Towards Replicability in Parsing

Daniel Dakota, Sandra Kübler
Indiana University
{ddakota, skuebler}@indiana.edu

Abstract

We investigate parsing replicability across 7 languages (and 8 treebanks), showing that choices concerning the use of grammatical functions in parsing or evaluation and the influence of the rare word threshold, as well as choices in test sentences and evaluation script options have considerable and often unexpected effects on parsing accuracies. All of those choices need to be carefully documented if we want to ensure replicability.

1 Introduction

Over the last 10 years, statistical constituent parsing has developed from a research area that mainly focused on parsing the Penn Treebank (Marcus et al., 1993) to covering more languages, including a wide range of morphologically rich languages, such as German or Arabic. While the extension across different languages resulted in many insights about parsing models for specific languages, the development also had a dramatic effect on replicability. For novices to become experts in parsing, they need not only understand the parsing algorithms and implementations, but must also become familiar with the intricacies of existing data sets and annotations across a range of languages. It should be a fairly well known fact that the Penn Treebank uses traces in form of numbers attached to constituent labels as well as empty categories. The failure to remove either traces or empty categories before training can lead to rather unexpected, low results, especially when the test data come from a different source and do not have empty categories in the sentences. After a careful study of the existing literature, we may also know that the German TiGer treebank (Brants et al., 2004) uses crossing branches, which need to be resolved before parsing with a CFG parser, and that the method with which the crossing branches are resolved has an influence on parsing results (Boyd and Meurers, 2008). But it may not be generally known that the two major German treebanks, TiGer and TüBa-D/Z (Telljohann et al., 2015), do not attach punctuation signs into the tree structure, which can lead to unexpected results if not handled in preprocessing. Again, the method chosen to attach punctuation signs has an effect on parsing accuracy (p.c. W. Maier).

The problem becomes more serious if we conduct experiments across a wider range of languages, some of which we may not be familiar with, in order to demonstrate good parser performance across languages. It is exacerbated by often minimal descriptions of the methods used in preprocessing for published results. We argue that in order to enable replicability in parsing research, we need to be more explicit and detailed with regard to pre- and post-processing steps.

In the current paper, we show the effects of different decisions in two major design issues on parsing results: We investigate two issues during parsing and during evaluation. In parsing, we 1) have a closer look at the (non-)use of grammatical functions during training and testing, and 2) we investigate the effect of selecting the threshold for rare words. Grammatical functions (GFs) are part of most constituent annotation schemes (such as NP-SBJ for subjects or NP-TMP for temporal NPs in the Penn Treebank). Since parsers often delete these GFs by default in their output, it may not be immediately obvious that using them in training has an effect on the quality of the results. The threshold for rare words is often neglected in parsing experiments, but has a considerable influence on results. With regard to evaluation, we investigate decisions in the test set and in the evaluation parameters. More specifically, we investigate 1)

Proceedings of Recent Advances in Natural Language Processing, pages 185–194,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_026

the effects of test set size, which varies across languages. 2) We also show evaluation results using different settings of the evaluation script.

Since all of those decisions have an effect on parsing results, we argue that they need to be documented in parsing literature to ensure replicability of parsing results. Whereas in the past, concern about paper length forced authors to focus on important findings and conclusions, which resulted in the omission of many details, today this is not necessarily an issue as many conferences allow the inclusion of supplementary information for exactly such purposes as well as the ability to make publicly available any additional settings.

2 State of the Art

Replicability has started to appear as a topic in NLP and machine learning, for example as an IJCAI 2015 Workshop on Replicability and Reproducibility in Natural Language Processing[1], and it has been described as one of the potentially negative factors of shared tasks in NLP by Parra Escartín et al. (2017). However, there is little work on more specific areas such as parsing.

In many cases, decisions concerning preprocessing, parser settings, or evaluation settings are not described in parsing literature, thus requiring assumptions ranging from what function tags to be included in evaluation (Gabbard et al., 2006) to the exact specifications in accompanying evaluation parameter files (van Cranenburgh and Bod, 2013).

In the following, we look at a few highly cited papers to document common practices in the literature. It is not our aim to single out those authors, but we rather want to document that even high quality papers do not provide enough information for replicability, thus documenting the need for more rigorous guidelines.

2.1 Preprocessing

For German, Cheung and Penn (2009) report that they could not replicate experiments by Becker and Frank (2002): "While the test set used in the paper [by Becker and Frank] was manually corrected for evaluation, we did not correct our test set, because it would be difficult to ensure that we adhered to the same correction guidelines. No details of the correction process were provided in the paper, ... Also, because we could not obtain the exact sets used for training, development, and testing, we had to recreate the sets by randomly splitting the corpus."

For English, Bod (2001) describes his preprocessing of the WSJ part of the Penn Treebank as follows: "All trees were stripped off their semantic tags, co-reference information and quotation marks." No further information is given. Charniak and Johnson (2005) describe their data split as "We used the division into preliminary training and preliminary development data sets described in (Collins, 2000)" but do not mention any preprocessing. Collins (2000) similarly defines his data split, but does not mention preprocessing. In their work on domain adaptation, McClosky et al. (2006) describe the splits they use for the Brown Corpus and the WSJ portion of the Penn Treebank, but they do not mention any preprocessing. Klein and Manning (2003) describe their use of the WSJ sections as: "For each model, input trees were annotated or transformed in some way, as in (Johnson, 1998)". The latter describes his tree transformations in detail, but only mentions the data splits without any details about preprocessing: "It is fairly straightforward to mechanically transform the Penn II tree representations in the WSJ corpus into something close to the alternative tree representations described above ...". In a multilingual setting, the only information Petrov and Klein (2007) provide concerns information about their treatment of the treebanks: "We trained models for English, Chinese and German using the standard corpora and splits as shown in Tab. 3. We applied our model directly to each of the treebanks, without any language dependent modifications." There is no information about preprocessing. And apart from the cross-references to previous papers shown in the quotes, there is no information about possible standards documented in earlier publications.

2.2 Evaluation

Johnson (1998) defines precision and recall, but does not provide information about implementation or parameters. Charniak and Johnson (2005) state that they "evaluated the performance of [their] reranking parser using the standard PARSEVAL metrics." (Bod, 2001) describes his evaluation as "We used 'evalb' to compute the standard PARSEVAL scores for our parse results. We

[1] https://sites.google.com/site/adaptivenlp2015/

	# train	# test	#POS	# GFs
Arabic (AR)	5k	1,958	35	20/52
Basque (EU)	5k	946	25	5
Chinese (ZH)	5k	1,905	33	26
English (EN)	5k	2,412	36	20
German (DE-TI)	5k	5,000	51	51
German (DE-Tü)	5k	5,000	54	51
Hebrew (HE)	5k	716	50	45
Swedish (SV)	5k	666	25	65

Table 1: Treebank Statistics.

focus on the Labeled Precision (LP) and Labeled Recall (LR) scores only in this paper, as these are commonly used to rank parsing systems." but does not provide information about parameter settings. McClosky et al. (2006) "We use evalb to calculate how similar the two sets of output are on a bracket level." Neither Collins (2000), Klein and Manning (2003) nor Petrov and Klein (2007) mention the evaluation script they use.

3 Experimental Setup

3.1 Treebanks

We focus on a range of languages, covering morphologically rich languages with different characteristics (Arabic, Basque, German, Hebrew, Swedish) and morphologically simpler languages (Chinese, English). For Arabic (Green and Manning, 2010), Basque (Aldezabal Roteta et al., 2008), Hebrew (Sima'an et al., 2001), the German TiGer treebank (Brants et al., 2004), and Swedish (Nivre and Megyesi, 2007), we use data from the 2013 and 2014 shared tasks on parsing MRLs (see (Seddah et al., 2013, 2014) for a description of treebank preparation). In addition, we utilize the Penn Chinese Treebank `CTB5` (Xue et al., 2005), the Penn Treebank of English (Marcus et al., 1993), and the German TüBa-D/Z treebank (Telljohann et al., 2015).

Table 1 lists the sizes of the training and test sets in terms of number of sentences, and it lists the number of POS tags and function labels. For Arabic, we list the number of single function labels and the number of the combined ones. Since the treebanks vary considerably in size, we used the small SPMRL data sets of 5,000 sentences. Basque and Swedish are by far the smallest data sets, which also affects the size of the test set. However, these two languages also show extremes in terms of grammatical functions: With 5 GFs, Basque has the lowest number, and Swedish the highest with 65.

It is common knowledge that the size of the tagset as well as the size of the GFs influences parsing results. However, it is difficult to separate effects of the POS tagset and the GFs from language characteristics. For this reason, the SPMRL shared tasks have concentrated on an evaluation that disregards the GFs for the inter-language comparison (Seddah et al., 2013, 2014).

3.2 Parser Setup

We use the Berkeley parser (Petrov and Klein, 2007), which has achieved state of the art results for a range of languages (Seddah et al., 2013, 2014). In order to ensure stable findings, we report results averaged over four train/test cycles using grammars trained with four different seeds. To ensure comparability, we use the same four seeds (1-4) for the grammar across all languages.

For all languages, we follow SPMRL and train on a subset of 5,000 sentences in order to eliminate effects of training set size. This size was chosen for ease of comparison to the SPMRL shared task and since it is the maximum training size available for the Swedish Treebank. We use gold POS tags for training and report results for both gold POS tags and parser internal tagging. Pre-processing for the English and Chinese treebanks involved removing traces and collapsing unary nodes into a single node using the Berkeley Parser Analyser (Kummerfeld et al., 2012, 2013).

3.3 Evaluation

We evaluate using the scorer for the SPMRL 2014 shared task[2], a reimplementation of EVALB[3] that allows for additional options such as scoring with grammatical functions and penalizing unparsed sentences. By default, EVALB only scores up to a *dash* that separates any complex label (e.g. SBJ-TMP) and inherently does not score GFs given this behavior. Furthermore, EVALB does not penalize unparsed sentences, and simply omits them from the final scores, reporting the number of unparsed sentences separately. This can be somewhat misleading, especially in cases of high numbers of unparsed sentences.

The average scores across the four grammars are reported with punctuation being scored, which

[2] `http://spmrl.org/spmrl2014-sharedtask`
[3] `http://nlp.cs.nyu.edu/evalb/`

is not standard practice for languages such as English, but whose inclusion influences parsing results (Kulick et al., 2006).

Results in section 4 and 5.1 are evaluated with a parameter file that deletes root nodes and accompanying brackets from gold and parsed files. Some treebanks contain additional root nodes that are not intrinsically part of the treebank, but are required for parsing (such as TiGer in which all punctuation is attached to a virtual root). However, the inclusion of these roots should not be included in evaluation as this can artifically increase scores. Additionally we enable settings that penalizes the parser for unparsed sentences and and score the POS tag but not the function labels (unless mentioned otherwise).

4 Parsing Decisions

In this section, we investigate decisions regarding grammatical functions (GFs) in parsing and/or evaluation as well as the influence of the rare word threshold on parsing results.

4.1 Grammatical Functions

We first investigate how decisions regarding the use of GFs influence parsing results. For each language, there are three settings: 1) NoGF: The complete experiment (training/parsing/evaluation) is carried out without grammatical functions. 2) Mixed: In this setting, we train and parse using GFs, but we ignore them in evaluation. I.e., the evaluation only considers the constituent structure, which is the standard setting for parsing English and has also been used for the SPMRL shared tasks even if parsers produced GFs. 3) AllGF: Here, the complete experiment, including evaluation, uses GFs. For the versions of trees without GFs, we remove GFs from the node to which they are attached. For example, NP-SBJ is shortened to NP.

The third setting may be a harsh evaluation for configurational languages, but for non-configurational languages, the GFs are the only indicators of subjects and direct objects and thus important information for many downstream applications.

The results of these experiments are shown in table 2.

POS tagging considerations The POS tagging results show several noteworthy trends. The first issue is that even in the setting where we provide gold POS tags, the POS tagging accuracy is generally lower than 100%. In these cases, the parser changes the POS tags because it cannot find a good analysis using the gold POS. I.e., the Berkeley parser trades POS tagging accuracy for parsability.

Swedish and the German TiGer treebank show the most untypical results: For gold POS, both reach (near-)perfect accuracy when no GFs are used, but when GFs are used in training/parsing or additionally for evaluation, the accuracy degrades to 92.58% and 89.66% for Swedish and to 90.16% and 86.25% for TiGer. The reason can be attributed to the fact that both the Swedish treebank and TiGer attach GFs to POS tags. A comparison of POS tagging accuracy between setting 1+2 (noGF/mixed) and 2+3 (mixed/allGF) for both treebanks shows that the former difference is twice as large as the latter. I.e., the parser more often changes from one POS tag to another rather than keeping the same POS tags and only changing the GF. When parsing with GFs, both parser internal and gold POS tags result in (near-)identical performance, suggesting that the parser is completely retagging all words to parse. A similar phenomenon has been observed by Maier et al. (2014) for German.

The most surprising results can be found in the comparison of not using GFs at all and using them for parsing but ignoring them in evaluation. For gold POS, we see almost identical results, but for automatically assigned POS tags, Arabic shows the smallest loss (0.52) and Swedish the highest (2.53), followed by German-TiGer (2.42). It is worth noting that the treebanks that attach GFs to POS tags, Swedish and TiGer, have the highest differences. These results show that using GFs internally has a noticeable negative effect on accuracy.

In conclusion, the decision whether to use GFs in parsing and evaluation has a considerable effect on POS tagging quality, which goes against our original expectations.

Parsing considerations When we look at the F-scores in table 2, we also see the expected decrease from parsing without GFs (noGF) to parsing and evaluation including GFs (allGF). There is a considerable decrease in the F-scores, ranging from 4.82 points (Basque) to >10 for Arabic, English, German, and Swedish, given gold POS tags. Thus, the divide does not correlate with morphological

		Gold POS				POS by parser			
Language	GFs	F-Score	Recall	Precision	POS	F-Score	Recall	Precision	POS
Arabic	noGF	76.82	75.98	77.66	99.98	73.88	72.73	75.06	94.24
	parse	74.19	73.94	74.44	99.93	71.63	70.95	72.32	93.72
	allGF	65.54	65.32	65.76	99.93	63.01	62.42	63.62	93.72
Basque	noGF	74.11	73.90	74.33	98.15	66.99	66.27	67.72	87.70
	mixed	72.57	72.12	73.04	98.14	65.35	64.22	66.51	87.10
	allGF	69.29	68.85	69.73	98.14	62.65	61.57	63.76	87.10
Chinese	noGF	83.52	83.18	83.87	99.94	72.43	71.08	73.82	88.37
	mixed	81.28	81.87	80.69	99.82	70.14	69.70	70.58	87.42
	eval	75.56	76.12	75.01	99.82	64.25	63.85	64.66	87.42
English	noGF	84.38	84.37	84.39	99.76	83.79	83.56	84.01	95.70
	mixed	82.30	82.69	81.90	99.72	81.86	81.91	81.80	95.12
	allGF	73.95	74.31	73.60	99.72	73.72	73.77	73.67	95.12
German-Ti	noGF	71.96	70.50	73.48	99.64	69.27	66.97	71.73	92.52
	mixed	65.72	64.34	67.18	90.16	65.73	64.34	67.17	90.10
	allGF	49.90	48.84	51.00	86.25	49.90	48.84	51.00	86.19
German-Tü	noGF	91.78	91.85	91.70	99.91	87.52	87.32	87.73	94.47
	mixed	90.70	91.01	90.41	99.82	85.88	85.92	85.85	93.53
	allGF	79.16	79.42	78.90	99.82	75.02	75.05	74.98	93.53
Hebrew	noGF	91.97	92.07	91.87	100.00	86.87	86.85	86.89	92.20
	mixed	89.92	90.14	89.70	100.00	84.58	84.70	84.45	91.30
	allGF	83.72	83.93	83.51	99.96	78.53	78.65	78.41	91.30
Swedish	noGF	82.88	82.66	83.10	100.00	79.06	78.73	79.39	95.11
	mixed	75.21	74.92	75.49	92.58	75.21	74.92	75.49	92.58
	allGF	63.22	62.97	63.46	89.66	63.22	62.97	63.46	89.66

Table 2: Results for training/parsing without GFs (noGF), training/parsing with GFs + evaluation w/o GFs (mixed), and training/parsing/evaluation with GFs (allGF).

richness, as we would have expected. The divide is unclear, but may partially depend on the annotation scheme or the number of GFs. However, further research is required to investigate this point. For Swedish and TiGer, the decrease is extreme, reaching almost 20 points for Swedish and >22 points for TiGer. This can be explained by the complete retagging (see above). I.e., the parser's performance is on a level of internally assigned POS tags. For automatic POS tags, the differences are similar or smaller.

However, there is a somewhat surprising decrease in F-scores between the setting in which we train/parse without GFs (noGF) and the setting where we use GFs in training/parsing but ignore them in evaluation (mixed). One would expect the presence of GFs to have only a minimal effect on phrase structure decisions. But results show decreases in the F-score, generally between 1.08 (German-TüBa-D/Z) and 2.63 (Arabic) for gold POS. Exceptions are Swedish with a drop of 7.67 points and German-Tiger with a decrease of 6.24 points. The drop is similar or less pronounced for parser internal POS tags.

In conclusion, we see several expected differences between the settings. But we also see unexpected differences, for example in the comparison of results between using or not using GFs in parsing if we do not consider them in the evaluation. Thus, if we do not delete GFs from the treebank data before we parse, we will obtain artificially low results, even when GFs are not part of the evaluation.

4.2 Handling Rare Words

The Berkeley parser's handling of rare words has been demonstrated to be biased towards an English lexicon (Hall et al., 2014). Rare words are mapped to classes based on automatically learned suffix and character information, as described by Petrov et al. (2006). During training, the parser performs smoothing over rare words, with a de-

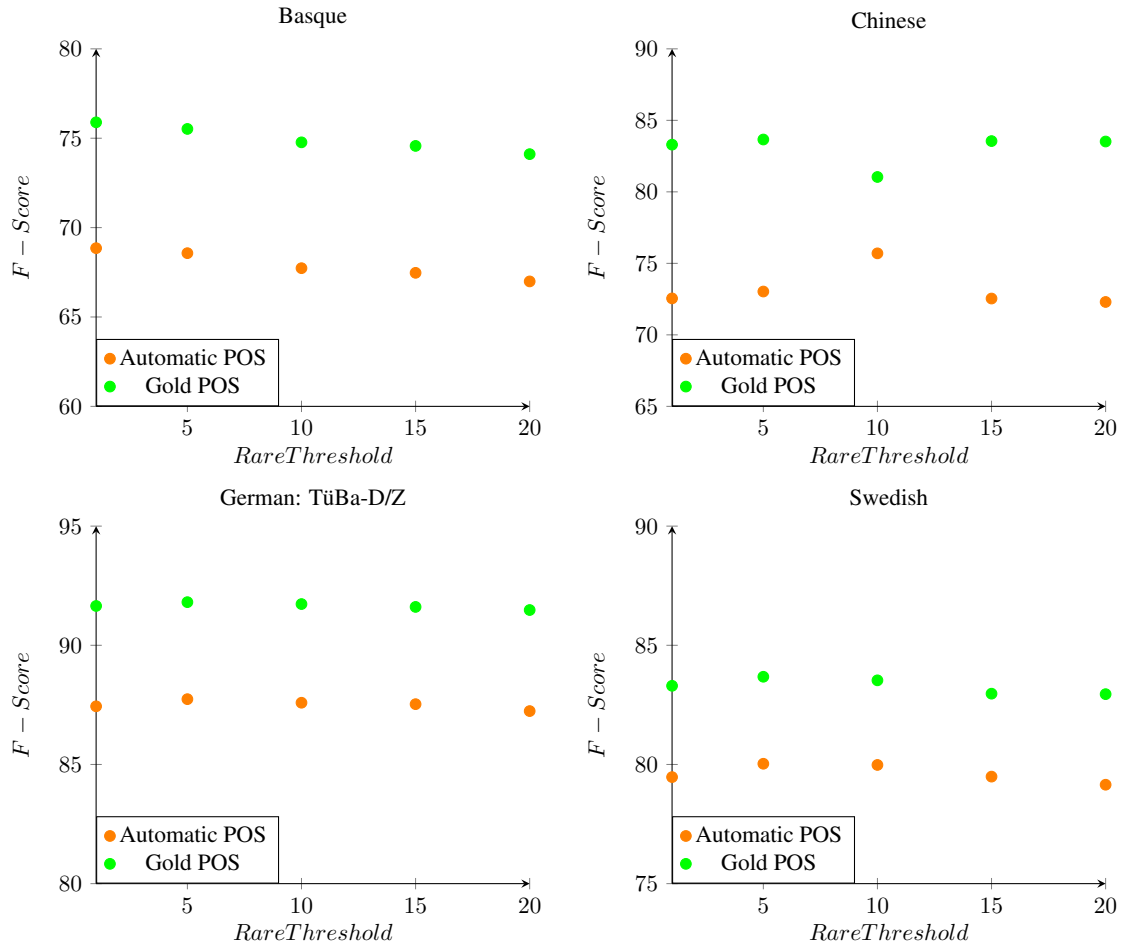

Figure 1: Experiments with different threshold for handling rare words.

fault threshold of 10. Some languages, such as Chinese, have additional lexicon information accessible to the parser (which we do not use here), but for most languages this is not the case. Yet when using the parser in a downstream application, the threshold for rare words at which the parser was trained has a direct impact on performance, particularly for languages with small training sets, but is often not adjusted.

The results of using different thresholds for rare words in the Berkeley parser are shown in figure 1. These experiments do not use grammatical functions. We show the results of four languages, Basque, Chinese, German using the TüBa-D/Z treebank, and Swedish.

The German results show only minimal differences in F-scores of around half a point. For Swedish, there is a difference of approximately 1 point. However, in both cases, the best setting is not the default setting of 10, but rather a lower threshold closer to 5. For Basque, the lowest setting of 1 provides the highest F-scores, and the difference to other settings reaches approximately 2 points. Given that differences in parsing results in the range of 0.5 points tend to be significant, this means that the differences are in the range of statistical significance.

The most interesting results, however, come from Chinese: In contrast to all other languages, in Chinese, we see a difference between the best threshold for the setting where we use gold POS tags as input for the parser, as opposed to having the parser assign POS tags internally. When using gold POS tags, a threshold of 10 gives us the lowest F-scores while for the automatic POS tags, it is the best setting. Additionally, the differences between settings are the most pronounced for Chinese, reaching more than 3 points (between 72.55 and 75.70 for automatic POS tags). The reasons for this behavior are unclear.

These results show very clearly that it is important to optimize the threshold for rare words, not only per language, but also per setting (gold POS tags versus automatic POS tags). And in order

Lang.	F_1	F_2	F_3	F_4	$diff_{max}$	# $test_{orig}$
Arabic	62.33	62.21	62.14	62.43	0.29	1,958
Basque	63.11	62.85	63.33	63.27	0.48	946
Chinese	64.86	63.29	63.97	64.42	1.57	1,905
English	73.36	72.70	73.88	73.68	1.18	2,412
German-TiGer	51.03	52.06	51.24	49.76	2.30	5,000
German-TüBa-D/Z	73.86	75.30	73.80	73.53	1.77	5,000
Hebrew	77.74	78.25	78.74	77.73	1.01	716
Swedish	63.47	63.54	62.48	63.01	1.06	666

Table 3: Parsing results based on sampled test sets of size 500.

to ensure replicability, it is important to document these parameter settings.

5 Evaluation Decisions

In this section, we have a closer look at decisions involving the evaluation step. We consider whether the size of the test set has an influence on parsing results. In other words, are some of the test sets too small to be representative? The other question concerns options that the evaluation scripts provide, such as deleting the virtual root that often serves as the unique root node that some parsers require.

5.1 Test Set Size

We investigate the effect of test set size by a controlled experiment in which we repeatedly randomly sample 500 sentences from the standard test set. The sampling is performed after parsing, which means that all the results we present are based on the same parses from section 4 (using automatic POS tags and GFs throughout). The number was chosen so that it is slightly below the size of the smallest test set (666 sentences for Swedish). We chose to repeat the sampling for each language four times. Rather than reporting the average and variance, we decided to show the actual results along with the difference between the highest and lowest result, to give the reader a better overview of the variation in results.

The results of evaluating the randomly sampled subsets are shown in table 3. They show that there are differences between 0.29 points (Arabic) and 2.30 (German-TiGer). Most of the differences are in the range between 1 and 1.8 points. Once again, we are in the range of statistical significance, and it is concerning to see that the selection of test sentences has such a large effect on parsing accuracy. It is also obvious that there is no correlation to the original size of the test set given that the two German treebanks with the highest number of test sentences have the highest variance while the smallest variance is found in a mid-sized language, Arabic, and the smallest treebank, Swedish, is on the lower end.

5.2 Evaluation Script Settings

To demonstrate the impact of evaluation decisions on parsing results, we present three evaluation settings: 1) Standard: standard EVALB metrics where there is no penalty for unparsed sentences, no inclusion of a parameter file, and no evaluation of function labels; 2) +DEL: standard EVALB metrics with a parameter file that deletes root nodes from the gold and test data and accompanying brackets; 3) +PEN: EVALB metrics with a parameter file and penalties for unparsed sentences. We show results with these settings when integrating or ignoring GFs in the evaluation.

The results are shown in table 4. The good news is that for Arabic, Chinese, English, and Swedish, there is (virtually) no difference across the different setting. But for all other languages, there are differences, and some of them are considerable. If results change in a language, they decrease from the default setting to using the parameter file to penalizing the parser for unparsed sentences. For the German TiGer treebank, we see the highest decrease, the results degrade from 69.44 to 65.73 given GFs are not evaluated, and from 55.31 to 49.90 if GFs are evaluated. This means that leaving virtual roots or similar nodes in the parses or the grammar can have a significant effect on results. The results also show that the difference between deleting virtual roots (+DEL) and punishing the parser for unparsed sentences (+PEN) is generally minimal; the only exception is Basque when GFs are ignored in evaluation. This means

	GFs in parse only			GFs in evaluation		
Lang.	standard	+DEL	+PEN	standard	+DEL	+PEN
Arabic	71.71	71.71	71.63	63.08	63.08	63.01
Basque	67.60	65.35	63.35	65.08	62.65	62.65
Chinese	72.45	72.45	72.43	53.47	53.47	53.45
English	81.89	81.89	81.86	73.75	73.75	73.72
German-TiGer	69.44	65.78	65.73	55.31	49.94	49.90
German-TüBa-D/Z	86.57	85.92	85.88	76.19	75.04	75.02
Hebrew	84.89	84.58	84.58	78.97	78.53	78.53
Swedish	75.36	75.36	75.21	63.34	63.34	63.22

Table 4: Parsing results based on different evaluation settings: +DEL = standard + parameter; +PEN = standard + parameter + parse penalty.

that the decision concerning the virtual roots often has more effect on results than the decision to punish the parser for unparsed sentences. Thus, their treatment needs to be reported to ensure replicability.

6 Conclusion and Future Work

In this paper, we have started looking into which factors can affect parsing results and replicability of results. We show that the choice of using grammatical functions in parsing and/or evaluation, the choice of the threshold for rare words, the choice of test sentences, and the choice of evaluation parameters all have considerable and potentially statistically significant effects on parsing results. Most of these details are not generally reported in research on parsing. However, only by reporting all choices meticulously, we can ensure replicability.

We have clearly just scratched the surface in this investigation. We will continue our investigation to better understand factors that influence parsing results, by having a closer look at how evaluation script settings influence parsing results, but also by investigating representational issues, such as double bracketing of sentences across different parser. We will also investigate more parsers. Preliminary results show that the LORG parser (Attia et al., 2010), a re-implementation of the Berkeley parser, shows systematic differences in how the choice of using grammatical functions and of the threshold for rare words influence parsing results.

References

Izaskun Aldezabal Roteta, Maria Jesús Aranzabe Urruzola, Arantza Diaz de Ilarraza Sánchez, and Enrique Fernández Terrones. 2008. From dependencies to constituents in the reference corpus for the processing of Basque. In *Procesamiento del Lenguaje Natural, n^o 41 (2008)*. Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN), pages 147–154.

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling Unknown Words in Statistical Latent-Variable Parsing Models for Arabic, English and French. In *Proceedings of SPRML 2010*.

Markus Becker and Anette Frank. 2002. A Stochastic Topological Parser for German. In *Proceedings of the 19th International Conference on Computational Linguistics*. Taipei, Taiwan, COLING '02, pages 1–7.

Rens Bod. 2001. What is the minimal set of subtrees that achieves maximal parse accuracy. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL) and the 10th Conference of the European Chapter of the ACL (EACL)*. pages 66–73.

Adriane Boyd and Detmar Meurers. 2008. Revisiting the impact of different annotation schemes on PCFG parsing: A grammatical dependency evaluation. In *Proceedings of the ACL Workshop on Parsing German*. Columbus, OH.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation, Special Issue* 2(4):597–620.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, MI, pages 173–180.

Jackie Chi Kit Cheung and Gerald Penn. 2009. Topological Field Parsing of German. In *Proceedings of*

the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP. Suntec, Singapore, pages 64–72.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. Stanford, CA.

Ryan Gabbard, Mitchell Marcus, and Seth Kulick. 2006. Fully parsing the Penn Treebank. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*. New York, pages 184–191.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, pages 394–402.

David Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, pages 228–237.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics* 24(4):613–632.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL-2003*. Sapporo, Japan, pages 423–430.

Seth Kulick, Ryan Gabbard, and Marcus Mitchell. 2006. Parsing the Arabic treebank: Analysis and improvements. In *Proceedings of the Treebanks and Linguistic Theories Conference*. Prague, Czech Republic, pages 31–42.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street Corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, South Korea, pages 1048–1059.

Jonathan K. Kummerfeld, Daniel Tse, James R. Curran, and Dan Klein. 2013. An empirical examination of challenges in Chinese parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 98–103.

Wolfgang Maier, Sandra Kübler, Daniel Dakota, and Daniel Whyatt. 2014. Parsing German: How much morphology do we need? In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (SPMRL-SANCL)*. Dublin, Ireland, pages 1–14.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*. Sydney, Australia.

Joakim Nivre and Beáta Megyesi. 2007. Bootstrapping a Swedish treeebank using cross-corpus harmonization and annotation projection. In *Proceedings of the the 6th International Workshop on Treebanks and Linguistic Theories*. Bergen, Norway, pages 97–102.

Carla Parra Escartín, Wessel Reijers, Teresa Lynn, Joss Moorkens, Andy Way, and Chao-Hong Liu. 2017. Ethical considerations in nlp shared tasks. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain, pages 66–73. http://aclweb.org/anthology/W17-1608.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. 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, pages 433–440.

Slav Petrov and Dan Klein. 2007. 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, pages 404–411.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. 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, pages 146–182.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin

Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2014. Overview of the SPMRL 2014 shared task on parsing morphologically rich languages. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*. Dublin, Ireland.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altmann, and Noa Nativ. 2001. Building a tree-bank of Modern Hebrew text. *Traitement Automatique des Langues* 42:347–380.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2015. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

Andreas van Cranenburgh and Rens Bod. 2013. Discontinuous parsing with an efficient and accurate DOP model. In *Proceedings of the International Conference on Parsing Technologies (IWPT 2013)*. Nara, Japan.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering* 11(2):207—238.