dialects. (Scherrer, 2011)

2 Past Work on the Syntax of Swiss German

The name Swiss German already indicates that it is closely related to (Standard) German. Many words and the basic syntactic structure are similar or equal to German. One often-cited characteristic of Swiss German is the existence of context-sensitive structures in some dialects as shown by Shieber (1985).

Due to the similarity to Standard German it makes sense to consider resources for Standard German syntax. There are many resources on Standard German syntax. One of them is the World

Atlas of Language Structures (Dryer and Haspelmath, 2013). It summarizes the syntactic and morphologic properties of a language as a list of features. Furthermore, there exist dependency treebanks and dependency parsers for Standard German. As an example Sennrich et al. (2009) present a hybrid dependency parser, which combines hand-written rules with a statistical model. The parser relies on supervised training data, which is not available for Swiss German yet.

We have seen that for Standard German there exist many powerful tools and datasets. Unfortunately, there are still considerable differences between the two. First, Swiss German is mainly a spoken language and no unified writing system exists. Furthermore, even within Switzerland the dialects vary considerably regarding pronunciation and in consequence spelling.

Scherrer (2007) attempted to normalize Swiss German words to their Standard German counterparts, which would allow using the Standard German resources. Although he created a working system, the results are below 50% for both precision and recall. Scherrer states also that "for many dialect words, it yields no result at all" (Scherrer, 2007, p. 60). Scherrer and Rambow (2010) worked towards machine translation from Standard German to specific Swiss German dialects and outlined how to use Standard German resources for creating a Swiss German constituent parser. This work is very interesting but it requires dialect identification and the performance is not good enough. Specifically, introducing a high error rate in a preprocessing step will make building a robust dependency parser even harder.

As stated before there are differences between Standard German and Swiss German as well as between the different Swiss German dialects. Scherrer (2011) proposes a system to normalize Swiss German dialects to Standard German syntax and explains some of the syntactic differences. Bucheli and Glaser (2002) started a project for mapping the differences between the Swiss German dialects in an atlas. However, as of the beginning of 2014 the atlas has not been published. As part of this project Glaser and Frey (2006) investigated reduplication phenomena in Swiss German dialects.

Beyond the resources describing Swiss German syntax and some tools to standardize Swiss German there are very few tools available. Also, there are no freely available corpora published yet, but Aepli and Hollenstein (2014) at the University of Zurich are working on a corpus of 50'000 tokens annotated with *Part-of-Speech* (PoS) tags. They are also building a PoS-Tagger for Swiss German. They trained a model which got them approximately 85% accuracy on Alemannic Wikipedia texts with 20'000 tokens training. (Aepli and Hollenstein, 2012). It does not discriminate between different dialects, which eliminates dialect identification as a source of error. Dialect identification is especially difficult because the borders are continuous and many people have lived in multiple dialect areas and have a mix of dialects (e.g. the writer of this document). This tagger could be very useful for the parser.

In conclusion, while Swiss German has many similarities with Standard German, for which many resources exist. The attempts to normalize Swiss German dialects to Standard German have yielded mediocre performance. There are some dialectometric studies detailing the differences between the distinct dialects but very few computational systems and no corpora. Aepli and Hollenstein (2014) are working on a corpus and PoS-Tagger for Swiss German. This tagger will not discriminate between different dialects which eliminates the dialect identification.

3 Available Resources

Since Swiss German is a mainly spoken language, written resources are sparse. In particular, organized corpora are basically non-existent or transcribed resources from interviews or otherwise artificial written text. The Zürcher Kompetenzzentrum für Linguistik (Zurich competence center for linguistics) compiled a list of corpora[1]. The problem with the list is that there are non-working links and most resources are only available on request and may not be shared. Finally, as mentioned above most of these resources are geared towards linguistic analysis use and are just transcriptions of spoken language.

For building their PoS tagger, Aepli and Hollenstein (2012) compiled data from the Alemannic Wikipedia[2] and annotated 20'000 tokens with PoS tags. Since the seminar thesis they have significantly increased the scope and size of their corpus. By now they have annotated over 50'000 tokens. First, there is a Swiss German edition of the 2012 corporate report of the Swiss firm Swatch[3]. The report contains more than 70'000 tokens of which 13'000 were annotated with PoS tags. Another source they used is the anniversary edition of a daily free newspaper in Switzerland [4], which was a special edition written in Swiss German. They annotated about 11'000 tokens. They also annotated extracts from murder stories by Viktor Schobinger[5] but I decided not to use this part, because the domain is quite different and probably even more difficult than the rest. (Aepli and Hollenstein, 2014)

In order to complement these resources, I decided to compile a new collection of Swiss German texts from the web, as an additional resource. I was considering Twitter text but this will probably be too noisy. Also, it might be hard to detect whether a short tweet is Swiss German or not. Therefore, I decided to search for Swiss German blogs. I restricted myself to BlogSpot blogs, in order to facilitate subsequent extraction. Using google queries involving typical Swiss German words and 'blog' I searched for mainly Swiss German blogs. Using this technique, I selected 17 blogs from people posting entries written in Swiss German. Interestingly, the queries I used returned almost exclusively returned blogs from Swiss people doing an exchange program or living abroad. I extracted 885 blog entries (but some contain only pictures no text and a few are in languages other than Swiss German).

I then proceeded to filter out useless posts. I extracted the xml, removed html tags and wrote each blog entry into its own file with the pattern 'blog_*blognumber*_*postnumber*.txt'. The blog number relates to one of the original 17 blogs. The postnumber indicates how old an entry is. Hence, the postnumber is in inverse chronological order. After extracting the text from the xml, I removed all files that were empty or less than 100 Bytes, since many posts only included pictures. Then, I manually went through all blog posts, deleting the entries with more than half written in another language. In a few special cases, e.g. song lyrics, I just removed these and kept the blog post if they were not vital to the content. At the end 615 blog posts remained.

After cleaning the data I wrote a script to count the tokens. As a token I defined a sequence of alphabetical or numerical characters of length 2 or more. Note that this definition ignores punctuation as tokens. I will later use a tokenization script provided by Aepli and Hollenstein (2014), where the count will be higher, but for now this is precise enough. The token count was 277'000 with 32'402 types. This is a type-to-token ratio of 11.7%, which is fairly high. Moreover, over half of the types (19'397) occur only once, i.e., they are hapax legonema.

Also, I will forward the data to Aepli and Hollenstein (2014) and they will annotate part of it to make it part of their future tagger training material. A side resource I might use (depending on the parsing framework I use) is a list of Swiss German words without German roots[6]. If this word

[1] `http://www.linguistik.uzh.ch/resources/korpora.html` last checked January 31st, 2014.

[2] `http://als.wikipedia.org` last checked last checked January 31st, 2014.

[3] Report available at `http://www.swatchgroup.com/de/investor_relations/jahres_und_halbjahresberichte` last checked January 31st, 2014.

[4] Edition May $28^t h$, 2013 in the archive at `http://www.blick.ch/blickamabend/epaper/` last checked January 31st, 2014.

[5] `http://www.zuerituetsch.ch/zueri-krimi.html` last checked January 31st, 2014.

[6] `http://www.dialektwoerter.ch` last checked January 31st, 2014.

Source	**Tokens with PoS**	**Tokens overall**
Wikipedia	20’000	400’000
Newspaper	11’000	17’000
Swatch Report	13’000	70’000
Blog Text	0	277’000
Total	**44’000**	**764’000**

Table 1: Rough estimated token counts

list helps, I might also consider German word lists.

For the reference corpora A and B, I plan to select data from each data set. In particular, corpus A will consist of a text-section from the Swatch report, Wikipedia, or the newspaper, which is at least 1000 tokens in one piece. Corpus B will consist of one or multiple blog posts from the same author, since the posts are usually reasonably contiguous. In short, I plan to use Swiss German text from Aepli and Hollenstein (2014), part of which was used to train the PoS tagger that I am using. In addition, I will use Swiss German blog text from mainly exchange students. As shown in Table 1, this collection will have over 250’000 tokens Swiss German text, of which I can use most for training. This is most likely the largest collection of Swiss German written text openly available with over 700’000 tokens. I hope that such data will reduce the effect of the many different ways to write the same word in Swiss German.

4 Survey of Phenomena in Swiss German

The basic word order is a bit tricky in German since the main clause has a subject-verb-object (SVO) order but the subordinate clauses are verb-final, i.e., subject-object-verb (SOV). German has a richer morphology than English but in Swiss German some of the distinct cases collapse to the same word form. Furthermore, Swiss German lacks a preterit tense, all sentences in the past are expressed using perfect. Finally, Swiss German does not have the genitive case. It is replaced by dative constructions. (Scherrer, 2011)

These are some general observations but the focus here is on syntactic phenomena that influence the dependency structure in crucial ways. Scherrer (2011) lists more interesting differences. One point are the non-projective dependencies in Swiss German. One reason for them is the verb raising with the auxiliary verb “lo” (*let*), where the dependency between the auxiliary and its argument and the main verb and its object will cross. There exist other forms of verb raising, where verb order differs from the Standard German word order. Also, this may depend on the dialect. In short, the word order of Swiss German is more free than Standard German.

Some more phenomena that illustrate differences between the well-studied Standard German and Swiss German, are a preposition before a dative in the case it is not already part of a prepositional phrase. This would have more impact on constituent parsing though. Scherrer (2011) lists more phenomena but some of them seem rather uncommon in central Switzerland. In particular, the very southern dialects have even more deviations, but most of the population lives in central and northern Switzerland, therefore these effects might not occur very much in the corpora. A last part that might be particularly tricking for dependency annotation are doubling phenomena in Swiss German as explained by Glaser and Frey (2006). For many of these phenomena it is not clear, which word is the head and which the dependent even if they are clearly related.

By inspection of the data, I looked for other interesting phenomena. One phenomena is the melting of verbs and personal pronouns (and similar combinations) that is also reflected in the tag set by Aepli and Hollenstein (2012). This is a major challenge for the annotation of dependencies, because the two words have independent syntactic roles but both its dependents will be added to this single token. There are many spelling variations, even within a document a term may not be written the same way or spaced differently in some cases. Hence, for data-oriented methods this may pose a problem.

For the blog corpus, there are some specific observations. First, often vowels are duplicated multiple times to place emphasis on a word, such as "gaaanz elei" (*totally alone*), which increases the number of types for the same lemma. Also, because the blogs are by Swiss people abroad there are many insertions of foreign language material, e.g. "sWetter isch totally crazy gsi" (*the weather was totally crazy*).

In conclusion, Swiss German is generally more free regarding word order than Standard German, especially with regard to auxiliary and main verb. This may yield non-projective dependency relations. Furthermore, verb and personal pronouns can sometimes be contracted to one word, as other combinations can. The spelling is not uniform and duplication of vowels is common. For dependency annotation especially the frequent use of auxiliaries and the duplication of words can be difficult, since the head-dependent relation is not obvious in these cases.

5 Initial Design

This section describes the annotation design and briefly outlines the next steps, i.e., building the actual parser. As stated before the two test corpora have a different focus. Corpus A consists of a small part of the data by Aepli and Hollenstein (2012). Corpus B consists entirely of blog posts. The data was annotated manually. To ensure annotation consistency the following documents the annotation decisions made.

Table 2 gives an overview of the annotation decisions. Before going into the detailed decisions an overall observation. The sentence splitting of the tokenizer is far from perfect, since capitalization cannot be used as a feature of sentence splitting. First, because some people write all lower case and second, because in German common nouns are capitalized. Therefore, it is possible to have multiple words attached to the root, which mean that the clauses are independent. Most of the decisions are semantically oriented. Also, I sometimes give English-only examples, when the matter is the same in both languages.

Since Swiss German does not use a past form, many sentences have auxiliaries. **Decisions 1 and 2** say that in sentences with multiple verbal parts, the inflected auxiliary is always the head. This is illustrated in the example below. The subscript numbers indicate the position of the parent in the sentence. Zero means root. The inflected verb *het* dominates *welle*, which dominates the main verb *lösche*. **Decisions 3 and 4** concern the attachment of the arguments. The subject always attaches to the inflected verb as a child. Thus, *hans* is a child of *het*, all other arguments and adjuncts attach to the main verb, e.g. *lösche*. There are certain special cases. Especially, modal verbs, such as *cha* ('can') can be modified by adverbial constructs. These attach to the verb they modify. Sometimes this is hard to decide, then they are attached to the main verb. Note that the verb *sött* ('should') can be used as a main verb. The reason for this is that the importance of the first auxiliary is reflected and the connection between subject and form of the inflected verb is explicit. Furthermore, the main verb defines what kind of objects are allowed, therefore it dominates the arguments.

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
De	hans$_3$	het$_0$	welle$_3$	d'	date$_{10}$	mit	em	magnet$_{10}$	lösche$_4$
The	Hans	has	want	the	data	with	the	magnet	delete

'Hans wanted to delete the data with the magnet'

Decision 5 states that subordinate clauses are headed by the subjunction that introduces them. Hence, the head verb in the embedded clause is dependent on the relative pronoun (or other subjunctive word). The reasoning is that the subjunction defines how the clause relates to the main sentence and hence, acts as the root of the embedded clause. **Decision 6** states that in a coordination the conjunction, typically *ond* or *und* ('and'), is the head of the conjuncts. Furthermore, should there be multiple parts to a conjunction of which all but the last are connected by commas, all parts of the conjunction will be direct dependents of and. (The comma is not reflected, since it is attached to the root, see below). If there are multiple 'and' (that are not semantically in a hierarchy), the 'and' dominate from left to right, i.e., the first and will be the highest. **Decision 7** determines that in numerical expressions like *öppe 3 stund* ('about 3 hours'), the unit is the head of the number and the number is the head of the modifier. The reason for this is that the number quantifies the unit and the modifier modifies the quantity. **Decision 8** is related to the fact that certain verbs are built by prepending a particle in front of the verb. In infinitive constructions this particle can be split from the verb. It is clear that this particle is a direct dependent of the verb it belongs to.

The important **Decision 9** says that prepositional phrases are headed by the noun. The reason is that there are large variations in the use of prepositions among the dialects, while the noun is always there. Furthermore, some dialects use sequences of prepositions and articles, while others contract both to one word. **Decision10** states that discourse markers, such as interjections and smileys are attached to the corresponding head of the phrase they belong to. Note that smileys might be attached to the root if they are outside of a sentence or clause context. Also, Swiss people use 'and' and other conjunctions sometimes as discourse fillers or simple linking elements, in that case this rule applies and not the coordination rule.

A special phenomena is the contraction of different words in Swiss German. For instance, one can contract a relative pronoun *wo* ('where, when') and the following personal pronoun *ich* ('I') to *wonich*. This was also observed by Aepli and Hollenstein (2012). Therefore, they introduced multiple PoS tags to deal with such phenomena. The problem is where in the dependency parse these constructions with multiple roles should end up. **Decision11** is the word should always adopt its strongest role, i.e., the word part that is higher in the dependency parse will determine the position of the contracted word. **Decision 12** concerns comparison phrases of the form below.

1	*2*	*3*	*4*	*5*	*6*
So	schön$_0$	wie$_2$	de	himmel$_6$	esch$_3$
So	beautiful	how	the	sky	is

'As beautiful as the sky (is)'

The point is that the main content of the phrase is the beauty. The verbal phrase is what is being compared to. It is a semantic construction to introduce a contrast. It cannot be inverted, thus, I decided that the verb is not the head in this case. **Decision 13** is similar to number 8. In German, 'no more' can be split and the object that is not available anymore will be between the two parts. Here, the more is clearly a dependent of no. **Decision 14** concerns sequences of multiple nouns. While general compounds are written together, some compounds can be split. There the head is the main content. However there is another special case. It is 'the word -bus-'. Here word will be the head and bus will be dependent. This is also a semantic decision, because the phrase is about a

ID	**Pattern**	**Head**
1	Mainverb Aux	Aux
2	Aux [...] Mainverb	Aux
3	Subj	Inflected Verb
4	Object, other Args, Adjuncts	Mainverb
5	RelPron Verb	RelPron
6	[XP] *and* [XP]	*and*
7	Mod Num Unit	Unit
8	Mainverb VerbPart	Mainverb
9	Prep N	N
10	DiscourseMarker	Corresponding Clause Head
11	Contractions	Placed At Highest
12	Adj Comparison MainVerb Aux	Comp
13	2 Part Negation (*no more*)	*no*
14	N_descriptor N_term	N_descriptor
15	Example	Main content
16	No Verb Sentence	Next Best, e.g. Noun
17	Punctuation	ROOT

Table 2: Annotation Decisions

word not a bus. **Decision 15** is also such a 'meta'-concern. In the phrase "for example [X]", X will be the head and example will be dependent, because the semantic content is X.

Decision 16 states that in a sentence that has no verb, the next most important word becomes the head. There are even sentences where a noun is the head and then a finite relative clause is embedded. This inheritance principle makes sense, although in the case of ellipsis it might be hard to determine what exactly should be the head. The decision should also be semantically motivated. **Decision 17** is simply all punctuation is strictly attached to the root. This is because punctuation is very unreliable in Swiss German, because the use by different speakers is inconsistent.

Finally. a couple of decisions not listed in the table, because they are minor. Foreign material, e.g. English, is annotated as much as possible according to English standards, where they exist or otherwise according to this manual. A sensitivity regarding passive formulations. In the passive voice, the main verb is a participle that may be hard to distinguish from an adjective, but it is as treated as a verb as long as it is clearly related to a verb. This leads to a distinction between 'has been possible', where 'been' will be the main verb that gets the arguments as its dependents and 'has been registered', where 'registered' is the main verb. There are subjunctions, such as *aber* ('but'), which can also be used instead to just modify a verb phrase, then they are dependent of the verb. Last, in 'dangerous to deadly', 'dangerous' is the head of 'to', which in turn is head of 'deadly'.

While in general the annotation worked quite well, it turns out that using and as the head of its conjuncts leads to unintuitive phenomena. In particular, if there is a coordinated verb phrase that applies to the same subject, then 'and' dominates both the conjuncts and the subject, which clearly do not have the same function. However, I decided to stick to the annotation schema because I did want to avoid preferring one of the two conjunct phrases over the other. Arguably, 'and' is also used as a linking word instead of a coordination, but this is a different case and should in my opinion not be mixed with the real coordination of phrases.

Model	3-fold X	10-fold X (min/max)	corpus A
Basic normal	61.04%	64.63% (54.69/71.15)	75.14%
Standard normal	60.97%	65.45% (54.69/70.81)	**75.25%**
Full normal	**61.39%**	–	74.81%
Basic lower	60.93%	64.89% (55.02/71.88)	74.92%
Standard lower	61.01%	**65.49%** (54.69/71.31)	74.59%
Full lower	61.23%	–	74.81%

Table 3: Results after first round

Finally, a couple of comments about the data. First, the data does not use Gold standard PoS but the ones that are assigned by the tagger model trained on 57'000 tokens Swiss German text. Second, the tokenization and in particular the sentence splitting are far from perfect. Part of this is due to the sloppy use of punctuation and part of it is due to the simplicity of the sentence splitter. Third, Wikipedia in particular tends to have special formatting characters which impede proper tokenization.

The parser will operate in a semi-supervised fashion, making use of annotated training material as much as possible. It is important to use unsupervised features as well, because a large amount of the available texts will not be annotated. It would be bad not to use this data. The semi-supervised method by Koo et al. (2008) based on Brown Clustering serves as the model for my unsupervised features. I consider using the TurboParser (Martins et al., 2010) as a baseline. Then I would try adding unsupervised features to improve over the supervised baseline. After annotation of the test data, I annotated 10'000 tokens (650 sentences) for training material in order to have the necessary supervised training. For annotation I used the DG Annotator by Giuseppe Attardi (`http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/` last visited February 22th, 2014.)

In conclusion, I tried to adopt a rather semantic notion of head and used this throughout the annotation. I will build a semi-supervised parser.

6 System Analysis on Corpus A

I trained various TurboParser models on the full training data. I also decided to check if there is a significant difference lowercasing everything after tagging (note that words in ALL-CAPITALS are lowercased before tagging in any case). It turns out that the difference between the models and casing are largely irrelevant. Performance is very similar even for the basic model, which is probably due to the small training size of 10'000 tokens.

The results in Table 3 show the unlabeled attachment accuracy results of cross-validation of the training set, as well as on corpus A. Note that corpus A contains only data from Aepli and Hollenstein (2014), i.e., Wikipedia, corporate report, and newspaper text. The 10-fold cross-validation was not done for the full model because training the full model takes over 10 minutes. For the 10-fold cross-validation also the score for the best and worst fold unlabeled accuracies are shown. I also did preliminary experiments with increasing regularization but they did not fundamentally change the results, therefore I do not report these numbers.

The table offers a few interesting insights. First, corpus A seems to be easy to parse, since the parser achieves very respectable 75% unlabeled accuracy. However, cross-validation on the training corpus draws a different picture. Accuracy on 10-fold cross-validation drops by about 10%. This indicates that the blogs, which are not present in corpus A are much harder to parse, because the drop is much higher than expected from a few sentences less in the training data. An indication of the same is also the big variance between different folds in 10-fold cross-validation. One reason is that the PoS tagger was trained on the same texts as used for training and evaluation, therefore the tagging accuracy is probably quite a bit higher than on the blog texts! Another reason might be that the blogs are written for a very different audience (family and friends in particular) and use more spoken phenomena than the texts that are intended to be read by abroad population, such as the corporate report and Wikipedia.

In conclusion, the performance of the TurboParser using supervised parsing is very good on corpus A but the blogs seem to be much harder. There are at least 2 reasons for that. First, the corpus A texts were used to train the PoS tagger. Second, the corpus A texts are intended for a much broader audience.

7 Lessons Learned and Revised Design

The results of this evaluation show that the parser is able to identify many structures correctly. I examined the results on corpus A by hand looking at the produced dependency trees. With 75% accuracy, the sentences are mostly correct with a couple of wrong attachments. I was surprised that it worked so well, because corpus A consists of multiple dialects and still it makes few mistakes in any of them. On the downside only few sentences are completely correct. Probably because many are rather long with an average of about 17 tokens per sentence.

A more in-depth analysis showed that many local structures were parsed correctly but longer attachments often lead to mistakes. It seems common mistakes are attaching the arguments to the wrong verbal form, attaching an adjective that has been used in adverb position to the right verb, and not identifying the main verb correctly. Some of the mistakes are still because of tagging mistakes, but overall the trees are mostly correct with a couple of wrong attachments.

The big difference between corpus A and cross-validation is partly due to different accuracy of the Part-of-Speech tagger. We saw that the accuracy for 10-fold cross-validation on the training set is about 10% lower than on corpus A. A small part of this is probably that there is fewer training material. The main reason is that the training material contains blogs. In particular, the blogs are not written for publication, their sentences are slightly longer and contain more pragmatic markers. Another important reason is that the PoS-tagger was trained only on the Wikipedia etc. data and not on blogs. Therefore, the tagging accuracy is probably quite a bit lower on the blogs.

In conclusion, the parser performs well on the first test corpus but corpus B will have lower accuracy. Since the tagger was trained on corpus A data, its tagging accuracy is higher, which improves score. Furthermore, these texts were written for publication, thus, they might follow more regulated structures.

For the next round, I will try to overcome the tagging mistakes on unknown words. Since most of the tagging mistakes are not due to ambiguity but due to too many unknown words, I will try to leverage unsupervised methods. Specifically, I want to use the unannotated data to train a Brown clustering, which will serve as additional features adding to the PoS-tags similar to Koo et al. (2008). This feature should help classifying the same kind of words similarly.

If I have time to develop more, I might think about additional features that can be learned in an unsupervised or semi-supervised fashion. Another approach I consider as a next step after clustering is using some simple rule-based morphological or similar features to help the parser.

The first priority is clearly implementing the clustering and testing the parser. Since there was not much difference between the models I will use the standard model. Since capitalization is less consistent in the blogs I will evaluate the impact of lowercasing all words again.

Model	**10-fold X**	**corpus A**	**corpus B**
Standard normal	65.45%	75.25%	56.57%
Standard lower	65.49%	74.59%	57.22%
C100 4P+F normal	67.39%	76.57%	59.91%
C100 4P+F lower	67.09%	75.36%	59.91%
Low C100 4P+F normal	67.64%	75.58%	60.88%
Low C50 4P+F normal	–	74.59%	59.48%
LowOnly C100 4P+F normal	67.41	76.02%	60.02%
LowOnly C50 4P+F normal	**68.01%**	**76.80%**	59.91%
LowOnly C50 4P+F+4Stem normal	67.91%	76.24%	**61.10%**

Table 4: Results after second round

8 System Analysis on Corpus B

As planned I decided to use clustering features to exploit all the unannotated data I had. For this purpose I took all data I had and removed all the data that was in the test corpora. Therefore, for all clustering experiments the training material contained about 860'000 unannotated tokens.

The idea behind clustering is that we can use information about words that appear in similar contexts to deal better with word forms that have not appeared in the small amount of annotated training data. This should increase performance on corpus A and B, compared to the featureless system used for corpus A.

Brown clustering (Brown et al., 1992) is a powerful clustering algorithm that produces a hierarchical clustering based on bigram token sequences. I used the implementation of Brown clustering by Percy Liang (Liang, 2005) for my experiments. I used the resulting bit strings directly as features. In particular, I create the clustering offline and use it at runtime of the parser. There can be multiple features from the same clustering. Concretely, I used both, the 4-bit prefix and the full bit string for most of my experiments. The idea is that the prefix acts more like a part of speech, while the full bit string acts more like a word form(Koo et al., 2008)[7].

Note that since at parsing time there might be words that are not part of the clustering set, there is a special value for unclustered words. This value does not provide any information and might even confuse the parser, yet I hoped there would be enough known words.

The user is able to change the number and form of features generated easily but the only experiments I performed were using only 4-bit prefix or only full bit string, both decreased the unlabeled accuracy slightly. Table 4 below shows my results, which will be explained in the following paragraphs. *10-fold X* is the average accuracy in 10-fold cross-validation on the training data. As expected, there is a dramatic drop in unlabeled accuracy on the blog corpus (corpus B) as compared to the text intended for publication (corpus A). As explored in the previous section, the reasons lie not only in the different text genres, but also that the PoS Tagger training material consists (at least partly) of corpus A, therefore the PoS tags are more accurate.

The models 'Standard normal' and 'Standard lower' are the same as in the previous round. The goal is to improve as much as possible over those in particular on the blog texts. I decided to concentrate on only using standard models, because full models take significantly longer to train. The remaining models are explained below.

[7] Note that I did not follow Koo et al. (2008), since they used a different parsing model and had much more

I started the clustering experiments with the above mentioned features of a 4-bit prefix string and the full bit string. I indicate this as *4P+F*. I did not try different prefix lengths, I only tried using only one of the features which decreased performance slightly. Using 800 clusters was not practical due to the training time. Thus, I used 100 clusters to start with indicated as C100. I used the original, tokenized text to create the clusters.

The first experiment just adding the features (4-bit prefix + full bit string) from the clustering with 100 clusters showed a significant improvement (C100 4P+F normal). All 3 measurements improved by more than 1% compared to the Standard normal model. In particular, corpus B with the blogs improved by more than 3% to almost 60% unlabeled accuracy. Therefore, the hopes of using clusters to generalize beyond word forms appearing in the training seem to work well.

Then I decided to see what the influence of casing would be. First, I used the clustering with true casing and used the lower-casing just for the parser as before. While it still improves significantly about the comparable previous system (Standard lower), it does not perform as good as with the true-cased parser. Thus, I decided to abandon the possibility of parsing lowercased text, since too much information is lost. It did not even improve on corpus B, compared to the normal casing. (But it improved before adding cluster features, which indicates casing is unreliable in corpus B, but the clusters mitigate this.)

So I wondered if I could make the clustering more reliable by lowercasing the data for clustering, but feeding the parser the true-cased data. This could at least slightly reduce the number of unknown words. This model improved corpus B but decreased corpus A performance each by about 1% (Low C100 4P+F normal). This was the first model that broke the 60% barrier for corpus B.

The next parameter I played with was the number of clusters in the original clustering. Koo et al. (2008) used 800 but for the amount of data available even small experiments with C200 showed a reduction in performance. Therefore, I focused on using 50 clusters instead of 100.

Comparing the previous model with 100 lowercased clusters and the typical 4P+F feature to the clustering with 50 clusters proved to be worse on both corpora. However, because of a mix-up of the results I found a third possibility to use the clusters. The models prepended with "LowOnly" take the following approach. The clustering is performed on the text with its original case. However, when the parser gets a sentence to parse, each word is lowercased for lookup (not for parsing). This effectively means only words that appear in lowercase in the clustered data are considered. I do not have a good explanation why this works well but it does. One possible theory is that words starting with a capital letter are much rarer and thus, they are less reliable, adding noise to the data.

For 100 clusters (LowOnly C100 4P+F normal) it does not work quite as well as the true lowercased version, but for the 50 clusters model (LowOnly C50 4P+F normal) it yields the overall best results for cross-validation and corpus A, while corpus B is stuck at 59.91% unlabeled accuracy.

After all my clustering results, I decided to try a very naïve kind of phonetic normalization, in order to account for the many spelling variations for words that are pronounced the same. Note that this does not account for any variations in pronunciation, of which there exist also many in the different dialects. I decided to distinguish only between 'dark' (a, o, u) and 'bright' (e,i) vowels and the umlauts. Additionally, 'ä' is considered the same as a bright vowel, because it is a common spelling difference. For the consonants, I decided to do less normalization but still normalize things that either sound the same or are common differences between standard German spelling and Swiss German pronunciation. Finally, I removed all repetitions of (normalized) letters.

Then I decided, I want to use only the first part of longer words. Therefore, I only use the first 4 characters of each normalized word, which might be more than just the first 4 letters of the original word. The number of 4 is also easily adjustable but I did not have time to perform experiments,

unannotated data.

Model	**corpus A**	**corpus B**
Low C50 4P+F+4Stem normal	74.14%	60.78%
LowOnly C50 4P+F+4Stem normal	75.25%	59.91%
Low C100 4P+F+4Stem normal	74.37%	**61.64%**
LowOnly C100 4P+F+4Stem normal	75.58%	61.31%
Low C100 4P+F normal	75.36%	60.78%
LowOnly C100 4P+F normal	**76.24%**	60.67%
Low C100 4P+F+6Stem normal	74.59%	61.42%
Low Only C100 4P+F+6Stem normal	75.47%	61.53%

Table 5: Results after putting all newspaper text also in NFC

whether the length is important. Using this feature decreased corpus A and cross-validation slightly, probably because it has more consistent spelling, but it increased corpus B by over 1% (LowOnly C50 4P+F+4Stem normal). This clearly shows how the different bloggers use different spellings. In the more formal texts the spelling is more consistent and messing with the spelling leads sometimes to different words being mapped to the same feature that have nothing in common, which reduces the accuracy a bit.

I tried to add the same feature to other configurations that worked well for all of them the performance would not increase and especially not get close to the 61.1% accuracy for corpus B.

In any case, all the parameters I showed can still be more thoroughly tested. The spelling difference normalization could be improved with more knowledge about Swiss German phonetics but it did improve performance by quite a bit, which was the main goal. The two models that performed best on a corpus are provided in the repository.

In conclusion, corpus B was much harder, as expected. Adding clusters and spelling difference normalization helped to improve the score on this blog corpus by almost 5%. These additions also helped to improve performance by about 1-2% on the cleaner data set where it also profits from better PoS-tag accuracy. Therefore, adding unsupervised features to the supervised parser was definitely a good idea for parsing this low-resourced language called Swiss German.

9 Final Revisions

Shortly after submitting the previous part, I discovered a strange error in the spelling normalization. In the data by the newspaper, the umlaut 'ä' normalizes as 'ö' instead of 'e'. However, in texts from other sources the normalization works correctly. An examination in a text editor and subsequent inspections in a hexadecimal editor showed what was going wrong.

Unicode knows two different ways to encode umlauts. In the composition normal form *NFC* the umlaut 'ä' is encoded as U+00E4, which is the codepoint for the character called *Latin small letter a with diaeresis*. In the decomposition normal form (NFD) the same letter is encoded using 2 characters: U+0061 (Latin small letter a) U+0308 (combining diaeresis). Thus, to my normalization algorithm the word 'Gepäck' looked like 'Gepa"ck'[8]. Thus, the 'a', which is a dark vowel was converted to 'o' and the combining diaeresis was ignored, resulting in 'ö'. For more on Unicode normal forms (including the compatibility normal forms) see Davis and Whistler (2013).

[8] TeXworks automatically converts the diaeresis to quotation mark but it should be the character with Unicode codepoint 0x0308.

Model Training | Model Test	**corpus A**	**corpus B**
old 57'000 no blogs	75.47%	61.53%
new 76'000 with blogs	73.59%	61.53%

Table 6: Result comparison of the two PoS-tagging models

Unfortunately, normalization made my results a bit worse as and quite different as shown in table 5. The two corpora proved again to be in a balance, increasing performance on the blogs resulted most often in a decreased performance on corpus A. One caveat is that the differences are so small that in a real task the difference in performance is not relevant at all. The normalization of spelling only had a minor impact if at all on corpus B and did not improve performance on corpus A at all. In fact, the LowOnly model without normalization performs best on corpus A (LowOnly C100 4P+F normal). Using a six characters prefix of the normalized word as opposed to a four characters prefix slightly increased performance. Nevertheless, the impact of the spelling normalization feature is very small.

The optimal number of clusters also changed from 50 to 100. Moreover, the properly lowercased clustering is now closer to the LowOnly version, which neglects a significant part of the data. For the final version I will provide the *LowOnly C100 4P+F+6Stem normal* model (bold), because it keeps unlabeled accuracy on corpus A above 75% but also performs better than the previous best model for the blog texts. (It is fair to say that other models could also be considered, but I want to provide one standard model.) It is interesting to note that what is still consistent is that the LowOnly version outperforms the Low version on corpus A. This corroborates, that lowercasing everything has a negative effect on corpus A. The results are less conclusive on corpus B.

As for why adding data decreases performance on corpus A, while it increases on corpus B. there are two explanations. First, the differences are so small that it is simply random whether it increases or decreases. Second, the newspaper redaction might have had some internal guidelines that were followed, when writing the articles. The umlauts that appear in most sentences allowed the parser to separate the sentence types in the newspaper from the rest of corpus A. The redaction guidelines promoted certain sentence structures. Thus, it had a better performance on the newspaper article in the test set, because it could leverage the knowledge of these typical structures, while on the rest of the data these typical structures were not present and thus did not introduce confusion. Please note, that this is a pure guess.

The last experiment looks at the influence of the PoS tags. In the second evaluation it was hypothesized, that the PoS tags have a large influence on the performance. Nora Hollenstein took the collected blog data and annotated blogs from 3 speakers and trained a new PoS-tagger model with data from both sources and a total of 76'000 tokens. Note that one of the blog texts in the test set was also part of the training set for the PoS tagger. Thus, the tagger kind of cheats on this document, similar to the documents in corpus A. As a consequence, I expect the performance on corpus B to improve. I compare the performance of the *LowOnly C100 4P+F+6Stem normal* model on the two test corpora. Table 6 shows the results. Nora Hollenstein already indicated the additional data might hurt performance and indeed the performance goes down on corpus A and does not improve on corpus B. One reason might be that the blog data is a lot less uniform. Thus, to get a bonus from using blog data more data would need to be annotated. Maybe it is just too varied and it is almost impossible to improve tagging performance when using blog texts. The model will not be provided, since there is no additional gain. This shows that PoS tags do have an influence on the result. It also emphasizes the big difference between the two domains!

In conclusion, the final round concentrated on analyzing the effects of an error induced in the preprocessing of the data. The difference in Unicode normalization between the newspaper text and the rest of the data skewed previous results. However, the absolute differences between all the different models are very small. This also means that the spelling normalization in the second round was not effective. Moreover, using a PoS tagger also trained on blog texts did not improve performance. The final unlabeled accuracies are about 75% for corpus A (Wikipedia, newspaper, corporate report) and about 61.5% on corpus B (blog texts).

10 Future Work

One important aspect is validating and extending the lexical resources. The annotation of the existing data sets was performed only by me and I also did not go back to check everything at second time after all decisions were made. Thus, there are likely annotation mistakes. Also, 10'000 tokens is still a very modest resource. Annotating more resources, maybe even from new domains could be very helpful to improve performance on general text. Of course, the biggest issue is still the different spelling. Since there are so many dialects, some speakers might get a very good performance for their texts, while others might get a very poor performance. Discriminating these different users and maybe even having different models per dialect can be one way to mitigate this but needs a lot more data.

Another point for further investigation is the use of rule-based features. The spelling normalization did not prove helpful. Further research could be directed into why it did not improve performance. From this new phonetically or linguistically oriented features could be derived. Another approach would be to look at the work of Scherrer and Rambow (2010) to see which features are implementable. Using phonetic features might help to reduce the spelling differences, which would reduce the data sparsity a bit. Using morphological features might improve the performance of distinguishing different forms, since Swiss German also has some case marking. Both help the parser using the data to make more informed choices. This project has shown it is not straight forward to design good features of this kind, but with more time and resources it will be possible.

Finally, leveraging standard German resources could mitigate the data sparseness problem. There is an abundance of resources for standard German. There are dependency treebanks and powerful morphological analyzers readily available. The big problem is that Swiss German has different spelling and some different syntactic structures. Scherrer (2011) has done some work to mitigate these differences. If the Swiss German text could be normalized enough to make use of Standard German resources, a parser trained on German and Swiss German material could result in very good performance. This would be a very good test for a "true" Swiss German parser that does not try to make it standard German. Only a parser with intricate knowledge of Swiss German could probably outperform the normalized standard German version that can use large amounts of existing data.

11 Acknowledgements

I'd like to thank Noëmi Aepli and Nora Hollenstein for supporting me with information about Research in Swiss German and especially for giving me early access to the Swiss German resources that they are developing. In addition, I'd like to thank Nora Hollenstein for annotating the blog texts and creating a new PoS tagger model.

References

Noëmi Aepli and Nora Hollenstein. Part-of-Speech tagging für Schweizerdeutsch. Seminar Thesis, Institute of Computational Linguistics, University of Zurich, 2012.

Noëmi Aepli and Nora Hollenstein. Personal Communication (January), 2014.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

Claudia Bucheli and Elvira Glaser. The syntactic atlas of Swiss German dialects: Empirical and methodological problems. In *Syntactic Microvariations*. Meertens Institute, 2002.

Mark Davis and Ken Whistler. Unicode normalization forms. `http://www.unicode.org/reports/tr15/`, 2013.

Matthew S. Dryer and Martin Haspelmath, editors. *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, 2013. URL `http://wals.info/languoid/lect/wals_code_ger`.

Elvira Glaser and Natascha Frey. Doubling phenomena in Swiss German dialects. In *Proceedings Workshop on Syntactic Doubling in European Dialects*, 2006.

Terry Koo, Xavier Carreras, and Michael Collins. Simple semi-supervised dependency parsing. In *Proceedings ACL*, 2008.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig. Ethnologue: Languages of the world. 17th Edition. SIL International. `http://www.ethnologue.com/language/GSW`, 2013.

Percy Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, 2005.

André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings EMNLP*, pages 34–44, 2010.

Yves Scherrer. Adaptive string distance measures for bilingual dialect lexicon induction. In *Proceedings ACL*, pages 55–60, 2007.

Yves Scherrer. Syntactic transformations for Swiss German dialects. In *Proceedings EMNLP*, 2011.

Yves Scherrer and Owen Rambow. Natural language processing for the Swiss German dialect area. In *Proceedings KONVENS*, 2010.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. A new hybrid dependency parser for German. In *Proceedings GSCL Conference*, 2009.

Stuart M. Shieber. Evidence against the context-freeness of natural languages. *Linguistics and Philosophy*, 8:333–343, 1985.

Swiss Federal Statistical Office. Ständige wohnbevölkerung nach geschlecht und staatsangehörigkeitskategorie, am ende des jahres. `http://www.bfs.admin.ch/bfs/portal/de/index/news/04/01.Document.141977.xls`, 2013.