Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, January 2012

Better tags give better trees – or do they?

Ines Rehbein[⋆], **Hagen Hirschmann**[†], **Anke Lüdeling**[†], **Marc Reznicek**[†]

Published by CSLI Publications

LiLT volume 7, issue 10 January 2012

Better tags give better trees – or do they?

Ines Rehbein[⋆], Hagen Hirschmann[†], Anke Lüdeling[†], Marc Reznicek[†], [⋆]*Universität Potsdam,* [†]*Humboldt-Universität zu Berlin*

Abstract

Parsing learner data poses a great challenge for standard tools, since non-canonical and unusual structures may lead to wrong interpretations on the part of the taggers and parsers. It is well known that providing a statistical parser with perfect part-of-speech (POS) tags is of great benefit for parsing accuracy, and that parsing results can decrease considerably when the parser has to predict its own POS tags. Therefore one might expect that even small improvements in POS accuracy have a positive effect on parsing performance. In this paper we test this assumption and assess the impact of POS tag accuracy on constituency parsing for German learner language. We compare different strategies to manual correction of the learner text and specific POS tags, and we measure the time requirements for each strategy. We show that tagging a canonical equivalent of the non-canonical learner text substantially improves POS tag accuracy. Correcting selected POS tags can only lead to parsing results comparable to a setting where all POS tags are corrected, while reducing annotation time substantially. However, the manual corrections of the POS tags do not result in a statistically significant improvement for parsing, giving evidence for the high quality of the automatically predicted parts-of-speech for the corrected learner data.

LiLT Volume 7, Issue 10, January 2012.
Better tags give better trees – or do they?.

1 Introduction

This paper reports on a small step - the evaluation of POS tag correction for parsing - in a larger research endeavour, namely the automatic analysis of learner language. Learner corpora (in our case essays produced by advanced L2 learners of German) are a valuable resource for the study of acquisition patterns. There are a number of error annotation schemes but schemes for the grammatical annotation of the learner language itself are rare. In order to understand acquisition phases it is necessary to compare learner patterns to native speaker patterns (Granger et al., 2002, Lüdeling, 2011). In order to compare grammatical patterns (in addition to lexical ones) we need high quality tagging of parts of speech as well as of syntactic relations (Doolittle, 2008, Hirschmann et al., to appear). In addition to the many conceptual problems (see below) there are, of course, problems that arise because learner language is so different from the training data used for the tools for automatic analyses of corpora.

Statistical parsers constitute robust tools for the syntactic analysis of text. In addition, they can be acquired from existing treebanks, which avoids the laborious process of handcrafting a grammar. This advantage, however, comes at a cost. Handcrafted grammars are often more accurate and better at handling generalisations (see, e.g., Krivanek and Meurers (2011) for a comparison of a statistical and a rule-based dependency parser on learner data), while statistical parsers are highly domain-dependent and give best results when applied to data similar to the training data. In which ways learner data is similar to or different from data produced by native speakers is still not well understood. There are only very few studies on learner data that look beyond lexical data (for syntactic annotation of learner data see (Dickinson and Ragheb, 2009, Dickinson and Lee, 2009, Rosén and Smedt, 2010)).

One factor influencing parsing accuracy is the quality of the POS tags. Best results are achieved when providing the parser with perfect POS tags for the input tokens. When the parser has to resort to automatically predicted POS tags, performance can drop considerably. Petrov and Klein (2008) report a decrease in f-score in the range of 0.6-1.8% on German text, while Rafferty and Manning (2008) observe 2-3% lower f-scores for automatically assigned POS tags on the same data. For other languages, this gap can be even larger (see, e.g., Marton et al. (2010), Ambati et al. (2010) for results on Arabic and Hindi).

In the paper, we investigate the impact of formulating target hypotheses on POS tagging accuracy as well as the impact of POS accuracy on statistical parsing. Our study is situated in the context of

treebanking. As stated above, we aim at creating a high-quality learner corpus for research in Foreign Language Acquisition as well as for studies in all areas of theoretical linguistics. This means that automatically derived analyses from the output of a state-of-the-art parser are not good enough for our purposes. Instead we aim for manually corrected syntactic analyses of high quality. Our interest is in semi-automatic techniques to support manual correction of automatically derived annotations to keep annotation costs low. We argue that for analysing learner data syntactically it is neccessary to formulate target hypotheses for structures that deviate from 'canonical' forms (in many cases these structures can be described as ungrammatical). We aim to prove this statement by testing the POS tagging accuracy for both the original learner data and the target hypotheses.

We present experiments where we a) assess the time requirements needed for manual correction of automatically assigned POS tags and b) compare the impact on parsing accuracy for different settings during the POS correction step. Furthermore, we want to demonstrate our approach to the syntactic analysis of learner language.

2 Related work

There exist a number of learner corpora, some of them augmented with error annotation (see e.g. Granger (2008)). However, not much work has been done on grammatical annotation of learner data, as stated above. Syntactic annotation of learner data is time-consuming and cannot rely on off-the-shelf NLP tools because learner data often deviates from 'native norms'.

Previous studies show that the accuracy of POS tagging of L2 English (Haan, 2000, van Rooy and Schäfer, 2002, Meunier and Mönnink, 2001) is substantially lower than the one for L1 English. We are not aware of any comparable evaluations for L2 German.

Another reason why we would expect lower results for POS tagging of our learner data is the fact that the data comes from argumentative essays which may differ from the training data of the tagger.

In addition, the variation within a learner corpus is larger than in a comparable L1-corpus (because learner language is influenced by the L1 of the learners, their proficiency level and many not yet well-understood factors). Furthermore, it is not clear whether we can apply annotation schemes developed for a standard variety of a native language to learner data. Díaz-Negrillo et al. (2010) discuss problems for part-of-speech annotation for areas where learner language systematically deviates from native language. They claim that for standard native language the part-

of-speech tag of a word token is determined on the basis of the token's distribution (syntactic slot), its morphological marking, and its lexical stem. It has often been shown that even for native language these pieces of evidence often lead to diverging POS systems (for discussions see e.g. Knobloch and Schaeder (2000) or Wierzbicka (2000)). All common POS tagsets use mixed systems to decide one way or another (see e.g. the AMALGAM project[1] for an overview of POS tagsets for English). For learners - who might use only one or two of the clues - we see more confusion and the clues point to divergent word classes for the same token. Therefore, Díaz-Negrillo et al. (2010) propose a tripartite POS analysis, encoding each of the linguistic clues without forcing a final, uniform word class label on the token which would fail to give an adequate description of its properties.

In contrast to Diaz-Negrillo et al., we do not try to process the learner data itself, but revert to so-called target hypotheses (TH) of the learner text (Hirschmann et al., 2007, Reznicek et al., 2010). Target hypotheses are manually corrected versions of non-canonical (in many cases: ungrammatical) learner utterances, including e.g. the correction of spelling, word formation, or word order errors.

The use of target hypotheses is not uncontroversial (Tenfjord et al., 2006, Lüdeling, 2008) because it is an interpretation of the learner data, and there could be many possible interpretations; here we use the target hypothesis simply as a technical device - a version of the text that can be parsed. In the end the parse will be mapped back to the original learner utterance. Even without an explicit target hypothesis learner data needs to be interpreted for any kind of annotation which means that there is always an implicit TH. We have argued elsewhere (Hirschmann et al., 2007, Lüdeling, 2008, 2011) that it is preferable to make the TH explicit. It is therefore formulated according to guidelines which warrant that the target hypotheses are as closely related to the learner utterances as possible. We corrected only spelling and inflection.

Relevant to our work are also studies focussing on reducing time requirements and cost for manual linguistic annotation. Dickinson and Meurers (2003) propose a method to automatically detect inconsistent POS annotations, based on variation in the corpus. This method is not suitable for us, as even corrected learner data is expected to display a high degree of variation, and thus inconsistencies do not necessarily indicate errors.

Dandapat et al. (2009) present experiments on fine-grained POS annotation of Hindi and Bangla and conclude that high-quality linguistic

[1] http://www.comp.leeds.ac.uk/amalgam/amalgam/amalghome.htm

annotation requires expertise and supervision, and that an intelligent annotation tool can crucially speed up the annotation process and enhance the inter-annotator agreement between the coders. We agree with their findings and provide our annotators with an annotation tool where the POS tags to be corrected are automatically pre-selected and highlighted, and where the most probable analysis is ranked highest in a list of alternative POS tags.

3 POS-correction and parsing experiments

We present experiments where we a) assess the time requirements needed for manual correction of automatically assigned POS tags and b) compare the impact on parsing accuracy for different settings. More precisely, we compare results when providing the parser with POS of varying quality, as obtained by three different POS taggers, and with manually corrected input.

We use the following off-the-shelf taggers: (1) the TreeTagger (Schmid, 2004), a probabilistic POS tagger using decision trees; (2) the Stanford Log-linear POS Tagger (Toutanova et al., 2003) which is based on a maximum entropy model; and (3) the RFTagger (Schmid and Laws, 2008) which combines the idea of a Hidden Markov Model with decision trees.[2] Our annotation scheme is the Stuttgart Tübingen Tag Set (STTS) (Schiller et al. (1995), see Appendix), the standard POS tag set for German.

3.1 Data

We test the impact of POS tag accuracy on statistical parsing on the Falko corpus (Lüdeling et al., 2008), an error-annotated learner corpus of German as a foreign language. Falko includes 248 argumentative essays (124.512 tokens) from advanced learners and 94 essays of German native speakers (69.526 tokens) collected under the same conditions. Each learner sentence is annotated with a target hypothesis (TH, see table 1). This TH is used for further processing to enable the tools trained on standard written German texts to process the data. THs are constructed in the same way for the native speaker data so that we have comparable corpora. It also allows to investigate systematic deviations of the learner texts from the corrected version of the learners' utterances, including over- and underuse of specific constructions.

[2]The RFTagger provides a fine-grained analysis including morpological information. For the sake of comparison we converted the output to the more coarse-grained tag set of the STTS.

LT:	Arzte	,	Lehrer	,	und	Bauern	spielen		Rolle
	doctors	,	teachers		and	farmers	play		role
TH:	Ärzte	,	Lehrer		und	Bauern	spielen	eine	Rolle
Diff:	CHA	,		DEL				INS	

TABLE 1 Target hypothesis for a learner utterance exemplifying changes, deletions and insertions

POS tagging of learner data and target hypotheses As stated before, the main motivation for creating target hypotheses is the need for a more normalised version of the learner data for automatic processing. We expect the POS tagging accuracy to be significantly higher for the THs than for the learner text. Since the THs are formulated token-based, each token (word form) of the TH has a corresponding token on the learner utterance layer which allows us to map back the POS tags of the TH to the original learner text. Exceptions are tokens that are inserted or deleted in the TH. We exclude these cases in our evaluation. Tokens which deviate between the two layers are predominantly spelling errors, but also inflectional errors, morphological errors and other word form errors. To show in how far these non-canonical tokens affect the tagger negatively, we let the RFTagger predict POS tags for the original learner data as well as for the target hypotheses. For the original learner data, we achieve a POS accuracy of 93.8% against the manually corrected POS tags of the whole Falko corpus which we get as the outcome of our experiment. For the target hypotheses, the accuracy crucially increases to 98.7%. This shows that the formulation of target hypotheses is worthwhile for POS tagging. We expect that the erroneous predictions of POS tags for the original learner data will have a negative impact on parsing results.

Gold standard We manually corrected POS tags and constituency structure for 200 sentences randomly extracted from the Falko corpus (100 L1 sentences and 100 L2 sentences) which had been parsed using the Berkeley parser (Petrov and Klein, 2007). This data set will be referred to as the Falko200 and will be used as a gold standard for parser evaluation. Each sentence has been corrected independently by two annotators, and disagreements between the resulting sets have been jointly resolved. All 5 annotators were post-graduates with linguistic training.

Experimental setup We divided the remaining Falko data into 12 batches with 500 sentences each (6000 sentences in total) which we used in our experiments (see table 2). Another 594 sentences served as

	description	no. sentences
Falko	Falko200 gold standard	200
	test set for assessing tagger quality	125
	coder training set	594
	batches 1 - 12	6000
TiGer	parser training set	48474

TABLE 2 experimental set-up of the Falko data

training data to ensure that the coders were familiar with the annotation tool, and to mitigate possible training effects. An additional set of 318 sentences from the Falko corpus was used in a pilot study to assess the quality of the different POS taggers as well as to measure inter-annotator agreement between the human coders. The POS correction has been done by two post-graduate students of linguistics, who also participated in creating the gold standard.

All sentences in batches 1-12 were automatically tagged by each of the three taggers. In the first setting, the annotators were instructed to correct all POS tags that the three taggers did not agree on (*correct-all*). In the second setting, they only had to correct those POS tags that the three taggers did not agree on *and* where one of the taggers had predicted a verb tag (*verbs-only*). The intuition behind this is that correctly identifying the verbs in each sentence is of crucial importance for the parser. We hypothesise that POS tagging errors concerning verb tags will cause more severe errors for the syntactic analysis than most other tagging errors, and that correcting verb tags will substantially improve parsing results while, at the same time, keep annotation costs low.

For assessing parsing accuracy, we used the Berkeley parser, an unlexicalised latent variable PCFG parser which uses a split-and-merge technique to automatically refine the training data. The splits result in more and more fine-grained subcategories, which are merged again if not useful. The parser was trained on the TiGer treebank (excluding sentences 8001-10000). For preprocessing, we resolved the crossing branches in the trees, following Kübler (2005), and attached all non-head constituents higher up in the tree. Grammatical functions (GF) were stripped off. Our parser output trees include GF, which we add in a post-processing step, using the method proposed in Seeker et al. (2010). This has the following advantages: 1) we obtain smaller grammars which are more efficient for parsing, and 2) we also avoid sparse

data problems. Finally, the accuracy of the GF assigned during post-processing is slightly higher than for the GF predicted by the parser. The tags which were not subject to correction were taken from the output of the RFTagger.

3.2 Pilot study

In the pilot study, we manually annotated 125 sentences (L2) from the Falko corpus with POS tags. The annotation was done by the same two coders who participated in the POS correction experiment. In the pilot study, however, they did not correct automatically assigned POS tags but annotated them from scratch. This data was used to assess the performance of the three POS taggers used in our experiments. We needed to know whether considering only those POS tags where the taggers disagree will result in many erroneous tags included in the data, as it might be possible that the taggers all predict the same incorrect POS tag for a specific word form. In addition, the pilot study allowed us to measure inter-annotator agreement (IAA) between our two coders.

Inter-annotator agreement IAA between our annotators was quite high with a percentage agreement of 97.9% and a Fleiss' κ of 0.978. There were 40 cases where the two annotators disagreed. The most ambiguous POS classes in the STTS are the following: the distinction between adverbs and adverbial adjectives, between prepositions and comparative particles in the case of "als" (*as*), between predicative adjectives and past participles, as well as between attributive adjectives and attributive indefinite pronouns. This is also reflected in the training data, the TiGer treebank, where these cases also display inconsistencies in annotation. As a result, these cases do also cause problems for statistical POS taggers. All these cases were resolved according to the STTS annotation guidelines, and the resulting sentences were then used for evaluating the POS taggers.

POS tagger accuracy Table 3 shows tagging accuracies on the target hypotheses of the learner data. The RFTagger obtains best results with only 33 incorrectly assigned tags. Results for the TreeTagger and the Stanford tagger are still acceptable with an accuracy of >0.96. In addition to the POS classes difficult for both humans and automatic POS taggers, the distinction between finite and non-finite verbs seems to be hard for automatic systems only (table 4).

We are interested in assessing the benefit we get from supporting manual correction by providing the annotators with the output of three

tagger	acc.	no. err.
Stanford	0.962	72
TreeTagger	0.969	60
RFTagger	0.983	33

TABLE 3 POS tagger accuracies

POS taggers. Thus, we need to know how many errors we would miss when correcting only POS tags where the three taggers disagree. In our test set with 1921 tokens this would be exactly 2 errors. This shows that disagreements between tagger predictions are a good indicator for tagging errors, and focussing on disagreements only results in a substantial time saving.

3.3 POS correction experiment

Having established that it is a valid strategy to correct tagger disagreements only, we now compare an experimental setting where the two coders correct all non-agreeing POS tags (*correct-all*) with a setting where the coders correct only those disagreements where at least one tagger predicted a verb tag (*verb-only*).

Annotation tool We support the correction process with a graphical user interface. The tool displays the whole sentence in a one-sentence-per-line format as well as each word token in a separate line. Further columns show the output of each tagger. Word tokens where the taggers disagree are highlighted in red. The annotator is presented with a list of candidate tags, where the most probable analysis is ranked highest. We record annotation time needed for correcting each individual POS tag, as well as the total time needed for correction.

gold		predicted	rf	tree	stanford
V*FIN	⇔	V*INF	7	8	23
ADJD	⇔	ADV	6	8	5
ADJD	⇔	VVPP	1	2	5
APPR	⇒	KOKOM	1	14	0
KOKOM	⇒	APPR	1	0	3
KON	⇒	ADV	1	7	1
ADV	⇒	KON	0	1	0
PDS	⇒	ART	0	0	4

TABLE 4 Errors made by the different taggers (for information on the tagset see Appendix); the arrows mark the direction of the erroneous prediction.

Time requirements for annotating in each setting First we want to know how much time can be saved when correcting (predicted) verb tags only, as compared to a setting where all POS tag mismatches are corrected. Table 5 shows that in the *verbs-only* setting the time needed for correcting 1500 sentences substantially decreases from 186.6 min (11.198 sec) to 54 min (3.243 sec). Also, the average time per tag is shorter than in the *correct-all* setting.

batch	setting	no. sent	no. #token corrected	time total avg.	time per tag avg.	coder1	coder2
1,2,5	*correct-all*	1500	1884	11198.02	6.25	6.16	6.35
3,4,6	*verb-only*	1500	587	3242.61	5.56	5.84	5.28

TABLE 5 Time requirements (sec.) for correcting POS tags in each setting: all (correct all POS tags where taggers disagree), verb only (correct only those where at least one of the taggers predicted a verb tag)

Impact on parsing accuracy Next we parsed the two subsets of the Falko200 (L1, L2), providing the parser with POS tags from the Stanford tagger, the TreeTagger, the RFTagger, and the tags manually corrected by our two annotators (*verb only*, *correct-all*). Table 6 shows PARSEVAL scores for the L1 and L2 data. When GFs are included, the results are around 14% higher but show the same trend. Using a dependency parser (McDonald et al., 2006) trained on the same data converted to dependencies along the lines of Foth (2006) also gives us the same basic picture.

We observed a statistically significant difference between using the POS tags predicted by the Stanford tagger and the TreeTagger as input for parsing, as compared to using gold POS tags.[3] For all other settings, the difference in results was not statistically significant. This means that the quality of the POS tags assigned by the RFTagger is already that high that further manual correction does not have a significant impact on parsing (at least on our small test sets). While results were not statistically significant, there are cases where the corrected POS tags result in the proper analysis, while the automatic POS tags cause severe errors (figure 1). Furthermore, our results show that the *verb-only* setting, while obtaining a reduction in annotation time of around one third, does yield results in the same range as the *correct-all* setting.

[3]For significance testing we use the Randomized Parsing Evaluation Comparator provided by Dan Bikel (`http://www.cis.upenn.edu/~dbikel/software.html#comparator`)

L1	prec	rec	f-sc.	tag acc
	L1 – tagger-assigned POS tags			
stanf.	73.5***	74.0***	73.8	97.2
tree	75.5**	75.4**	75.4	98.0
rf	77.1 .	76.7	76.9	98.8
	L1 – parser-assigned POS tags			
berkley	**77.9**	**77.6**	**77.8**	98.2
	L1 – manually corrected POS tags			
A1(vo)	77.4	76.9	77.1	99.2
A2(vo)	77.8	77.5	77.7	99.9
A1(all)	77.5	76.9	77.2	99.3
A2(all)	77.4	77.1	77.2	99.6
gold	**77.9**	77.5	77.7	100.0

L2	prec	rec	f-sc.	tag acc
	L2 – tagger-assigned POS tags			
stanf.	75.3***	77.1***	76.2	96.4
tree	76.2***	77.3***	76.7	97.8
rf	79.6	80.6	80.1	98.9
	L2 – parser-assigned POS tags			
berkley	80.0	80.6	80.3	97.7
	L2 – manually corrected POS tags			
A1(vo)	**80.5**	**81.0**	**80.8**	99.4
A2(vo)	80.4	**81.0**	80.7	99.9
A1(all)	80.1	80.7	80.4	99.3
A2(all)	79.7	80.6	80.1	99.6
gold	80.3	80.9	80.6	100.0

TABLE 6 Parsing results (Parseval precision, recall, f-score and tag accuracies) for Falko200 for the different POS correction settings, excluding GF from the evaluation (asterisks indicate significant differences between a correction setting and gold: p=0.001***, p=0.005**, p=0.01*, p=0.05 .)

Despite the higher tag accuracy for the RFTagger the parser benefits more when using its own POS tags (77.8 vs. 76.9% f-score for L1 and 80.3 vs. 80.1% f-score for L2). This observation is slightly puzzling and becomes more profound when comparing the Berkeley results with the TreeTagger. POS accuracy for the parser-assigned POS tags is more or less the same as for the TreeTagger, but parsing results are 2.4% (L1) and 3.6% (L2) higher when we let the Berkeley parser predict its own POS tags. This clearly shows that the overall accuracy is not enough to predict parsing scores, but that particular error types are more harmful for parser performance than others.

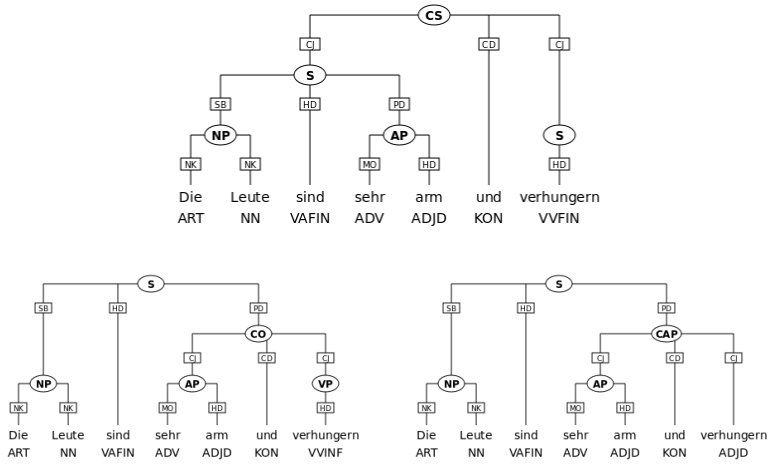

FIGURE 1 Analyses for manually corrected (top), TreeTagger-assigned (left) and Berkeley-assigned POS tags (right). (*The people are very poor and starve to death.)* FalkoEssayL2V2_0:sa007_2006_09

Error analysis We compared the POS tags assigned by the TreeTagger to the L2 part of Falko200 with those predicted by the parser. The Berkeley parser and the TreeTagger both assign nearly the same number of incorrect tags (Tree: 42, Berkeley: 43), and thus have the same average POS tag accuracy on the L2 sentences. The errors, however, are very different. There is an overlap of 5 errors only, all other erroneous predictions are unique to one of the taggers.

The most frequent error made by the Berkeley parser is to confuse past participles and adjectives (8 occurences), a distinction which was hard also for our human annotators. This, however, does not have an impact on the syntactic structure, in contrast to some of the errors frequently made by the TreeTagger, such as annotating particles with adjectives or adverbs (PTKA) as a preposition merged with a determiner (APPRART), or mistaking the adjective for a substitutive personal pronoun (PIS). Other errors resulting in erroneous syntactic structures are caused by mistaking finite verb forms for non-finite forms, an error which is also more frequent for the TreeTagger. In conclusion, both the Berkeley parser and the TreeTagger assign more or less the same number of incorrect tags, but the errors made by the TreeTagger have a greater impact on the syntactic structure of the trees, which explains the difference of 2.4% (L1) and 3.6% (L2) f-score for both settings.

4 Conclusions

In this paper we presented one step on the way towards a syntactic analysis of learner data, with a focus on semi-automatic correction of POS tags to improve statistical parsing. We showed that the most important step to obtain correct POS tags for a syntactic analysis is the formulation of a target hypothesis. It might be argued that in doing so we lose the insights about learner patterns provided by a multi-tier system like the one discussed in Diaz-Negrillo et al. Since we have the direct comparison between the TH and the learner utterance, however, we can identify those word forms where the learner utterance differs from the TH. We can then add annotation layers that specify the form or cause of deviation (orthography, inflection, etc.; compare Reznicek et al., submitted).

We presented experiments assessing the impact of POS accuracy on constituency parsing, and compared different strategies for manual correction, supported by an annotation tool which tells the annotator which tags to correct, based on the (disagreeing) predictions of different POS taggers. We showed that correcting only those POS tags where one of the taggers had assigned a verb results in considerable time savings and, at the same time, does not cause a significant decrease in parsing accuracy. For our data, however, the quality of the POS tags assigned by the RFTagger as well as for the ones predicted by the Berkeley parser is good enough to make a manual correction superfluous.

Our results showed that the accuracy of the POS tags is not sufficient to predict parsing performance, but that much depends on the particular error types made by the POS tagger. This finding should have consequences for POS tagger evaluation, as it shows that the average accuracy (as measured against a goldstandard) is not sufficient for a meaningful comparison of the performance of different taggers.

The lessons learned are the following. While it is worthwhile to create target hypotheses and use those for the automatic analysis of learner language, no further improvements are to be expected for the manual correction of automatically assigned POS tags, given that the accuracy of the POS is as high as 98%. In future work we want to explore the adequacy of dependency representations for analysing learner data.

Acknowledgments

Part of this work was supported by a grant from the DFG awarded to SFB 632. We want to express our gratitude to Yannick Versley for his assistance with the dependency conversion and to Amir Zeldes and the three anonymous reviewers for insightful comments and suggestions.

References

Ambati, Bharat Ram, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010. Two methods to incorporate 'local morphosyntactic' features in hindi dependency parsing. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 22–30. Los Angeles, CA.

Dandapat, Sandipan, Priyanka Biswas, Monojit Choudhury, and Kalika Bali. 2009. Complex linguistic annotation – no easy way out! A case from Bangla and Hindi POS labeling tasks. In *Proceedings of the Third Linguistic Annotation Workshop*, LAW '09, pages 10–18. Suntec, Singapore.

Díaz-Negrillo, Ana, Detmar Meurers, Salvador Valera, and Holger Wunsch. 2010. Towards interlanguage pos annotation for effective learner corpora in SLA and FLT. *Language Forum* 36(1–2):139–154.

Dickinson, Markus and Chong Min Lee. 2009. Modifying corpus annotation to support the analysis of learner language. *CALICO Journal* 26(3):545–561.

Dickinson, Markus and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '03, pages 107–114. Budapest, Hungary.

Dickinson, Markus and Marwa Ragheb. 2009. Dependency annotation for learner corpora. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT-8)*, pages 59–70. Milan, Italy.

Doolittle, Seanna. 2008. *Entwicklung und Evaluierung eines auf dem Stellungsfeldermodell basierenden syntaktischen Annotationsverfahrens für Lernerkorpora innerhalb einer Mehrebenen-Architektur mit Schwerpunkt auf schriftlichen Texten fortgeschrittener Deutschlerner*. Master's thesis, Humboldt University, Berlin.

Foth, Kilian. 2006. Eine umfassende Constraint-Dependenz-Grammatik des Deutschen. Tech. rep., Universität Hamburg.

Granger, Sylviane. 2008. Learner corpora. In A. Lüdeling and M. Kytö, eds., *Corpus Linguistics. An International Handbook*, pages 259–275. Berlin: Mouton de Gruyter.

Granger, Sylviane., Joseph Hung, and Stephanie Petch-Tyson. 2002. *Computer learner corpora, second language acquisition, and foreign language teaching*. Language learning and language teaching. Amsterdam/Philadelphia: John Benjamins.

Haan, Pieter de. 2000. Tagging non-native english with the TOSCA-ICLE tagger. In C. Mair, ed., *Corpus linguistics and linguistic theory: Papers from the twentieth International Conference on English Language Research on Computerized Corpora (ICAME 20), Freiburg im Breisgau 1999*, vol. 33 of *Language and computers*, pages 69–79. Amsterdam: Rodopi.

Hirschmann, Hagen, Seanna Doolittle, and Anke Lüdeling. 2007. Syntactic annotation of non-canonical linguistic structures. In

Proceedings of Corpus Linguistics 2007. Birmingham, UK. http://ucrel.lancs.ac.uk/publications/CL2007/paper/128_Paper.pdf.

Hirschmann, Hagen, Anke Lüdeling, Ines Rehbein, Marc Reznicek, and Amir Zeldes. to appear. Underuse of syntactic categories in Falko. A case study on modification. In S. Granger, G. Gilquin, and F. Meunier, eds., *20 years of learner corpus research: looking back, moving ahead*. Louvain-la-Neuve, Belgium: Presses universitaires de Louvain.

Knobloch, Clemens and Burkhard Schaeder. 2000. Kriterien für die Definition von Wortarten. In G. Booij, C. Lehmann, and J. Mugdan, eds., *Morphologie - Ein internationales Handbuch zur Flexion und Wortbildung*, pages 674–692. Berlin, New York: Walter de Gruyter.

Krivanek, Julia and Detmar Meurers. 2011. Comparing rule-based and data-driven dependency parsing of learner language. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, pages 128–132. Barcelona, Spain.

Kübler, Sandra. 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *Proceedings of Recent Advances in Natural Language Processing*, RANLP '05, pages 293–300. Borovets, Bulgaria.

Lüdeling, Anke. 2008. Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora. In M. Walter and P. Grommes, eds., *Fortgeschrittene Lernervarietäten*, pages 119–140. Niemeyer, Tübingen.

Lüdeling, Anke. 2011. Corpora in linguistics: Sampling and annotation. In K. Grandin, ed., *Going Digital. Evolutionary and Revolutionary Aspects of Digitization. [Nobel Symposium 147]*, pages 220–243. Science History Publications/USA, New York.

Lüdeling, Anke, Seanna Doolittle, Hagen Hirschmann, Karin Schmidt, and Maik Walter. 2008. Das Lernerkorpus Falko. *Deutsch als Fremdsprache* 2:67–73.

Marton, Yuval, Nizar Habash, and Owen Rambow. 2010. Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 13–21. Los Angeles, CA.

McDonald, Ryan, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 216–220. New York City, New York.

Meunier, Fanny and Inge de Mönnink. 2001. Assessing the success rate of EFL learner corpus tagging: Online abstract. In *ICAME 2001 Future Challenges in Corpus Linguistics*. http://cecl.fltr.ucl.ac.be/Events/icamepr.htm.

Petrov, Slav and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411. Rochester, NY.

Petrov, Slav and Dan Klein. 2008. Parsing German with latent variable grammars. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 33–39. Columbus, Ohio.

Rafferty, Anna N. and Christopher D. Manning. 2008. Parsing three German treebanks: lexicalized and unlexicalized baselines. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 40–46. Columbus, Ohio.

Reznicek, Marc, Maik Walter, Karin Schmidt, Anke Lüdeling, Hagen Hirschmann, Cedric Krummes, and Torsten Andreas. 2010. *Das Falko-Handbuch: Korpusaufbau und Annotationen*. Institut für deutsche Sprache und Linguistik, Humboldt-Universität zu Berlin, Berlin.

Rosén, Victoria and Koenraad De Smedt. 2010. Syntactic annotation of learner corpora. In H. Johansen, A. Golden, J. E. Hagen, and A.-K. Helland, eds., *Systematisk, variert, men ikke tilfeldig*, pages 120–132. Novus forlag.

Schiller, Anne, Simone Teufel, and Christine Thielen. 1995. Guidelines für das Tagging deutscher Textkorpora mit STTS. Tech. rep., Universität Stuttgart, Universität Tübingen.

Schmid, Helmut. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04. Geneva, Switzerland.

Schmid, Helmut and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 777–784. Manchester, UK.

Seeker, Wolfgang, Ines Rehbein, Jonas Kuhn, and Josef van Genabith. 2010. Hard constraints for grammatical function labelling. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10. Uppsala, Sweden.

Tenfjord, Kari, Jon Erik Hagen, and Hilde Johansen. 2006. The hows and whys of coding categories in a learner corpus (or How and why an error-tagged learner corpus is not ipso facto one big comparative fallacy). *Rivista di Psicolinguistica Applicata (RiPLA)* 3:93–108.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180. Edmonton, Canada.

van Rooy, Bertus and Lande Schäfer. 2002. The effect of learner errors on pos tag errors during automatic POS tagging. *Southern African Linguistics & Applied Language Studies* 20(4):325–335.

Wierzbicka, Anna. 2000. Lexical prototypes as a universal basis for crosslinguistic identification of parts of speech. In P. Vogel and B. Comrie, eds., *Approaches to the Typology of Word Classes*, pages 285–318. Berlin, New York: Mouton de Gruyter.

Appendix

STTS POS tags

Label	Description
ADJA	attributive adjective
ADJD	adverbial or predicative adjective
ADV	adverb
APPR	preposition; circumposition left
APPRART	preposition with article
APPO	postposition
APZR	circumposition right
ART	definite or indefinite article
CARD	cardinal number
FM	foreign material
ITJ	interjection
KOUI	subordinating conjunction with “zu” and infinitive
KOUS	subordinating conjunction with clause
KON	coordinating conjunction
KOKOM	compative conjunction
NN	noun
NE	proper name
PDS	substitutive demonstrative pronoun
PDAT	attributive demonstrative pronoun
PIS	substitutive indefinite pronoun
PIAT	attributive indefinite pronoun without determiner
PIDAT	attributive indefinite pronoun with determiner
PPER	irreflexive personal pronoun
PPOSS	substitutive possessive pronoun
PPOSAT	attributive possessive pronoun
PRELS	substitutive relative pronoun
PRELAT	attributive relative pronoun
PRF	reflexive personal pronoun

PWS	substituive interrogative pronoun
PWAT	attributive interrogative pronoun
PWAV	adverbial interrogative or relative pronoun
PROAV	pronominal adverb
PTKZU	"zu" before infinitive
PTKNEG	negation particle
PTKVZ	separated verb prefix
PTKANT	answer particle
PTKA	particle with adjective or adverb
TRUNC	initial constituent of a compound
VVFIN	finite verb, full
VVIMP	imperative, full
VVINF	infinitive, full
VVIZU	infinitive with "zu", full
VVPP	perfect participle, full
VAFIN	finite verb, aux
VAIMP	imperative, aux
VAINF	infinitive, aux
VAPP	perfect participle, aux
VMFIN	finite verb, modal
VMINF	infinitive, modal
VMPP	perfect participle, modal
XY	non-word, with special characters
$,	comma
$.	sentence-final punctuation
$(	clause-internal punctuation

TABLE 1: Stuttgart-Tübingen-POS-Tags (STTS)