Toward a Treebank Collecting German Aesthetic Writings of the Late 18th Century

Alessio Salomoni
University of Bergamo-Pavia
Corso Strada Nuova 65
Pavia, Italy, 27100
alessio.salomoni@unibg.it

Abstract

English. In this paper, I will describe the methodology to develop the first sample of a dependency treebank collecting German aesthetic writings of the late 18th century. A gold standard of the target data was annotated in order to evaluate some data-driven tools, trained on contemporary web news. Results are reported and discussed.

Italiano. *In questo articolo descriverò la metodologia adottata nello sviluppo di un sample preliminare di una treebank per il tedesco, che raccoglierà scritti di estetica della fine del XVIII secolo. È stato annotato un campione della varietà target, ed è stata valutata l'accuratezza di alcuni strumenti data-driven addestrati su una varietà giornalistica contemporanea. I risultati sono stati riportati e commentati.*

1 Introduction

A constantly increasing amount of digital texts of the German literary history is freely available online as downloadable raw texts, especially thanks to important ongoing projects, such as deutschestextarchiv.de or zeno.org, to name but a few. In spite of this, we still lack annotated corpora gathering them per author and genre. Indeed, this is a strong bottleneck in exploiting such textual treasure for linguistic analysis through computational methods. At the same time, available training data for data-driven annotation tools mainly come from the domain of contemporary web news. Therefore, models have to be trained on this particular variety of the German language, which could be very different, in terms of linguistic features, from the target unannotated data. Such variation between the training set and the test set could cause tools' performances to drop (Gildea, 2001). Therefore, testing such models on a portion of the target texts is crucial. On the one hand, to show their robustness. On the other hand, more practically, to understand to what extent available tools can actually boost the semi-automatic annotation of new data.

In this paper, I will highlight the methodology behind the development of a first sample of a dependency treebank aiming to collect German aesthetic essays of the late 18th century. By aesthetic essays I mean theoretical writings about art, poetics, beauty and related issues, which were mainly published on literary magazines, chiefly targeting non-academic middle-class readers. [1] In that period, there was a remarkable production of these texts in Germany, and they contributed to popularize the recently born modern 'Hochdeutsch', i.e. the modern variety of the German language. To the best of my knowledge, despite its importance, such textual genre has never been studied in depth at any linguistic level. In a long-term perspective, a dependency treebank will surely provide empirical data to fill the gap, especially concerning syntax and semantics. Indeed, many studies can be done on such resource, ranging from using dependency networks to describe syntactic phenomena (Passarotti, 2014), to extracting a valency lexicon (Passarotti et al., 2016).

In the rest of this paper, some fundamental issues concerning the treebank design are highlighted and preliminary results concerning automatic lemmatization, POS-tagging and dependency parsing are reported and discussed.

[1] Philosophical monographs about aesthetics from the same period are not part of the target data for this resource, belonging to a different genre.

2 Methodology

2.1 Data

Even if we are dealing with texts in prose in a defined domain, style between authors may vary substantially, especially in terms of syntax and lexicon. Therefore, to avoid too much variation in my data, for this first sample I focused on a particular text typology inside the target genre: fragments, i.e. really short texts, sometimes in aphorism-like form. I assumed that such texts could be dealt with as a whole, in spite of their different authorship. [2] For the first sample of the treebank, I selected the following data: F. Schlegel, *Lyceum Fragmente*, fragments from 1 to 90; F. Schlegel and other authors, *Athenaeum Fragmente*, fragments from 1 to 50; Novalis, *Blüthenstaub*, fragments from 1 to 31. All the raw texts in .txt format were obtained from zeno.org. Overall, this initial corpus counts 7337 tokens.

2.2 Annotating a Gold Standard

Such corpus was semi-automatically annotated to build a gold standard. As for the annotation scheme, I adhered to the Universal Dependencies (UD) 2.0 scheme (Nivre et al., 2017). Texts were tokenized and brought into conllu format with UDPipe1.1 (Straka et al., 2016). Then they were brought into conll09 format (Hajič et al., 2009) and processed with Anna 3.6 pipeline (Bohnet, 2010).[3] I had used this suite in previous preliminary experiments on some data from the same period and domain, attaining good initial results for POS-tagging and dependency parsing. I assigned the following metadata: LEMMA, UPOS (the coarse-grained POS-tag, based on the Google tagset (Petrov et al., 2011)), XPOS (the fine-grained POS-tag, based on the STTS tagset (Brants et al., 2002)), HEAD (the regent element of the dependency relation) and DEPREL (the kind of dependency relation). As for LEMMA and XPOS, pre-trained models based on the Tiger Corpus (Brants et al., 2002) were used. As for UPOS, HEAD and DEPREL, I trained a model on the training file of the German treebank in UD 2.0[4]. Then, at each stage of the processing, the automatic output was manually checked. An annotated fragment is shown in Figure 1 in a tree-like form.

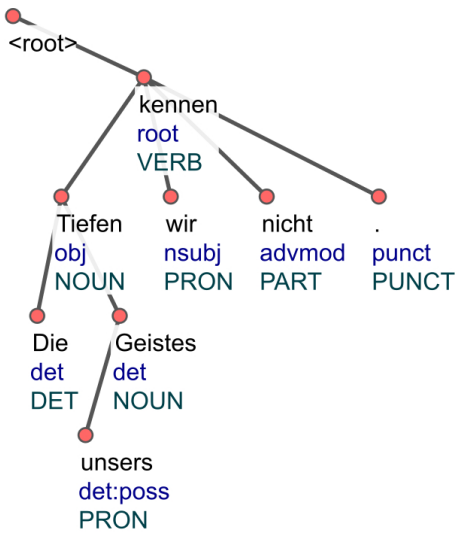

Figure 1: Dependency representation of the simple German sentence 'Die Tiefen unsers Geistes kennen wir nicht.' (We don't know the depths of our soul) by Novalis, according to UD 2.0 scheme.

I briefly describe the formalism in Figure 1. The main node of each sentence usually is the main verb, which is 'kennen' in this case, whose relation is tagged as 'root'. The article 'Die' depends on the common noun 'Tiefen' as determiner, while 'Tiefen' depends on 'kennen' as nominal object. 'Wir' is a personal pronoun playing the role of nominal subject. 'unsers' is a possessive pronoun modifying the common noun 'Geistes', which is in genitive case and modifies the subject. According to the current scheme, such modifier depends on the noun it refers to through 'det' relation.

2.3 Lemmatization

Training	Lemmatizer	Frag
Tiger (pre-trained)	Anna 3.6	97.6

Table 1: Accuracy by Anna 3.6 lemmatizer on the target data. 'Frag' stands for accuracy on fragments.

As for lemmatization, I measured the accuracy by Anna 3.6 lemmatizer (Björkelund et al., 2010) on fragments only. Results are shown in Table 1.[5] Given the high overall accuracy by Anna 3.6, I did

[2] Preliminary clustering and syntactic parsing experiments confirmed this hypothesis.

[3] Double multi-word tokens such as 'der+im' for the determined article 'dem' or 'in+dem' for the preposition 'im' had to be removed to work with this format.

[4] It counts about 287.000 tokens.

[5] All the results in this paper are expressed as percentage.

not test any other system. I briefly report some issues concerning this task: inflected adjectives such as 'andre' or 'unsrer' where 'e' in stem drops after inflection (for instance, the stem of 'unsrer' is 'unser') are lemmatized without 'e'; deadjectival nouns such as 'Langweile' or 'Kürzeste' are lemmatized as nouns with the same form, not as adjectives; the non-finite verb 'seyn' is lemmatized as 'seyn', not with the current spelling 'seien'.

2.4 POS-Tagging

As for POS-tagging, I tested some candidate POS-taggers on fragments first. Once the best-performing one was detected, I tested it also on the source variety to measure the accuracy gap. Before doing that, I had to cope with some issues concerning models and training data. According to the documentation provided with the treebank file, in the UD German treebank UPOS was assigned manually, while XPOS was assigned automatically by using Tree Tagger, trained on Tiger Corpus, with no manual checking. Thus, the UD treebank was not ideal to train a model for XPOS. At the same time, I was interested in testing both tagsets on the target data. Consequently, I followed two different methods. First, I considered the UPOS tagset. I picked up two candidate POS-taggers, I trained them on the whole training file of the German treebank in UD and I tested them on fragments. They were fed with the automatically lemmatized texts by Anna 3.6. Overall accuracy is shown in Table 2. [6]

Training	POS-tagger	Frag
100% de-ud-train	Anna 3.6	93
	UDPipe 1.1	88.5

Table 2: Accuracy by Anna 3.6 and UDPipe 1.1 POS-tagger (Straka et al., 2016) assigning UPOS to fragments.

The best POS-tagger was Anna 3.6, thus I run it on UD, performing a ten-fold validation. I split up the training file of the UD 2.0 German treebank into two partitions with ratio 9:1. I trained the POS-tagger on the 90% and tested it on the remaining 10%. I repeated the experiment ten times, varying each time the two partitions. Overall accuracy concerning these experiments, i.e. the average of the ten measures, is shown in Table 3.

Training	POS-tagger	UD
90% de-ud-train	Anna 3.6	93.6

Table 3: Overall average accuracy by Anna 3.6 in assigning UPOS to the UD test set.

As for Anna 3.6 POS-tagger, I report the accuracy on some specific part-of-speeches on both test sets. The first number in brackets refers to UD [7], while the second one to the fragments: VERB (95.1/94.8), PROPN (proper nouns) (84.01/83.6); NOUN (93.71/94.23); SCONJ (subordinate conjunctions) (89.1/79); ADJ (adjectives) (91.2/94); AUX (auxiliaries) (83.9/77.7) and ADV (adverbs) (90.7/83.02). There is a remarkable gap between the two varieties on adverbs, subordinating conjunctions and auxiliaries. On fragments, a lot of adverbs have been mismatched with adjectives, for instance when they modify adjectives, while many occurrences of the subordinate conjunction 'daß' have been wrongly assigned. As for AUX, the modal verb 'müssen' was frequently assigned a wrong POS. As for VERB, the verb 'sein' was frequently tagged as AUX when it occurs as verbal part of a nominal predicate, while, in this case, it should be tagged as VERB, according to the UD scheme.

Training	POS-tagger	Frag
Tiger (p)	Anna 3.6	97.3
Tiger (p)	RFTagger	88
Negra (p)	Stanford	92.9

Table 4: Accuracy by Anna 3.6, RFTagger (Schmid and Laws, 2008) and Stanford Tagger (Manning et al., 2014) assigning XPOS to fragments. 'p.' stands for pre-trained model.

Second, I considered XPOS. At first, I tested three POS-taggers which are commonly used with the STTS tagset on fragments. I used pre-trained models provided by developers. Overall results are shown in Table 4. Anna 3.6 outperformed other candidates, and its overall accuracy is clearly

[6] In all these POS-tagging experiments, accuracy is the number of correctly assigned POS-tags divided by the total number of POS-tags in the test set.

[7] The reported value is the average of the ten accuracy values attained on each POS in each experiment of the ten-fold validation.

higher than that on UPOS on the same test set. Such a significant improvement could be due to the considerably different size of the training sets.[8]

Following the method adopted in the UPOS session, I performed a ten-fold validation of Anna 3.6 POS-tagger on Tiger Corpus 2.2. Overall average accuracy was 97.7. Results concerning single selected POS on both test sets is shown in Table 5. To remind the difference in granularity between the two tagsets, for each group of XPOS I reported the corresponding UPOS as well. In contrast to UPOS, problems concerning auxiliaries and subordinating conjunctions on fragments seem to be overcome, while there are still issues concerning non-finite modal verbs, such as 'müssen'.

UPOS	XPOS	Tiger	Frag
VERB	VVFIN	93.3	94.5
	VVINF	93.4	96.1
	VVPP	95.8	96
	VVIZU	93	100
AUX	VMFIN	98.6	100
	VMINF	75	88
	VAFIN	98.4	100
	VAINF	94	95.4
ADJ	ADJA	98.3	97.7
	ADJD	94	95.5
ADV	ADV	97.2	88.4
NOUN	NN	98.7	99.2
PROPN	NE	92.1	95.5
SCONJ	KOUS	97.7	100

Table 5: Overall accuracy by Anna 3.6 in assigning XPOS to UD and fragments. As for verbs, VVFIN stands for finite verbs, VVINF for non-finite verbs, VVPP for past participle, VVIZU for non-finite verbs in non-finite clauses. As for auxiliaries, it is the same, with A standing for auxiliary and M standing for modal. For further details, I redirect to STTS online documentation.

2.5 Dependency Parsing

As for dependency parsing, I tested four different candidate parsers. First, I performed a ten-fold validation on the training set of the UD German treebank, using the same partitions from the POS-tagging session. Second, the parsers were trained on the whole training set of the German treebank and tested on fragments. In this case, morphological features were removed from the training set, because they have not been annotated in my test set yet, therefore the parsing model should not include them. All the four parsers were fed with the automatically lemmatized and POS-tagged texts (both with UPOS and XPOS). Such metadata were assigned by Anna 3.6. The candidate parsers and their settings are introduced below, while overall results are reported in Table 6. Parsing accuracy was measured through Malt Eval (Nivre et al., 2010) and it is expressed in terms of *labeled attachment score* (LAS).

- **Malt Parser 1.9.0** (Nivre et al., 2006), a transition-based system. This parser performs better with an optimized configuration obtained through Malt Optimizer, i.e. a software able to suggest the best parsing configuration after reading the training data. First, I run Malt Optimizer on the ten partitions of the training file of the UD German Treebank. Then, for each of them, the suggested configuration was used to parse the corresponding test set. Second, Malt Optimizer (Ballesteros and Nivre, 2012) was run on the whole UD training file, and the suggested configuration [9] was used to parse the target variety.

- **Anna 3.6** (Bohnet, 2010) by Mate Tools, a graph-based system. It was run with 10 training iterations.

- **Joint Parser 1.30** (Bohnet and Nivre, 2012), a transition-based system with beam search, graph completion model and an integrated part-of-speech tagger. It was run with the R6J transition, 25 training iterations and beam search parameter fixed at 40.

- **Parsito**, a transition-based system with a neural network classifier, included in the UDPipe 1.1 suite (Straka et al., 2016). It was run in the standard configuration.

Overall, Anna 3.6 attained the highest accuracy on both test sets. However, there is a 19.2% accuracy gap between the two top scores on the two varieties.

[8] Indeed, Tiger Corpus 2.2 is about three times bigger than the UD German treebank used to train the model for UPOS.

[9] system: liblinear; feature model: addMergPOSTAGS0I0FORMLookahead0; algorithm: stackproj

Training	Parser	UD	Frag
100% de-ud-train	Malt 1.9	81.1	61.3
	Anna 3.6	84. 6	65.4
	Joint	81	64.2
	Parsito	83	60.6

Table 6: Overall accuracy by four different dependency parsers on UD and on fragments.

2.6 Parsing in-depth Evaluation

In order to detect which syntactic relations are more difficult to correctly parse in fragments, I did an in-depth evaluation for all the parsers. Accuracy concerning some of the most problematic relations is reported in Table 7. [10]

Deprel	Parser	F-Score Frag
acl	Malt	52.7
	Anna	69.2
	Joint	61.7
	Parsito	63.8
xcomp	Malt	27.6
	Anna	36.5
	Joint	30.8
	Parsito	33.6
advcl	Malt	39.7
	Anna	62.5
	Joint	51.6
	Parsito	55.6
conj	Malt	67.9
	Anna	77.2
	Joint	61.8
	Parsito	725
root	Malt	68.2
	Anna	73.8
	Joint	74.6
	Parsito	73.2

Table 7: Parsing accuracy on single dependency relations.

I supply a brief description of the dependency relations I reported in Table 7. 'acl' stands for adjectival clause modifier, i.e. it refers to all those finite and non-finite clauses modifying a noun, such as the relative clauses. For instance, it occurs between the noun 'Apfel' in the main clause and the subordinate verb 'liegt' in the sentence 'Die Apfel, die auf dem Tisch liegt' (The apple, that is on the table). It is different from 'advcl', which stands for adverbial clause, i.e. a clause modifying a predicate not as a core argument. It occurs, for instance, between the subordinate verb and the main verb in the sentence 'Ich denke, dass diese Prüfung ganz schwierig ist' (I think that this exam is really difficult). 'xcomp' stands for all those predicative or clausal complements without their own subject. In German, such function matches different syntactic phenomena. For example, it occurs between the main verb and the subordinate verb in non-finite clauses introduced by the particle 'zu', such as in 'Ich habe viel zu tun' (I have a lot to do); or between the predicative part of verbs such as 'lassen' , 'scheinen' or even 'nennen' and the verb, such as in 'Ich lasse dich gehen' (I let you go). 'conj' is the relation occurring between coordinate items, while 'root', as shown in Figure 1, is the dependency relation assigned to the main predicate of each sentence.

In German, the subordinate verb lies at the end of the clause, thus relation length, i.e the number of tokens between the head (in this case the main verb) and the dependent (the subordinate verb), may be really high. This can play a crucial role in parsing accuracy, especially for transition-based systems. Malt parser mostly attained low accuracy on this kind of relations, while performances by this system increases on 'conj' relation. This could be do to the relatively low frequency of coordinate relations occurring between verbs in this test set (23% of all 'conj' relations), which are usually more likely to generate long relations. Anna 3.6 sensibly outperformed the other systems on 'acl', 'advcl' and on 'conj' too. As for the 'root' relation, a part from Malt Parser, performances are almost similar. On 'xcomp', accuracy by all the systems dramatically drops. This could be due to the high relation length between some non-finite verbs and their heads, but also to the wide range of different syntactic constructions in which such relation occurs.

3 Conclusion and Future Work

In this work, I described the methodology behind the development of a first sample of a German treebank collecting a particular kind of aesthetic essays from the late 18th century, called

[10] I have not done an in-depth evaluation of the results on UD yet.

fragments. A gold standard was annotated adhering to UD 2.0. Then some data-driven tools were tested either on the target data and on a test set of the source variety. Some core issues concerning the automatic annotation were highlighted. As for LEMMA and XPOS, overall accuracy on the target data was high and very close to that on the source variety. As for UPOS, the accuracy by the best tagger dropped, especially on the target data. Therefore, to assign POS-tag, the very good results on the STTS tagset may suggest to automatically assign XPOS first and then derive UPOS from XPOS. Furthermore, the influence of POS-tagging granularity on parsing has not been studied yet. As for dependency parsing, the overall gap between the target variety and the source variety was remarkable (19%). An in-depth comparison between the two varieties concerning single relations will surely help to better detect parsing problems on fragments. In addition, parsing manually lemmatized and POS-tagged texts will surely shed light on the error propagation on parsing.

References

Miguel Ballesteros and Joakim Nivre. 2012. Maltoptimizer: an optimization tool for maltparser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–62. Association for Computational Linguistics.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 33–36. Association for Computational Linguistics.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465. Association for Computational Linguistics.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd international conference on computational linguistics*, pages 89–97. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, volume 168.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Joakim Nivre, Laura Rimell, Ryan McDonald, and Carlos Gomez-Rodriguez. 2010. Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 833–841. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Marco Passarotti, Berta González Saavedra, and Christophe Onambélé. 2016. Latin vallex. a treebank-based semantic valency lexicon for latin. In *LREC*.

Marco Passarotti. 2014. The importance of being sum. network analysis of a latin dependency treebank. In *Proceedings of La Prima Conferenza Italiana di Linguistica Computazionale*, pages 291–295.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Helmut Schmid 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*, pages 777–784. Association for Computational Linguistics.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *LREC*.