

Transition-based dependency parsing with topological fields

Daniël de Kok and Erhard Hinrichs

Seminar für Sprachwissenschaft

Wilhemstraße 19

72072, Tübingen, Germany

{daniel.de-kok, erhard.hinrichs}@uni-tuebingen.de

Abstract

The topological field model is commonly used to describe the regularities in German word order. In this work, we show that topological fields can be predicted reliably using sequence labeling and that the predicted field labels can inform a transition-based dependency parser.

1 Introduction

The topological field model (Herling, 1821; Erdmann, 1886; Drach, 1937; Höhle, 1986) has traditionally been used to account for regularities in word order across different clause types of German. This model assumes that each clause type contains a *left bracket* (*LK*) and a *right bracket* (*RK*), which appear to the left and the right of the *middle field* (*MF*). Additionally, in a verb-second declarative clause, the *LK* is preceded by the *initial field* (*VF*) with the *RK* optionally followed by the *final field* (*NF*).¹ Table 1 gives examples of topological fields in verb-second declarative (*MC*) and verb-final relative (*RC*) clauses.

Certain syntactic restrictions can be described in terms of topological fields. For instance, only a single constituent is typically allowed in the *VF*, while multiple constituents are allowed in the *MF* and the *NF*. Many ordering preferences can also be stated using the model. For example, in a main clause, placing the subject in the *VF* and the direct object in the *MF* is preferred over the opposite order.

In parsing, topological field analysis is often seen as a task that is embedded in parsing itself. For instance, Kübler (2005), Maier (2006), and Cheung and Penn (2009) train PCFG parsers on

¹The abbreviations are derived from the German terms *linke Klammer*, *rechte Klammer*, *Mittelfeld*, *Vorfeld*, and *Nachfeld*.

treebanks that annotate topological fields as interior nodes. It is perhaps not surprising that this approach works effectively for phrase structure parsing, because topological fields favor annotations that do not rely on crossing or discontinuous dependencies (Telljohann et al., 2006).

However, the possible role of topological fields in statistical dependency parsing (Kübler et al., 2009) has not been explored much. We will show that statistical dependency parsing of German can benefit from knowledge of clause structure as provided by the topological field model.

2 Motivation and corpus analysis

Transition-based dependency parsers (Nivre, 2003; Kübler et al., 2009) typically use two transitions (*LEFT_ARC* and *RIGHT_ARC*) to introduce a dependency relation between the token that is on top of the processing stack and the next token on the buffer of unprocessed tokens. The decision to make an attachment, the direction of attachment, and the label of the attachment is made by a classifier. Consequently, a good classifier is tasked to learn syntactic constraints, ordering preferences, and selectional preferences.

Since transition-based dependency parsers process sentences in one deterministic linear-time left-to-right sweep, the classifier typically has little global information. One popular approach for reducing the effect of early attachment errors is to retain some competition between alternative parses using a globally optimized model with beam search (Zhang and Clark, 2008). Beam search presents a trade-off between speed (smaller beam) and higher accuracy (larger beam). More recently, Dyer et al. (2015) have proposed to use *Long short-term memory networks* (*LSTMs*) to maintain (unbounded) representations of the buffer of unprocessed words, previous parsing ac-

	VF	LK	MF	RK	NF
MC:	In Tansania	ist	das Rad mehr	verbreitet	als in Uganda
	In Tansania	is	the bike more	common	than in Uganda
RC:		der	fünfmal mehr nach Bremerhaven	liefert	als Daewoo
		who	five-times more to Bremerhaven	delivers	than Daewoo

Table 1: Topological fields of a verb-second clause and a verb-final clause.

tions, and constructed tree fragments.

We believe that in the case of German, the topological field model can provide a linguistically-motivated approach for providing the parser with more global knowledge of the sentence structure. More concretely, if we give the transition classifier access to topological field annotations, it can learn regularities with respect to the fields wherein the head and dependent of a particular dependency relations lie.

In the remainder of this section, we provide a short (data-driven) exploration of such regularities. Since there is a myriad of possible triples² consisting of *relation*, *head field*, and *dependent field*, we will focus on dependency relations that virtually never cross a field and relations that nearly always cross a field.

Table 2 lists the five dependency relation that cross fields the least often in the TüBa-D/Z treebank (Telljohann et al., 2006; Versley, 2005) of German newspaper text. Using these statistics, a classifier could learn hard constraints with regard to these dependency relations — they should never be used to attach heads and dependents that are in different fields.

Dependency label	Cross-field (%)
Particles	0.00
Determiner	0.03
Adjective or attr. pronoun	0.04
Prepositional complement	0.04
Genitive attribute	0.07

Table 2: The five dependency relations that most rarely cross fields in the TüBa-D/Z.

Table 3 lists the five dependency relations that cross fields most frequently.³ These relations (virtually) always cross fields because they are verbal attachments and verbs typically form the LK and RK. This information is somewhat informative,

²335 in the TüBa-D/Z treebank.

³Dependency relations that connect two clauses are excluded.

since a classifier should clearly avoid to attach tokens within the same field using one of these relations. However, we can gain more interesting insights by looking at the dependents’ fields.

Dependency label	Cross-field (%)
Expletive <i>es</i>	100.00
Separated verb prefix	100.00
Subject	100.00
Prepositional object	99.80
Direct object	99.51

Table 3: The five dependency relations that most frequently cross fields in the TüBa-D/Z.

Table 4 enumerates the three (where applicable) most frequent head and dependent field combinations of the five relations that always cross fields. As expected, the head is always in the LK or RK. Moreover, the dependents are in VF or MF in the far majority of cases. The actual distributions provides some insights with respect to these dependency relations. We will discuss the *direct object*, *prepositional object*, and *separated verb prefix* relations in some more detail.

Direct objects In German, direct objects can be put in the VF. However, we can see that direct object fronting only happens very rarely in the TüBa-D/Z. This is in line with earlier observations in corpus-based studies (c.f. Weber and Müller (2004)). Since the probability of having a subject in the VF is much higher, the parser should attach the head of a noun phrase in the VF as a subject, unless there is overwhelming evidence to the contrary, such as case markers, verb agreement, or other cues (Uszkoreit, 1984; Müller, 1999).

Prepositional objects The dependency annotation scheme used by the TüBa-D/Z makes a distinction between prepositional phrases that are a required complement of a verb (*prepositional objects*) and other prepositional phrases. Since a statistical dependency parser does not typically have access to a valency dictionary, it has difficulty de-

Dependency label	Head	Dep	%
Expletive <i>es</i>	RK	MF	44.23
	RK	VF	32.99
	LK	VF	13.43
Separated verb prefix	LK	RK	99.95
	RK	RK	00.05
Subject	LK	VF	36.40
	LK	MF	35.10
	RK	MF	20.11
Prepositional object	RK	MF	51.04
	LK	MF	39.81
	LK	VF	04.11
Direct object	RK	MF	54.84
	LK	MF	35.64
	RK	LK	03.38

Table 4: The three most frequent head-dependent field combinations of the five relations that always cross fields.

deciding whether a prepositional phrase is a prepositional object or not. Topological field information can complement verb-preposition co-occurrence statistics in deciding between these two different relations. The prepositional object mainly occurs in MF, while a prepositional phrase headed by the LK is almost as likely to be in the VF as in the MF (42.12% and 55.70% respectively).

Separated verb prefixes Some verbs in German have separable prefixes. A complicating factor in parsing is that such prefixes are often words that can also be used by themselves. For example, in (1-a) *fest* is a separated prefix of *bindet* (present tense third person of *festbinden*), while in (1-b) *fest* is an optional adverbial modifier of *gebunden* (the past participle of *binden*).

- (1) a. Sie bindet das Pferd fest .
 She ties the horse tight .
 b. Das Buch ist fest gebunden .
 The book is tightly bound .

Similarly to prepositional objects, a statistical parser is handicapped by not having an extensive lexicon. Again, topological fields can complement co-occurrence statistics. In (1-a), *fest* is in the RK. As we can see in Table 4, the separated verb prefix is always in the RK. In contrast, an adverbial modifier as in (1-b) is rarely in the RK (0.35% of the adverbs cases in the TüBa-D/Z).

3 Predicting fields

As mentioned in Section 1, topological field annotation has often been performed as a part of phrase structure parsing. In order to test our hypothesis that topological field annotation could inform dependency parsing, it would be more appropriate to use a syntax-less approach. Several shallow approaches have been tried in the past. For instance, Veenstra et al., (2002) compare three different chunkers (finite state, PCFG, and classification using memory-based learning). Becker and Frank (2002) predict topological fields using a PCFG specifically tailored towards topological fields. Finally, Liepert (2003) proposes a chunker that uses support vector machines.

In the present work, we will treat the topological field annotation as a sequence labeling task. This is more useful in the context of dependency parsing because it allows us to treat the topological field as any other property of a token.

Topological field projection In order to obtain data for training, validation, and evaluation, we use the TüBa-D/Z treebank. Topological fields are only annotated in the constituency version of the TüBa-D/Z, where the fields are represented as special constituent nodes. To obtain token-level field annotations for the dependency version of the treebank, we project the topological fields of the constituency trees on the tokens. The recursive projection function for projection is provided in Appendix B. The function is initially called with the root of the tree and a special unknown field marker, so that tokens that are not dominated by a topological field node (typically punctuation) also receive the topological field feature.

We should point out that our current projection method results in a loss of information when a sentence contains multiple clauses. For instance, an embedded clause is in a topological field of the main clause, but also has its own topological structure. In our projection method, the topological field features of tokens in the embedded clause reflect the topological structure of the embedded clause.

Model Our topological field labeler uses a recurrent neural network. The inputs consist of concatenated word and part-of-speech embeddings. The embeddings are fed to a bidirectional LSTM (Graves and Schmidhuber, 2005), on which we stack a regular LSTM (Hochreiter and Schmidhu-

ber, 1997), and finally an output layer with the softmax activation function. The use of a recurrent model is motivated by the necessity to have long-distance memory. For example, (2-a) consists of a main clause with the LK *wird* and RK *begrünt* and an embedded clause *wie geplant* with its own clausal structure. When the labeler encounters *jetzt*, it needs to ‘remember’ that it was in the MF field of the main clause.

- (2) a. Die neue Strecke wird , wie geplant ,
The new stretch is , as planned ,
jetzt begrünt .
now being-greened .

Moreover, the use of a bidirectional LSTM is motivated by the need for backwards-flowing information to make some labeling decisions. For instance, *die Siegerin* is in the VF of the verb-second clause (3-a), while it is in the MF of the verb-final clause (3-b). The labeller can only make such choices by knowing the position of the finite verb.

- (3) a. die Siegerin wurde disqualifiziert
die winner was disqualified
b. die Siegerin zu disqualifizieren
the winner to disqualify

4 Parsing with topological fields

To evaluate the effectiveness of adding topological fields to the input, we use the publicly available neural network parser described by De Kok (2015). This parser uses an architecture that is similar to that of Chen and Manning (2014). However, it learns morphological analysis as an embedded task of parsing. Since most inflectional information that can be relevant for parsing German is available in the prefix or suffix, this parser learns morphological representations over character embeddings of prefixes and suffixes.

We use the same parser configuration as that of De Kok (2015), with the addition of topological field annotations. We encode the topological fields as one-hot vectors in the input of the parser. This information is included for the four tokens on top of the stack and the next three tokens on the buffer.

5 Evaluation and results

To evaluate the proposed topological field model, we use the same partitioning of TüBa-D/Z and the word and tag embeddings as De Kok (2015). For training, validation, and evaluation of the parser, we use these splits as-is. Since we want to test the

parser with non-gold topological field annotations as well, we swapped the training and validation data for training our topological field predictor.

The parser was trained using the same hyperparameters and embeddings as in De Kok (2015). Our topological field predictor is trained using Keras (Chollet, 2015).⁴ The hyperparameters that we use are summarized in Appendix A. The topological field predictor uses the same word and tag embeddings as the parser.

In Table 5, we show the accuracy of the topological field labeler. The use of a bi-directional LSTM is clearly justified, since it outperforms the stacked unidirectional LSTM by a wide margin.

Parser	Accuracy (%)
LSTM + LSTM	93.33
Bidirectional LSTM + LSTM	97.24

Table 5: Topological field labeling accuracies. The addition of backward flowing information improves accuracy considerably.

Table 6 shows the labeled attachment scores (LAS) for parsing with topological fields. As we can see, adding gold topological field annotations provides a marked improvement over parsing without topological fields. Although the parser does not achieve quite the same performance with the output of the LSTM-based sequence labeler, it is still a relatively large improvement over the parser of De Kok (2015). All differences are significant at $p < 0.0001$.⁵

Parser	LAS	UAS
De Kok (2015)	89.49	91.88
Neural net + TFs	90.00	92.36
Neural net + gold TFs	90.42	92.76

Table 6: Parse results with topological fields and gold topological fields. Parsers that use topological field information outperform parsers without access to such information.

6 Result analysis

Our motivation for introducing topological fields in dependency parsing is to provide the parser with

⁴The software is available from: <https://github.com/danieldk/toponn>

⁵Using paired approximate randomization tests (Noreen, 1989).

a more global view of sentence structure (Section 2). If this is indeed the case, we expect the parser to improve especially for longer-distance relations. Figure 1 shows the improvement in LAS as a result of adding gold-standard topological fields. We see a strong relation between the relation length and the improvement in accuracy. The introduction of topological fields clearly benefits the attachment of longer-distance dependents.

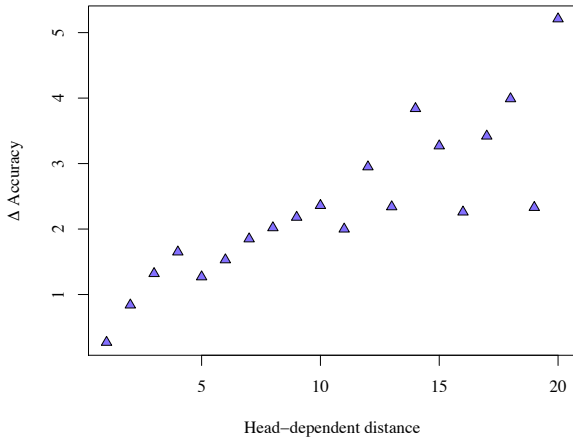


Figure 1: The improvement in labeled attachment score as a result of adding gold topological fields to the parser by dependency length.

Since the introduction of topological fields has very little impact on short-distance relations, the differences in the attachment of relations that virtually never cross fields (Table 2) turn out to be negligible. However, for the relations that cross fields frequently, we see a marked improvements (Table 7) for every relation except the *prepositional object*. In hindsight, this difference should not be surprising — the relations that never cross fields are usually very local, while those that almost always cross fields tend to have longer distances and/or are subject to relatively free ordering.

Dependency label	LAS Δ
Expletive <i>es</i>	2.71
Separated verb prefix	1.64
Subject	1.22
Prepositional object	-0.29
Direct object	1.59

Table 7: The LAS Δ of the parser with access to gold standard topological fields compared to the De Kok (2015) parser for the relations of Table 4.

Dependency label	LAS Δ
Coordinating conjunction (clausal)	11.48
Parenthesis	8.31
Dependent clause	3.49
Conjunct	3.38
Sentence root ⁷	2.92
Expletive <i>es</i>	2.71
Sentence	2.64
Comparative	1.87
Separated verb prefix	1.64
Direct object	1.59

Table 8: The ten dependency relations with the highest LAS Δ of the parser with access to gold topological fields compared to the (de Kok, 2015) parser.

The ten dependency relations with the highest overall improvement in LAS are shown in Table 8. Many of these relations are special when it comes to topological field structure and were not discussed in Section 2. The relations *parenthesis*, *dependent clause*, and *sentence* link two clauses; the *sentence root* marks the root of the dependency tree; and the *coordinating conjunction (clausal)* relation attaches a token that is always in its own field.⁶ This confirms that the addition of topological fields also improves the analysis of the overall clausal structure.

7 Conclusion and outlook

In this paper, we have argued and shown that access to topological field information can improve the accuracy of transition-based dependency parsers. In future, we plan to see how competitive the bidirectional LSTM-based sequence labeling approach is compared to existing approaches. Moreover, we plan to evaluate the use of topological fields in the architecture proposed by Dyer et al., (2015) to see how many of these regularities that approach captures.

Acknowledgments

The authors gratefully acknowledge the financial support of their research by the German Ministry for Education and Research (BMBF) as part of the CLARIN-D research infrastructure grant given to the University of Tübingen. Furthermore, we would like to thank Jianqiang Ma for his extensive comments on an early draft of this paper.

⁶The *KOORD* field, see Telljohan et al. (2006).

References

- Markus Becker and Anette Frank. 2002. A stochastic topological parser for German. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.
- Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750.
- Jackie Chi Kit Cheung and Gerald Penn. 2009. Topological field parsing of German. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 64–72. Association for Computational Linguistics.
- François Chollet. 2015. Keras. <https://github.com/fchollet/keras>.
- Daniël de Kok. 2015. A poor man’s morphology for German transition-based dependency parsing. In *International Workshop on Treebanks and Linguistic Theories (TLT14)*.
- Erich Drach. 1937. *Grundgedanken der Deutschen Satzlehre*. Frankfurt/Main.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.
- Oskar Erdmann. 1886. *Grundzüge der deutschen Syntax nach ihrer geschichtlichen Entwicklung dargestellt*. Stuttgart: Cotta. Erste Abteilung.
- Alex Graves and Jürgen Schmidhuber. 2005. Frame-wise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.
- Simon Herling. 1821. Über die Topik der deutschen Sprache. In *Abhandlungen des frankfurterischen Gelehrtenvereins für deutsche Sprache*, pages 296–362, 394. Frankfurt/Main. Drittes Stück.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Tilman Höhle. 1986. Der Begriff ‘Mittelfeld’. Anmerkungen über die Theorie der topologischen Felder. In A. Schöne, editor, *Kontroversen alte und neue. Akten des 7. Internationalen Germanistenkongresses Göttingen*, pages 329–340. Tübingen: Niemeyer.
- Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127.
- Sandra Kübler. 2005. How do treebank annotation schemes influence parsing results? or how not to compare apples and oranges. *Proceedings of RANLP 2005*.
- Martina Liepert. 2003. Topological fields chunking for German with SVM’s: Optimizing SVM-parameters with GA’s. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.
- Wolfgang Maier. 2006. Annotation schemes and their influence on parsing results. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 19–24. Association for Computational Linguistics.
- Gereon Müller. 1999. Optimality, markedness, and word order in German. *Linguistics*, 37(5):777–818.
- Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.
- Eric W Noreen. 1989. Computer intensive methods for hypothesis testing: An introduction.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Heike Telljohann, Erhard W Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2006. Stylebook for the tübingen treebank of written German (TüBa-D/Z). In *Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany*.
- Hans Uszkoreit. 1984. *Word order and constituent structure in German*. CSLI Publications.
- Jorn Veenstra, Frank Henrik Müller, and Tylman Ule. 2002. Topological field chunking for German. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20, COLING-02*, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Yannick Versley. 2005. Parser evaluation across text types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*.
- Andrea Weber and Karin Müller. 2004. Word order variation in German main clauses: A corpus analysis. In *Proceedings of the 20th International conference on Computational Linguistics*, pages 71–77.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–571. Association for Computational Linguistics.

A Hyperparameters

The topological field labeler was trained using Keras (Chollet, 2015). Here, we provide a short overview the hyperparameters that we used:

- **Solver:** *rmsprop*, this solver is recommended by the Keras documentation for recurrent neural networks. The solver is used with its default parameters.
- **Learning rate:** the learning rate was determined by the function $0.01(1 + 0.02^i)^{-2}$, where i is the epoch. The intuition was to start with some epochs with a high learning rate, dropping the learning rate quickly. The results were not drastically different when using a constant learning rate of 0.001.
- **Epochs:** The models was trained for 200 epochs, then we picked the model of the epoch with the highest performance on the validation data (27 epochs for the unidirectional LSTM, 124 epochs for the bidirectional LSTM).
- **LSTM layers:** all LSTM layers were trained with 50 output dimensions. Increasing the number of output dimensions did not provide an improvement.
- **Regularization:** 10% dropout (Srivastava et al., 2014) was used after each LSTM layer for regularization. A stronger dropout did not provide better performance.

B Topological field projection algorithm

Algorithm 1 Topological field projection.

```

function PROJECT(node,field)
  if IS_TERMINAL_NODE(node) then
    node.field ← field
  else
    if IS_TOPO_NODE(node) then
      field ← node.field
    end if
    for child ∈ node do
      PROJECT(child,field)
    end for
  end if
end function

```
