The Devil is in the Details: Parsing Unknown German Words

Daniel Dakota(✉)

Department of Linguistics, Indiana University, Bloomington, USA
ddakota@indiana.edu

Abstract. The statistical parsing of morphologically rich languages is hindered by the inability of parsers to collect solid statistics because of the large number of word types in such languages. There are however two separate but connected problems, reducing data sparsity of known words and handling rare and unknown words. Methods for tackling one problem may inadvertently negatively impact methods to handle the other. We perform a tightly controlled set of experiments to reduce data sparsity through class-based representations in combination with unknown word signatures with two PCFG-LA parsers that handle rare and unknown words differently on the German TiGer treebank. We demonstrate that methods that have improved results for other languages do not transfer directly to German, and that we can obtain better results using a simplistic model rather than a more generalized model for rare and unknown word handling.

1 Introduction

Parsing morphologically rich languages (MRLs) has proven to be a challenge for statistical constituent parsing. The relative success for English has not been achieved on other languages, particularly MRLs as the computational methods and algorithms that yield good results are not directly transferable to other languages, which have been shown to be intrinsically harder to parse (Nivre et al. 2007). This can be attributed to the various linguistic properties these languages possess (e.g. freer word order), which present difficulties for capturing their more complex syntactic behaviors. Such properties are attributed to a higher degree of inflectional morphology, resulting in increased data sparsity from a substantial proportion of word types occurring rarely in a text (Tsarfaty et al. 2010).

German contains characteristics of more rigid word order languages like English, such as verb placement, but also possesses many phenomena that are present in MRLs, such as generally freer word order, resulting in being coined as a morphologically rich-less configurational language (MR&LC), a position between configurational and non-configurational languages (Fraser et al. 2013). The language also possesses problematic phenomena for NLP, such as case syncreticsm, which require information between morphology and syntax to more accurately disambiguate constituents.

G. Rehm and T. Declerck (Eds.): GSCL 2017, LNAI 10713, pp. 23–39, 2018.
https://doi.org/10.1007/978-3-319-73706-5_3

In order to improve statistical parsing in general, but especially for MRLs, two problems must be addressed: the need to reduce data sparsity and the treatment of unknown words. Many tools, such as POS taggers and parsers, have sophisticated internal mechanisms to handle unknown words and by default often perform better than simplistic probability models. However, the weakness of sophisticated models is they can over-generalize, biasing themselves against the very words for which they are trying to compensate. A simplistic unknown word handling model, which is not affected in this way, can benefit greatly from both the reduction of data sparsity and simplistic treatment of unknown words, surpassing results from more sophisticated models. We examine two separate but connected problems, the interaction between parser-internal probability models for handling unknown and rare words and performance, while also simultaneously reducing data sparsity issues using Brown clustering and word signatures for rare and unknown words.

The paper is structured as follows. We discuss previous literature in Sect. 2 followed by our experimental methodology in Sect. 3. Results and discussion are presented in Sects. 4 and 5 respectively, before concluding in Sect. 6.

2 Related Work

2.1 Handling Rare and Unknown Words

Handling rare and unknown words are two separate but equal components that are intrinsic to many NLP applications such as lemmatizers, POS taggers, and parsers. The task becomes more difficult when processing MRLs, due to the exponential increase of word forms as they have higher ratios of word forms to lemmas (Tsarfaty et al. 2010). The chosen methodology however has different practical applications for different NLP tasks. A POS tagger may only need to be concerned with the lexical level within a trigram, whereas a parser may be concerned with an unlexicalized trigram and the constraints of its own grammar. Thus the probability models and goals of the tools are not the same, and an effective method used for one task may not be ideal for another.

A reasonable treatment of rare words that occur below a given threshold is to handle them identically to unknown words due to the inability to obtain reliable distributional statistics. The most simple approach is to reserve a proportion of the probability mass, assigning each word equal weight and mapping them to an UNK symbol. This *simple lexicon* is universal in its application, but suffers from an oversimplification of the problem, and its inability to make more informed decisions. Specifically, each unknown word will be given equal weight when intuitively we know that certain words are more likely to occur in a sequence. These probabilities are obtained from a training set with the majority tag becoming the default (see Attia et al. (2010) for a comprehensive parsing example), which, strictly speaking, determines the the tool's performance on any subsequent application.

More sophisticated approaches try to allow for generalization while taking into account that not all words are equally likely. The PCFG-LA Berkeley parser

(Petrov and Klein 2007a,b) uses rare words to estimate unknown words by obtaining statistics on the rare words with latent tags and uses linear smoothing to redistribute the emission probabilities across the rare words (Huang and Harper 2009, 2011). Although this allows for good generalizations in a PCFG-LA setting, this has been shown to cause rare words to suffer more from over-fitting than frequent words (Huang and Harper 2009) and to not effectively handle out-of-vocabulary (OOV) words as it can only generate probabilities of words seen in the training data (Huang et al. 2013). The parser also has what is referenced as a *sophisticated model*, which uses a more linguistically informed approach to handle OOV words by exploiting word formation characteristics, such as affixes and capitalization, but its approach has been shown to be biased towards an English lexicon (Hall et al. 2014). The development of language specific signatures can considerably improve performance (Huang and Harper 2009; Attia et al. 2010), but is often ignored in practice.

2.2 Word Clustering

A by-product of a more robust morphological system in a language is an increase in word forms, resulting in an increase of data sparsity. Various forms of clustering have been utilized to reduce sparsity issues and increase class-based representations to improve performance through better probability emissions.

Brown clustering (Brown et al. 1992) is a unsupervised hard clustering algorithm that obtains a pre-specified number of clusters (C). The algorithm assigns the C most frequent tokens to their own cluster. The C+1 most frequent token is assigned to one of the pre-specified C clusters by creating a new cluster and merging the C+1 cluster with the cluster that minimizes the loss in likelihood of the corpus based on a bigram model determined from the clusters. This is repeated for every each $(C+N)th$ individual word types within the corpus, resulting in a binary hierarchical structure with each cluster encoded with a bit string. Words can be replaced by their bit string, thus choosing a short bitstring can drastically reduce the number of words in a corpus, allowing for a flexible granularity between POS tags and words. The distributional nature of the algorithm lends itself to the problem of clustering words that behave similarly syntactically by grouping words based on their most likely distribution, adding a semantic nuance to the clustering.

On what linguistic information: words, lemmas, or inflected forms; to perform clustering for MRLs is not obvious. Various linguistic information on which Brown clustering has been performed has yielded different results for different languages. This is further compounded by how cluster information can be incorporated into different parsers and the impact this has on each parser's performance.

Koo et al. (2008) demonstrated that cluster-based features for both English and Czech outperformed their respective baselines for dependency parsing. Ghayoomi (2012) and Ghayoomi et al. (2014) created clusters using word and POS information to resolve homograph issues in Persian and Bulgarian respectively, significantly improving results for lexicalized word-based parsing.

Candito and Crabbé (2009) clustered on *desinflected* words, removing unnecessary inflection markers using an external lexicon, subsequently combining this form with additional features. This improved results for unlexicalzed PCFG-LA parsing for both medium and higher frequency words (Candito and Seddah 2010), but was comparable to clustering the lemma with its predicted POS tag.

In contrast to Candito et al. (2010) who did not achieve substantial improvements for French dependency parsing using clusters, Goenaga et al. (2014) created varying granularities of clusters using words (for Swedish) and lemmas plus morphological information (for Basque) to obtain noticeable improvements for dependency parsing. Versley (2014) noted that cluster-based features improved discontinuous constituent parsing results for German considerably, but results are influenced by cluster granularities.

3 Methodology

3.1 Clustering Data

The data used for generating Brown clustering is a German Wikipedia dump consisting of approximately 175 million words (Versley and Panchenko 2012). The data includes POS and morphological information representative of the annotation schemas of TiGer. A single sequence of POS tags and morphological features was assigned using the MATE toolchain (Björkelund et al. 2010) with a model trained using cross-validation on the training set via a 10-fold jackknifing method assigning information regarding lemmas, POS tags, and morphology. We added the TiGer corpus into the Wikipedia data and retained punctuation, which may provide contextual clues for certain words for clustering purposes. We clustered on raw words, lemmas, and a combination of lemma and part of speech tags (lemma_POS) to obtain 1000 clusters for tokens occurring with a minimum frequency of 100.

3.2 Methods

For training and development, the TiGer syntactic treebank 2.2 (Brants et al. 2004) was utilized, specifically the 5k train and dev set from the SPMRL 2014 shared task data version (Seddah et al. 2014). Importantly, punctuation and other unattached elements are attached to the tree following Maier et al. (2012), resolving crossing-branches (for a full description of the data preprocessing, see Seddah et al. (2013b)).

Parsing experiments were performed using the Berkeley parser (Petrov and Klein 2007a,b) and the Lorg parser (Attia et al. 2010) which is a reimplementation of the Berkeley parser. The parsers learn latent annotations and probabilities (Matsuzaki et al. 2005; Petrov et al. 2006) in a series of split/merge cycles that evaluate the impact of these new annotations and merge back those deemed

least useful, performing smoothing after each cycle, while calculating the EM after each step.[1]

The Lorg parser uses a *simple lexicon* unless a specific language signature file is specified.[2] In principle this is equivalent to the Berkeley setting of *simple lexicon* option, a point that will be further investigated in Sect. 4. The default unknown threshold for Lorg is five while the default rare word threshold for Berkeley it is 20. We experimented with German signatures for German unknown words and clusters to test the impact on results.

3.3 Evaluation

The SPMRL 2013 shared task scorer (Seddah et al. 2013b) was used for evaluation to report F-scores and POS accuracy. This script is a reimplementation of EVALB (Sekine and Collins 1997), but allows for additional options, such as completely penalizing unparsed sentences, which we include. We do not score grammatical functions and remove virtual roots with a parameter file, but do score for punctuation. We report results for both providing the parser with gold POS tags and parser-internal tagging on the development set[3] reporting the average over four grammars using four different random seeds $(1, 2, 3, 4)$ as Petrov (2010) noted that EM training within a PCFG-LA framework is susceptible to significant performance differences.

4 Results

4.1 Rare and Unknown Word Thresholds

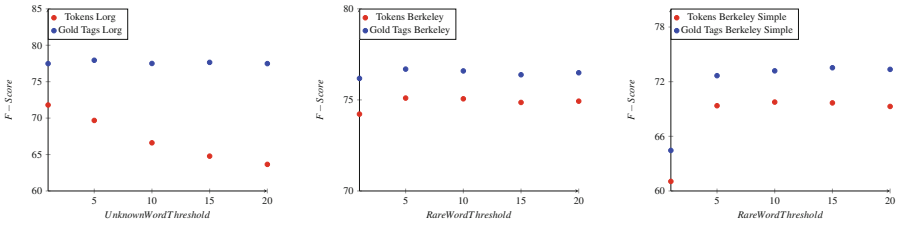
Figures 1a to c show results for different settings of the unknown threshold for Lorg and the rare word threshold for Berkeley. The influence of the unknown threshold on Lorg's performance is negligible when the parser is given tags, but is significant for parser-internal tagging, with performance dropping by around 10% absolute. This is expected considering how easily influenced the *simplex lexicon* is by word frequencies. The small data sets may have an impact, but preliminary experiments with the full data sets show a similar trend, but less pronounced. The impact the rare word threshold have on Berkeley (see Fig. 1b) using the *sophisticated lexicon* however is not as pronounced for both gold tags and parser-internal tagging. The internal smoothing algorithm seemingly allows it to be less influenced by a change in its rare word thresholds, even with a small data set, as more words are simply subsumed, keeping the grammar rather intact.

[1] We trained without grammatical functions, due to the time it took in preliminary experiments to parse TiGer with grammatical functions, and use a split/merge cycle of 5.

[2] This currently only exists for English, French, and Arabic.

[3] The test set is left for final evaluation after further experimentation, although we note that the TiGer test set has been shown to be substantially harder to parse than the dev set (see Maier et al. 2014).

It is worth nothing however that the optimal setting is around 5 and not the default setting of 20. In order to examine the impact smoothing has on Berkeley, we performed experiments using the parser's *simple lexicon* option, presented in Fig. 1c, which is said to be the same as Lorg's *simple lexicon* model. These results stand in contrast to not only the results with the Berkeley's *sophisticated lexicon* smoothing of rare words, but the *simple lexicon* model of Lorg. Although the curves in Figs. 1b and c are similar, the actual performance is better using Berkeley's *sophisticated lexicon* approach, but these results can be partially attributed to the number of unparsed sentences (in the 100 s in some cases) for which the parser is penalized, as it is unable to find rules within its grammars for the given inputs. There is a substantial increase in F-score from a threshold of 1 to 5, but minimal increases there afterwards, with the best performance at a threshold of 15. The stark differences between the *simple lexicon* model implemented by Berkeley and Lorg suggests that there are undocumented implementation differences which are not strictly identical.



a UNK Thresholds Lorg Simple Lexicon

b Rare Word Thresholds Berkeley Sophisticated Lexicon

c Rare Word Thresholds Berkeley Simple Lexicon

Fig. 1. Rare and unknown word thresholds

In order to examine on what linguistic representations Brown clustering can be performed that has yielded improvements for other languages, we perform experiments on German by replacing all terminals with their POS tags, their lemmas, and lemmas and pos_information, with results presented in Table 1. Only results for the best performing unknown threshold (UNK TH.) for each parser is given, as well as for the lexicon reduction (Lex. Red.). Lexicon reduction is defined as the proportional decrease in the vocabulary size of the word types from the original Tiger dev set to the dev set replaced with clusters and UNK types.

For both lemmas and lemma_POS, all terminals with the following tags were replaced with their tags respectively: CARD, FM, and XY. Punctuation was left in its original form. When replacing terminals with POS tags, there is a drop in the F-score between gold tags and parser-internal tag of between 4–6% absolute for Lorg while this drops to between 1–2.5% for Berkeley. Every Lorg with gold tags outperforms its Berkeley counterpart, which is noteworthy given

Table 1. Results for orig, lemma, POS, and lemma_pos blue = gold POS tags | red = parser-internal tags

Parser	Terminal type	Parsed	UNK TH.	F-score	POS Acc.	Lex. Red.
Lorg	orig	tokens	1	71.80	90.81	N/A
	orig	tagged	5	77.94	99.54	N/A
	POS	tokens	15	74.64	99.54	**99.61**
	POS	tagged	15	74.65	99.54	**99.61**
	lemma	tokens	1	71.54	90.87	27.83
	lemma	tagged	5	77.25	99.53	27.83
	lemma_pos	tokens	1	73.15	93.70	18.95
	lemma_pos	tagged	5	77.30	99.54	18.95
Berkeley	orig	tokens	5	75.10	94.04	N/A
	orig	tagged	5	76.69	99.87	N/A
	POS	tokens	15/20	74.20	98.89	**99.61**
	POS	tagged	15/20	74.17	99.92	**99.61**
	lemma	tokens	5	73.56	92.89	27.83
	lemma	tagged	5	75.91	99.83	27.83
	lemma_pos	tokens	10	75.21	95.97	18.95
	lemma_pos	tagged	10	76.01	99.93	18.95

that Lorg consistently has a higher number of unparsed sentences for which it is penalized, while Berkeley outperforms Lorg for parser-internal tagging, except for POS terminals. This suggests that the default handling of rare and unknown words is influential on the parsers subsequent performance on any downstream application without further enhancements, as Berkeley outperforms Lorg in its base form. Furthermore, a threshold of 1 on Lorg consistently achieving the best results should not be surprising as Attia et al. (2010) explicitly note that lower thresholds for Lorg perform best, thus the default thresholds are not necessarily ideal for a given language. This is supported by Seddah et al. (2013a), who noted that a threshold of 1, or true unknown words, resulted in the best performance for French out-of-domain parsing.

For Berkeley, the original treebank outperforms all other variations with gold POS tags, but for Lorg, replacing the terminals with their POS actually achieves the best performance for parser-internal tagging with lemma_pos performing second best overall. The results regarding replacing POS tags confirm the findings of Benoît and Candito (2008). Given that latent variables are obtained by splitting a terminal into two categories, it would seem reasonable that variation in terminals is needed for better approximation of latent categories, as such differences percolate up the tree. However, it is interesting to note that terminals consisting of POS tags still outperform replacing terminals with lemmas for parser-internal tagging. Replacing terminals with lemmas likely results in increased ambiguity of the syntactic nature of terminals.

4.2 Suffix Results

Not all words in the treebank have a cluster ID. In such cases, words can be considered rare or even unknown, even though they may appear in both the training and dev set, but are infrequent. In order to group infrequent words into more generalized categories, each non-clustered word is replaced with a UNK token, with various suffix lengths. Here a suffix is not inherently linguistically oriented, but strictly character length. Table 2 shows the impact that various suffix lengths of unknown words have on performance on Lorg.[4] The experiment *raw+orig* replaces terminals with cluster IDs and leaves the original terminal untouched if no cluster can be found. For all other experiments, words with no assignable cluster were changed to UNK_suffix$N$ where N is the length of the suffix on the UNK token (e.g. UNK_suffix2 for the word *spielen* "to play" would be UNK_en). The parser with gold POS tags shows little variation in performance on the suffix length. For parser-internal tags, there is slightly more variation but not substantial.

Table 2. Suffix length for UNK words for Lorg

Token type	Parsed	UNK TH.	F-score	POS Acc.	Lex. Red.
raw+orig	tokens	1	75.90	93.45	59.24
raw+orig	tagged	1	78.16	99.52	59.24
raw+unk_suffix0	tokens	1	75.88	93.26	**93.86**
raw+unk_suffix0	tagged	1	78.26	99.45	**93.86**
raw+unk_suffix1	tokens	5	76.14	94.05	93.45
raw+unk_suffix1	tagged	5	78.05	99.53	93.45
raw+unk_suffix2	tokens	5	76.27	94.23	91.09
raw+unk_suffix2	tagged	10	78.20	99.40	91.09
raw+unk_suffix3	tokens	1	76.05	93.86	86.61
raw+unk_suffix3	tagged	5	78.10	99.40	86.61
raw+unk_suffix4	tokens	1	76.03	93.92	80.63
raw+unk_suffix4	tagged	5	78.34	99.49	80.63

Although the best suffix length is not clear, we choose a suffix of length 2 for our additional experiments for three reasons: (1) it achieves the best results on average for parser-internal tagging; (2) it adequately balances between lexicon reduction and additional information as the German alphabet consists of 30 letters,[5] thus a suffix of length two will have at most $30^2 = 900$ possible combinations where a suffix of length 4 will have $30^4 = 810000$ possible combinations;

[4] Experiments with Berkeley showed less variation.

[5] We note that not all possible letter sequences are likely or plausible (e.g. *ßß*).

(3) a suffix of length 2 has more linguistic motivation as most inflectional morphology in German is identifiable within 2 characters thus categorization of unknown words in terms of POS type is feasible, though not absolute.

4.3 Cluster and Signature Results

In order to examine the interaction between different signatures, cluster-based features, and lexicon reduction, we performed experiments with various additional modifications of unknown words as well as open and closed classes to better understand the interaction between such treebank representations and parsing models, presented in Tables 3 and 4. If a token had no corresponding cluster it was replaced with a UNK representation with additional information attached, with capitalization (C) indicated on all tokens (clustered and UNK). We also experimented with not replacing closed class words with their corresponding cluster ID, and instead leaving them in place ($noCC$). Once again, we see little difference in F-scores when providing the parser tags, but we see more range with parser-internal tagging.

Table 3. Results for Lorg on raw words and lemma_pos clusters

Token type	Parsed	UNK TH.	F-score	POS Acc.	Lex. Red.
Craw	tokens	1	76.47	94.22	**93.38**
Craw	tagged	5	78.34	99.52	**93.38**
raw_suffix2	tokens	5	76.27	94.23	91.09
raw_suffix2	tagged	10	78.10	99.40	91.09
Craw_suffix2	tokens	1	76.50	94.57	89.98
Craw_suffix2	tagged	1	78.17	99.40	89.98
raw_noCC	tokens	1	76.00	93.68	92.73
raw_noCC	tagged	1	78.10	99.54	92.73
Craw_suffix2_noCC	tokens	1	76.57	94.93	88.86
Craw_suffix2_noCC	tagged	5	78.20	99.54	88.86
Clemma_pos	tokens	1	76.86	96.54	93.32
Clemma_pos	tagged	1	77.44	99.51	93.32
lemma_pos_suffix2	tokens	1	76.78	96.69	91.63
lemma_pos_suffix2	tagged	1	77.67	99.52	91.63
Clemma_pos_suffix2	tokens	5	76.77	96.63	90.54
Clemma_pos_suffix2	tagged	5	77.46	99.54	90.54
lemma_pos_noCC	tokens	1	73.67	94.08	**94.04**
lemma_pos_noCC	tagged	10	77.48	99.53	**94.04**
Clemma_pos_suffix2_noCC	tokens	1	76.08	95.61	90.53
Clemma_pos_suffix2_noCC	tagged	5	77.45	99.53	90.53

Results for Lorg indicate a distinct split. When *noCC* is not included, lemma_pos clusters obtain consistently higher performance, but when *noCC* is included, raw words perform consistently better. One reason may be that there is still too much ambiguity present with a lemma_pos combination, particularly with articles. However, we are still able to increase results for parser-internal tagging by over 5% absolute and more than .3% with gold tags. It is worth noting that the best achieved score is using gold tags with a suffix of length 4 (see Table 2) or simply marking capitalization on raw clusters and unknown words (see Table 3).

Table 4. Results for Berkeley raw words and lemma_pos clusters

Token type	Parsed	UNK TH.	F-score	POS Acc.	Lex. Red.
Craw	tokens	5	75.59	93.72	**93.38**
Craw	tagged	5	76.89	99.76	**93.38**
raw_suffix2	tokens	5	75.28	93.82	91.09
raw_suffix2	tagged	10	76.50	99.84	91.09
Craw_suffix2	tokens	5	75.66	94.27	89.98
Craw_suffix2	tagged	5	76.65	99.76	89.98
raw_noCC	tokens	1	75.23	93.29	92.73
raw_noCC	tagged	1	76.95	99.36	92.73
Craw_suffix2_noCC	tokens	1	75.73	94.68	88.86
Craw_suffix2_noCC	tagged	10	76.60	99.87	88.86
Clemma_pos	tokens	5	75.76	96.27	93.32
Clemma_pos	tagged	5	75.90	99.87	93.32
lemma_pos_suffix2	tokens	5	75.64	96.46	91.63
lemma_pos_suffix2	tagged	5	75.93	99.85	91.63
Clemma_pos_suffix2	tokens	10	75.82	96.69	90.54
Clemma_pos_suffix2	tagged	10	75.93	99.88	90.54
lemma_pos_noCC	tokens	1	72.49	93.33	**94.04**
lemma_pos_noCC	tagged	1	75.81	93.32	**94.04**
Clemma_pos_suffix2_noCC	tokens	1	75.00	95.23	90.53
Clemma_pos_suffix2_noCC	tagged	1	75.91	99.83	90.53

For Berkeley there are some similar trends (see Table 4), including the steep decline in lemma_pos performance when *noCC* is included. Although we are able to improve results over the Berkeley baselines, the increase in performance is around .3% absolute for gold tags and .6% for parser-internal tagging, although there is significantly less variation between settings.

5 Discussion

There is no direct correlation between lexicon reduction and parser performance. Clearly, reducing the lexicon helps performance, but it is not the case that the largest reduction results in the best performance. As discussed in Sect. 2, previous research has yielded strategies that have improved performance in other languages, such as lemmatization, but these do not benefit German to the same extent. This suggests that for German, simply reducing the lexicon is not enough, rather certain linguistic information, particularly at the morphological level, may need to be retained for certain word classes to help resolve errors.

A break-down of the most frequent UNK tokens is presented in Tables 5 and 6 extracted from the Craw_suffix2_noCC data from the train and dev set respectively. For some suffixes, NNs are either the only tag or represent almost all

Table 5. Top 10 UNK_ in raw train

UNK type	Count	Top 3 POS categories		
CUNK_en	897	NN (836)	NE (36)	ADJA (15)
UNK_en	624	ADJA (279)	VVINF (134)	VVFIN (89)
CUNK_er	429	NN (332)	NE (72)	ADJA (22)
CUNK_ng	255	NN (231)	NE (23)	ADJA (1)
CUNK_te	127	NN (115)	ADJA (8)	NE (3)
CUNK_es	112	NN (86)	NE (18)	ADJA (7)
CUNK_rn	110	NN (110)		
UNK_er	108	ADJA (79)	ADJD (18)	NN (7)
CUNK_in	106	NN (69)	NE (37)	
CUNK_el	103	NN (74)	NE (27)	PITA (1)

Table 6. Top 10 UNK_ in raw Dev

UNK type	Count	Top 3 POS categories		
CUNK_en	884	NN (795)	NE (32)	VVPP (6)
CUNK_er	515	NN (351)	NE (123)	ADJA (34)
UNK_en	462	ADJA (185)	VVINF (122)	VVFIN (82)
CUNK_ng	265	NN (253)	NE (10)	FM/ADJD (1)
CUNK_te	174	NN (166)	NE (4)	ADJA (4)
CUNK_rn	108	NN (103)	NE (3)	ADV (2)
CUNK_ft	101	NN (95)	NE (6)	
UNK_er	94	ADJA (68)	ADJD (17)	NN (6)
CUNK_es	91	NN (74)	NE (11)	ADJA (6)
UNK_te	89	VVFIN (49)	ADJA (38)	ADV/NN (1)

words in the signature. This can most likely be attributed to German orthography, where all nouns, both common and proper, are capitalized. From a syntactic perspective, they behave similarly, even though they may have different POS tags with NN being a common noun and NE being a proper noun. Results indicate this is perhaps the single most important signature, especially given German's notorious ability to generate new compounds words, many of which will seldom be seen.

The consistency between the types of UNK found between the two sets is indicative of why the suffix information is pertinent, as, although none of the words have a corresponding cluster ID, their POS tag and suffix information allow more unknown words to be grouped together for better probabilistic emissions. From a *simple lexicon* perspective, such a grouping of words should allow for better probabilistic modeling due to an increase in frequency.

However, the distinction between adjectives and verbs is a point that could use more refined signature differences, which is most evident with the *UNK_en* signature which handle words ending in *en*. Linguistically the intermingling makes sense as infinitive verbs will end in *-en*[6] while strong adjective endings will also have the same ending. Obtaining morphological characteristics of this UNK type, either case or person information, may resolve this overlap and improve performance as adjective and verbs exhibit syntactically different behaviors. However, past participles can behave similarly to adjectives when used as such, which may also influence the coalescence in this unknown type.

Further exploration of the POS tags and larger groups of the UNK words will allow for a better understanding of how the parsers choose to tag these words and whether they align consistently with provided tags as well as the linguistic characteristics of the true word.

5.1 External POS Tagger

We also examined the interaction between using an external POS tagger trained on the same data set, but with its own rare and unknown word probabilistic model on parsing performance. We trained the TnT tagger (Brants 2000) on the Craw_suffix2_noCC and Clemma_pos training sets and tagged the development sets respectively. TnT is a language-independent HMM tagger that employs multiple smoothing techniques using linear interpolation and handles unknown words using suffix information. The predicted tags were used as input for both Lorg and Berkeley, results of which are presented in Table 7. Using the TnT tags with the Berkeley parser are extremely similar to results with Berkeley-internal tagging, consistent with the findings of Maier et al. (2014). However, this may be attributed to the fact that both use smoothing within their probabilistic models and simply converge to a similar outcome. However, the results for Lorg are worse than those seen in Table 3. This is good evidence that the smoothing techniques used to generate tags by TnT directly conflict with the preferred tags generated by *simple lexicon* grammar model of Lorg and is ultimately detrimental to its

[6] or "-n" in many cases.

performance. This motivates that a closer examination between the interaction of different methods of both unknown word handling among not just among parsers, but also this interaction between parsers and POS taggers in a pipeline approach. Different tools in the pipeline handle unknown words differently and the chosen methods will influence the interactions between tools in the pipeline, impacting performance.

Table 7. TnT results

Token type	System	F-Score	POS Acc.
Craw_suffix2_noCC	TnT	n/a	94.43
	Lorg w/TnT Tags	74.62	94.28
	Berkeley w/TnT Tags	**75.70**	**94.65**
Clemma_pos	TnT	n/a	**96.66**
	Lorg w/TnT Tags	**76.03**	96.26
	Berkeley w/TnT Tags	75.56	96.26

5.2 Number of Clusters

In order to examine how much the impact on the number of clusters has on the performance of the *simple lexicon*, we performed a set of experiments with Lorg where we used an unknown threshold of 1 for both Craw_suffix2_noCC and Clemma_pos on parser-internal tagging, presented in Table 8. We chose our initial clustering parameters based on what has been a standard approach, but determining the optimal clustering size is not intuitive and requires extensive experimentation (see Derczynski et al. (2015)), as which clusters are splitting and which are combined when the number of clusters size is changed cannot be determined beforehand. The results indicate little variation between the cluster sizes, with 800 being optimal for the raw clusters and 1000 for the lemma_pos clusters. Interestingly, as the cluster sizes increase, the POS accuracy also increases, although the parsing performance does not. Changing the number of clusters will not increase the overall coverage, but simply alter the learned probabilities of the words already covered. Experiments by Dakota (2016) noted that although a minimum frequency of 100 may cover almost 90% of the tokens, it only covers roughly 30% of the actual token types in the TüBa-D/Z treebank (Telljohann et al. 2015). Reducing the minimum frequency to 3 ultimately yielded the best results for the creation of data-driven POS tags. Changing the minimum frequency a word must appear to be clustered will thus require optimal cluster sizes to be determined anew. Furthermore, when not replacing closed class words (*noCC*), a more in-depth evaluation is needed to see which cluster IDs (and by extension which types of words) are most prevalent and which are not. This will allow a better understanding of which types of words are being covered and excluded, but will naturally be influenced by any adjustment to the minimum frequency during the clustering process.

Table 8. Different cluster sizes

Token type	Cluster size	F-score	POS Acc.
Craw_suffix2_noCC	500	76.48	94.06
	800	76.65	94.64
	1000	76.57	94.93
	1500	76.60	95.12
	2000	76.45	95.22
Clemma_pos	500	76.67	95.73
	800	76.78	96.37
	1000	76.86	96.54
	1500	76.81	96.72
	2000	76.66	96.87

6 Conclusion and Future Work

We have shown that there is an intricate interaction between reducing data sparsity and the handling of unknown words. Better understanding this interaction allowed us to increase parser performance over our baselines, with best results obtained by using Brown clusters created from a combination capitalization and lemma_pos information. Although smoothing helps create better generalized models, it biases itself against the handling of rare and unknown words, which is in line with previous work examining such interactions within a PCFG-LA framework (Huang and Harper 2009, 2011). This technique has somewhat unexpected effects as although it helps with data sparsity, it results in lower performance. We were able to achieve maximum results when using a *simple lexicon* model for unknown word handling, as the simplistic division of the probability mass allowed us to better exploit the clustering of data through cluster IDs and word signatures without the bias against seldom seen word types.

There are a number of interacting variables that occur in the process of reducing data sparsity, each requiring an extensive in-depth evaluation to better understand how a modification or implementation to solve one aspect directly positively or negatively impacts another aspect. Future work will examine what linguistic information can be exploited on different word classes as well as exploring cluster granularity. There is a balance between the reduction of data sparsity and the need to create generalized enough models, the interaction of which is an area worth further exploration, particularly for MRLs; which consistently present such challenges. We will also examine whether the minimum frequencies during the clustering process can help reduce the number of unknown words further while adjusting the cluster numbers, to compensate for too much of an increase.

Acknowledgments. I would like to thank Djamé Seddah for his gracious help and input with the experiment design and implementation. I would also like to thank Sandra Kübler and Andrew Lamont for their comments, as well as the anonymous reviewers.

The research reported here was supported by the Chateaubriand Fellowship of the Office for Science & Technology of the Embassy of France in the United States and by the Labex EFL.

References

Attia, M., Foster, J., Hogan, D., Roux, J.L., Tounsi, L., van Genabith, J.: Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In: Proceedings of SPRML 2010 (2010)

Crabbé, B., Candito, M.: Expériences danalyse syntaxique statistique du français. In: Actes de la 15éme Conférence sur le Traitement Automatique des Langues Naturelles (TALN08), Avignon, France, pp. 45–54, June 2008

Björkelund, A., Bohnet, B., Hafdell, L., Nugues, P.: A high-performance syntactic and semantic dependency parser. In: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, Beijing, China, pp. 33–36, August 2010

Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., Rohrer, C., Smith, G., Uszkoreit, H.: TIGER: linguistic interpretation of a German corpus. J. Lang. Comput. **2004**(2), 597–620 (2004)

Brants, T.: TnT: a statistical part-of-speech tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, ANLC 2000, Seattle, Washington, pp. 224–231, April 2000

Brown, P., Della, V., Desouza, P., Lai, J., Mercer, R.: Class-based n-gram models of natural language. Comput. Linguist. **19**(4), 467–479 (1992)

Candito, M., Crabbé, B.: Improving generative statistical parsing with semi-supervised word clustering. In: Proceedings of the 11th International Conference on Parsing Technologies, IWPT 2009, Paris, France, pp. 138–141 (2009)

Candito, M., Seddah, D.: Parsing word clusters. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, SPMRL 2010, Los Angeles, California, pp. 76–84 (2010)

Candito, M., Nivre, J., Denis, P., Anguiano, E.H.: Benchmarking of statistical dependency parsers for french. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010, Beijing, China, pp. 108–116 (2010)

Dakota, D.: Brown clustering for unlexicalized parsing. In: Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016), Bochum, Germany, pp. 68–77, September 2016

Derczynski, L., Chester, S., Bøgh, K.: Tune your brown clustering, please. In: Proceedings of Recent Advances of Natural Language Processing (RANLP) 2015, Hissar, Bulgaria, pp. 110–117, September 2015

Fraser, A., Schmid, H., Farkas, R., Wang, R., Schütze, H.: Knowledge sources for constituent parsing of German, a morphologically rich and less-configurational language. Comput. Linguist. **39**(1), 57–85 (2013)

Ghayoomi, M.: Word clustering for Persian statistical parsing. In: Isahara, H., Kanzaki, K. (eds.) JapTAL 2012. LNCS (LNAI), vol. 7614, pp. 126–137. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33983-7_13

Ghayoomi, M., Simov, K., Osenova, P.: Constituency parsing of Bulgarian: word-vs class-based parsing. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, pp. 4056–4060, May 2014

Goenaga, I., Gojenola, K., Ezeiza, N.: Combining clustering approaches for semi-supervised parsing: the BASQUE TEAM system in the SPRML'2014 shared task. In: First Joined Workshop of Statistical Parsing of Morphologically Rich Language and Syntactic Analysis of Non-Canonical Languages, Dublin, Ireland (2014)

Hall, D., Durrett, G., Klein, D.: Less grammar, more features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, pp. 228–237, June 2014

Huang, Q., Wong, D.F., Chao, L.S., Zeng, X., He, L.: Augmented parsing of unknown word by graph-based semi-supervised learning. In: 27th Pacific Asia Conference on Language, Information, and Computation, Wenshan, Taipei, pp. 474–482, November 2013

Huang, Z., Harper, M.: Self-training PCFG grammars with latent annotations across languages. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 832–841, August 2009

Huang, Z., Harper, M.: Feature-rich log-linear lexical model for latent variable PCFG grammars. In: Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, pp. 219–227, November 2011

Koo, T., Carreras, X., Collins, M.: Simple semi-supervised dependency parsing. In: Proceedings of ACL-08: HLT, Columbus, Ohio, pp. 595–603 (2008)

Maier, W., Kaeshammer, M., Kallmeyer, L.: Data-driven PLCFRS parsing revisited: restricting the fan-out to two. In: Proceedings of the Eleventh International Conference on Tree Adjoining Grammars and Related Formalisms (TAG+11), Paris, France, pp. 126–134, September 2012

Maier, W., Kübler, S., Dakota, D., Whyatt, D.: Parsing German: how much morphology do we need? In: Proceedings of the First Joined Workshop of Statistical Parsing of Morphologically Rich Language and Syntactic Analysis of Non-Canonical Languages (SPMRL-SANCL 2014), Dublin, Ireland, pp. 1–14, August 2014

Matsuzaki, T., Miyao, Y., Tsujii, J.: Probabilistic CFG with latent annotations. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, Ann Arbor, Michigan, pp. 75–82, June 2005

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Prague, Czech Republic, pp. 915–932, June 2007

Petrov, S.: Products of random latent variable grammars. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, pp. 19–27, June 2010

Petrov, S., Klein, D.: Improved inference for unlexicalized parsing. In: Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, pp. 404–411, April 2007a

Petrov, S., Klein, D.: Learning and inference for hierarchically split PCFGs. In: Proceedings of the National Conference on Artificial Intelligence, Vancouver, Canada, pp. 1663–1666, July 2007b

Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp. 433–440, July 2006

Seddah, D., Candito, M., Anguiano, E.H.: A word clustering approach to domain adaptation: robust parsing of source and target domains. J. Logic Comput. **24**(2), 395–411 (2013a)

Seddah, D., Tsarfaty, R., Kübler, S., Candito, M., Choi, J.D., Farkas, R., Foster, J., Goenaga, I., Galletebeitia, K.G., Goldberg, Y., Green, S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski, A., Roth, R., Seeker, W., Versley, Y., Vincze, V., Woliński, M., Wróblewska, A., de la Clergerie, E.V.: Overview of the SPMRL 2013 shared task: a cross-framework evaluation of parsing morphologically rich languages. In: Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages, Seattle, Washington, USA, pp. 146–182, October 2013b

Seddah, D., Kübler, S., Tsarfaty, R.: Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In: Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages, Dublin, Ireland, pp. 103–109. Dublin City University, August 2014. http://www.aclweb.org/anthology/W14-6111

Sekine, S., Collins, M.: EVALB bracket scoring program (1997). http://nlp.cs.nyu.edu/evalb/

Telljohann, H., Hinrichs, E.W., Kübler, S., Zinsmeister, H., Beck, K.: Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Seminar für Sprachwissenschaft. Universität Tübingen, Germany (2015)

Tsarfaty, R., Seddah, D., Goldberg, Y., Kübler, S., Candito, M., Foster, J., Versley, Y., Rehbein, I., Tounsi, L.: Statistical parsing of morphologically rich languages (SPMRL): what, how and whither. In: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, SPMRL 2010, Los Angeles, California, pp. 1–12 (2010)

Versley, Y.: Incorporating semi-supervised features into discontinuous easy-first constituent parsing. In: First Joined Workshop of Statistical Parsing of Morphologically Rich Language and Syntactic Analysis of Non-Canonical Languages, Dublin, Ireland (2014)

Versley, Y., Panchenko, Y.: Not just bigger: towards better-quality web corpora. In: Seventh Web as Corpus Workshop (WAC7), Lyon, France, pp. 44–52 (2012)