Exploring Ensemble Dependency Parsing to Reduce Manual Annotation Workload

Jessica Sohl(✉) and Heike Zinsmeister

Institute for German Language and Literature, Universität Hamburg, Hamburg, Germany
jessica.katharina.sohl@studium.uni-hamburg.de,
heike.zinsmeister@uni-hamburg.de

Abstract. In this paper we present an evaluation of combining automatic and manual dependency annotation to reduce manual workload. More precisely, an ensemble of three parsers is used to annotate sentences of German textbook texts automatically. By including a constrained-based system in the cluster in addition to machine learning approaches, this approach deviates from the original ensemble idea and results in a highly reliable ensemble majority vote. Additionally, our explorative use of dependency parsing identifies error-prone analyses of different systems and helps us to predict items that do not need to be manually checked. Our approach is not innovative as such but we explore in detail its benefits for the annotation task. The manual workload can be reduced by highlighting the reliability of items, for example, in terms of a 'traffic-light system' that signals the reliability of the automatic annotation.

1 Introduction

Corpus-based linguistic analyses that rely on annotated data require high-quality annotations to be accepted by the community. Working with reference corpora is not useful in many cases because their data is very limited and not suitable for many research questions. Simultaneously, creating manual annotation for new data is very time-consuming, so it is necessary to make use of automated means. However, it is often not feasible for corpus-linguistic projects to create their own annotation tools. They have to rely on off-the-shelf programs. Fortunately, infrastructure efforts such as CLARIN[1] or META-NET[2] have made existing tools much easier accessible for reuse by the community.

One of the issues of working with off-the-shelf tools is that they are developed for or trained on particular texts, which are not necessarily of the same text type as the data of interest. This means that using off-the-shelf tools often coincides with applying the tools to out-of-domain data.

In this paper, we investigate the approach of applying a set of syntactic dependency parsers that are trained on a large newswire corpus to a corpus of

[1] CLARIN-D: https://www.clarin-d.de/en/.
[2] META-NET: http://www.meta-net.eu.

G. Rehm and T. Declerck (Eds.): GSCL 2017, LNAI 10713, pp. 40–47, 2018.
https://doi.org/10.1007/978-3-319-73706-5_4

'non-standard' texts to support manual annotation. The idea of such ensemble parsing is introduced in Sect. 2. After briefly discussing related work, we describe the setting of our study (Sect. 3): the set of parsers that constitute our parser ensemble; the training domain, which refers to the actual training data in the case of statistical parsers and to the data the constrained-based parser was incrementally tested and improved on, and finally, the test corpus, which consists of data from our target domain. In Sect. 4, we first present quantitative results (Sect. 4.1): We establish the accuracy of the parsers individually on the 'training domain'; we test the parsers individually on the target domain; and, finally, establish the best combination of three parsers in an ensemble setting. Second, in addition to these quantitative results, we analyze which kind of items the ensemble fails to parse correctly (Sect. 4.2). A detailed qualitative analysis helps to estimate the extent to which the parser ensemble can support manual annotation which is discussed in Sect. 5. The choice of parsers is motivated by taking the perspective of a corpus linguistics or digital humanities project that has only limited means for parser optimization itself but has to rely on well described ready-to-use tools.[3]

2 Ensemble Parsing

The concept of ensemble parsing has been thoroughly discussed by Van Halteren et al. (2001) for part-of-speech tagging. The crucial point is that a cluster of taggers is employed instead of a single tagger. There are several methods of combining the output of a tagger ensemble. In this paper we follow the 'multi-strategy approach' (Van Halteren et al. 2001, p. 201), in which tagger models are employed that result from training different learning algorithms on the same data. The key idea is that different taggers create their analyses in different ways such that their errors are uncorrelated. Van Halteren et al. (2001) suggest that a reasonable weighted combination of the tagger choices can obtain better results than the individual taggers do. Many studies applied the multi-strategy approach in a successful way also to dependency parsing (Brill and Wu 1998, Søegaard 2010, Rehbein et al. 2014, a. o.).

In this paper, we deviate from the original approach and include one constrained-based parser in addition to two statistically trained parsers and investigate to what extent this ensemble can support manual annotation of textbook texts.

3 Setting

We train both statistical parsers on a large reference corpus that was manually annotated and also used as a test-bed for developing our constrained-based parser. This ensures that all members of the ensemble are based on the same linguistic analyses (Fig. 1).

[3] We presented this work as an unpublished poster at the DGFS-CL poster session in 2017, http://dfgs2017.uni-saarland.de/wordpress/abstracts/clposter/cl_6_zins.pdf.

```
1   Das             das             ART ART     _   2   SUBJ    _   _
2   ist             sein            V   VVFIN   _   0   S       _   _
3   ein             ein             ART ART     _   4   DET     _   _
4   Beispielsatz    Beispielsatz    N   NN      _   2   PRED    _   _
5   .               .               $.  $.      _   0   ROOT    _   _
```

Fig. 1. Dependency parse in CoNLL format

3.1 Parser Ensemble

Our ensemble consists of three different parsers. The MALT parser (Nivre et al. 2006) creates its dependency trees by means of transition-based hypotheses.[4] The MATE parser (Bjöerkelund et al. 2010) is partly related but takes second order maximum spanning trees into account for creating its trees.[5] Finally, the JWCDG parser (The CDG Team 1997-15)[6] consists of (manually) weighted hand-written rules which were developed on the basis of Hamburg Dependency Treebank (HDT), see subsection 3.2.[7] For the ensemble, we took into account different combinations of parser outputs. In Sect. 4.1, we will present results for the two highest-scoring ensembles evaluated on the gold standard:

- Ensemble 1 (ENS-1): Majority vote of all three parsers agreeing on the annotation (Match-3) or at least two out of three parsers agreeing (Match-2); MATE as the best individual parser serves as the default when all parsers differ from each other.
- Ensemble 2 (ENS-2): Majority vote of all three parsers agreeing on the annotation (Match-3); MATE serves as the default otherwise, except MATE assigns one of the labels *S* or *OBJA* then the annotation of JWCDG is used instead.

Note that both ensembles rely heavily on the MATE parser: ENS-1 takes the output of MATE except for instances in which the other two parsers agree on a different label. ENS-2 accepts the annotation of MATE except for two labels which MATE generally overgenerates. In such instances, the ensemble takes the annotation of JWCDG independent of whether there is a majority vote or not.

3.2 Training Domain

Our training corpus is the Hamburg Dependency Treebank (HDT).[8] In particular, we used part A of the HDT (Foth et al. 2014) which contains 10,199 sentences produced by manual annotation and subsequent cross-checking for consistency with DECCA (Dickinson and Meurers 2003). The texts of the HDT

[4] We trained Maltparser v1.9.0 with default settings which results in a *non-optimized version* that does not do justice to the parser system as such.
[5] We used MATE transition-1.24 for training.
[6] The CDG Team (2997-2915): https://gitlab.com/nats/jwcdg; Version: 1.0.
[7] We had to dismiss the Turbo parser from our ensemble due to compilation problems.
[8] HDT: https://nats-www.informatik.uni-hamburg.de/HDT.

are crawled from the website *heise online*, a German-language technology news service mostly covering IT, telecommunications and technology.[9]

We divided HDT into ten equally sized bins and performed a 10-fold cross-validation of the statistical parsers, MALT and MATE, to estimate their in-domain performances. The final versions of the parsers were trained on the full corpus.

3.3 Test Domain and Gold Standard

Our test domain is textbook texts as used in books for German secondary schools. In particular, we used texts from an unpublished textbook corpus: 144 sentences from three different geography textbooks which correspond to one double page per book. We refer to double pages here because they commonly represent one informational unit in such textbooks. In the evaluation, we average the performances on the three double pages.

We developed a gold standard on the test corpus. To this end, two annotators annotated the data independently from scratch using the tagset of the HDT (see Sect. 3.2). The manual annotation resulted in an inter-annotator agreement (IAA) of unlabeled attachment score (UAS) of 0.95 ($\pm$0.01) and labeled attachment score (LAS) of 0.93 ($\pm$0.01) according to MaltEval (Nilsson and Nivre 2008). We also computed a chance-corrected IAA score for dependency annotation and obtained $\alpha = 0.93$ ($\pm$0.02) agreement (Skjærholt 2014).

4 Results

We present quantitative results for the individual parsers both on the training domain and on the test corpus. We also present quantitative results for two different ensemble settings. In the second part of this section, we take a closer look at the parsing failures and analyze the linguistic structures qualitatively that turned out to be problematic for the parsers.

4.1 Quantitative Results

The quantitative results on parsing accuracy are summarized in Fig. 2.

The x-axis represents our three different data sets: the training data from the HDT ("10-fold cross"), the test corpus ("Gold"), and finally the subset of gold instances on which all three parsers of the ensemble agreed upon ("Match-3"). The x-axis is furthermore divided into two different evaluation scores (see the top header): the labeled attachment score (LAS) to the left-hand side, which provides the percentage of tokens for which the system has predicted both the correct head and the correct dependency relation; the unlabeled attachment score (UAS) to the right-hand side, which is the more relaxed score that only checks for the correct head. The different parsers and ensembles are depicted by

[9] Heise online: heise.de.

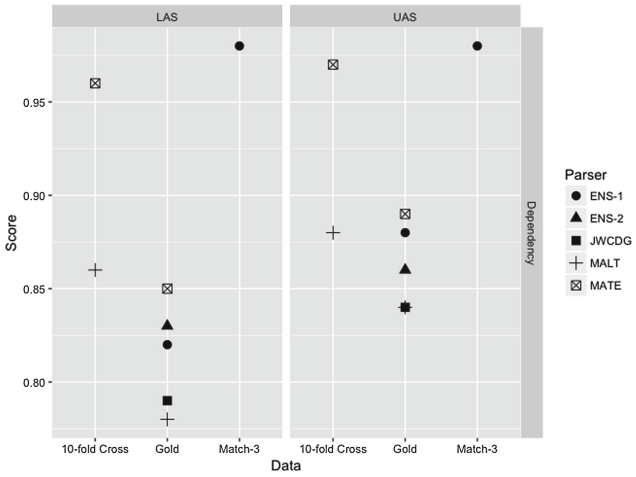

Fig. 2. Parsing accuracy (LAS and UAS) of the parsers (JWCDG, MALT, MATE) and two parser ensembles (ENS-1, ENS-2) on different data sets: training data (10-fold cross-validation), gold standard, and Match-3 items of the gold standard.

different shapes (for details see Sect. 3.1). It is expected that the LAS scores are generally lower than the UAS scores which holds true for all but the Match-3 data which we will discuss further below.

10-fold cross-validation. The evaluation on the HDT corpus of about 10,000 sentences shows that MATE outperforms the non-optimized version of MALT in a range of about 10% points (LAS: 0.86 MALT vs. 0.96 MATE; UAS: 0.88 MALT vs. 0.97 MATE).

Gold standard evaluation. We get a similar tendency on our 144 sentence test corpus (1,697 tokens; on average 566 tokens per double page) even if the difference is not as pronounced and the performance of both parsers drops substantially in comparison to the in-domain cross-validation results (LAS: 0.78 MALT vs. 0.84 MATE; UAS: 0.84 MALT vs. 0.88 MATE on average). The difference between the parsers is still significant (according to a one-tailed t-test for UAS: $t = 3.05$, $df = 2$, $p = 0.04639$; for LAS: $t = 3.1$, $df = 2$, $p = 0.02995$). The constrained-based JWCDG parser has similar performance to MALT and is also outperformed by MATE. The ensemble settings ENS-1 and ENS-2 (cf. Sect. 3.1) outperform JWCDG and MALT but do not quite reach the accuracy of MATE. Interestingly, ENS-1 is better than ENS-2 in assigning the overall dependency structures (UAS) whereas ENS-2 is more reliable in assigning the labels correctly (LAS).

Match-3. The final set is the subset of gold standard instances on which all three parsers of the ensemble agreed in head and label assignment.[10] This subset

[10] Figure 2 depicts ENS-1 only but the results hold for both ensemble settings.

consists of 1,128 tokens overall, on average 276 tokens per double page, i. e. about 71% (±0.10) of the tokens are a complete match of the three parsers. The ensembles performed very well on these instances (LAS and UAS both equal 0.98 on average, LAS having a greater variance than UAS). For practical issues it is relevant to look for complete sentences in this set. We observe that Match-3 contains 22 complete sentences, i. e. about 15% of the sentences per double page. All in all 21 out of 22 completely agreed-on sentences are correct.

4.2 Qualitative Results

Some of the parser failures can be related to general challenges in dependency parsing such as the decision of a prepositional phrase functioning as an object (OBJP) or an adverbial (PP) for a given verb (which strongly depends on the training data) and also attachment ambiguities (which require semantic decisions). Table 1 summarizes the major weaknesses of the individual parsers which we employed in creating the parser ensembles (cf. Sect. 3.1).

In addition to these parser-specific errors, we observed domain-specific challenges. The text in textbooks is presented in particular ways. For example, it contains a high amount of lists and exercises that are characterized by incomplete sentences which include list items and nominal structures as in Example (1).

(1) M4 Auswirkungen des Klimawandels am Beispiel “Starkregen”
‘M(aterial)4 Impact of climate change on the example of “severe rain”’

There are also non-finite verbal structures featuring the verb in its canonical VP-final position, cf. *kennen* ‘to know’ in Example (2).

(2) check-it:
Merkmale einer thematischen Karte – hier Bodennutzung – kennen
‘check it: - knowing the characteristics of a thematic map – here soil use.’

Table 1. Labels overgenerated by the parsers, potential trigger structures and overall number of errors per parser (APP = generalized apposition, S = root of sentence/fragment, OBJA = accusative object).

Parser	False label	Comment
JWCDG	APP (26%)	Default attachment
	S (15%)	Fragments
		#errors: 262
MALT	S (64%)	Incomplete sentences
		#errors: 335
MATE	S (30%)	Fragments
	OBJA (26%)	Confusion of SUBJ/ OBJA
		#errors: 225

Another issue that is claimed to be characteristic of German scholarly language in text books is complex syntax (e. g., (Griehaber 2013)). Our corpus contains some complex coordinations that are hard to parse even for humans. Example (3) is one of them.

(3) Als praktisch sicher gilt, dass es über den meisten Landflächen wärmere und weniger kalte Tage und Nächte sowie wärmere und häufiger heiße Tage und Nächte geben wird.
'It is virtually certain that there will be warmer and less cold days and nights, as well as warmer and more frequently hot days and nights over most areas.'

We expect that some of the domain-specific structures could be parsed in a more reliable way if the training corpus included data also from the target domain.

5 Conclusion

Our application of ensemble dependency parsing is highly reliable in terms of its ensemble majority vote. However, the ensembles do not outperform the best individual parser. Nevertheless, we can make use of the ensemble to support manual correction. This again means we can very well skip certain labels (e. g., AUX(iliary), DET(erminer), G(enitive)MOD(ifier)) and also complete sentence matches. In addition, we can support manual annotation by highlighting error-prone labels that are easily confused such as OBJP and PP and also areas of the text that are sensitive to errors, e. g., lists and exercises.

The results could be further improved by applying domain adaptation methods such as re-training the statistical parsers and including the gold standard in the training data. More sophisticated methods such as optimizing the parsers' features or combining the parsers with other dependency parsers (e. g., Nivre and McDonald (2008); Köhn and Menzel (2013)) are out of the scope of this project.

Acknowledgments. Jessica Sohl's work was supported by a scholarship of the Hamburg Humanities Graduate School. We would like to thank Sarah Tischer and the anonymous reviewers for helpful comments, and Piklu Gupta for improving our English. All remaining errors remain ours.

References

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

Brill, E., Wu, J.: Classifier combination for improved lexical disambiguation. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol. 1, pp. 191–195 (1998)

Dickinson, M., Meurers, W.D.: Detecting Errors in Part-of-Speech annotation. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 107–114 (2003)

Foth, K., Köhn, A., Beuck, N., Menzel, W.: Because size does matter: the Hamburg dependency treebank. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2014), pp. 2326–2333 (2014)

Griehaber, W.: Die Rolle der Sprache bei der Vermittlung fachlicher Inhalte. In: Rhner, C., Hvelbrinks, B. (eds.) Fachbezogene Sprachfrderung in Deutsch als Zweitsprache. Theoretische Konzepte und empirische Befunde zum Erwerb bildungssprachlicher Kompetenzen, pp. 58–74. Juventa, Weinheim (2013)

Köhn, A., Menzel, W.: Incremental and predictive dependency parsing under real-time conditions. In: Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2013), pp. 373–381 (2013)

Nilsson, J., Nivre, J.: Malteval: an evaluation and visualization tool for dependency parsing. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), pp. 161–166 (2008)

Nivre, J., McDonald, R.: Integrating graph-based and transition-based dependency parsers. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT), pp. 950–958 (2008)

Nivre, J., Hall, J., Nilsson, J.: MaltParser: a data-driven parser-generator for dependency parsing. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), pp. 2216–2219 (2006)

Rehbein, I., Schalowski, S., Wiese, H.: The KiezDeutsch Korpus (KiDKo) release 1.0. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), pp. 3927–3934 (2014)

Skjærholt, A.: A chance-corrected measure of inter-annotator agreement for syntax. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 934–944 (2014)

Søgaard, A.: Simple semi-supervised training of part-of-speech taggers. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 205–208 (2010)

Van Halteren, H., Zavrel, J., Daelemans, W.: Improving accuracy in word class tagging through the combination of machine learning systems. Comput. Linguist. **27**(2), 199–229 (2001)