Extraposition: A case study in German sentence realization

Michael GAMON[†], Eric RINGGER[†], Zhu ZHANG[‡],
Robert MOORE[†], Simon CORSTON-OLIVER[†]

[†]Microsoft Research
Microsoft Corporation
Redmond, WA 98052
{mgamon, ringger, bobmoore,
simonco}@microsoft.com

[‡]University of Michigan
Ann Arbor, MI 48109
zhuzhang@umich.edu

Abstract

We profile the occurrence of clausal extraposition in corpora from different domains and demonstrate that extraposition is a pervasive phenomenon in German that must be addressed in German sentence realization. We present two different approaches to the modeling of extraposition, both based on machine learned decision tree classifiers. The two approaches differ in their view of the movement operation: one approach models multi-step movement through intermediate nodes to the ultimate target node, while the other approach models one-step movement to the target node. We compare the resulting models, trained on data from two domains and discuss the differences between the two types of models and between the results obtained in the different domains.

Introduction

Sentence realization, the last stage in natural language generation, derives a surface string from a more abstract representation. Numerous complex operations are necessary to produce fluent output, including syntactic aggregation, constituent ordering, word inflection, etc. We argue that for fluent output from German sentence realization, clausal extraposition needs to be included. We show how to accomplish this task by applying machine learning techniques.

A comparison between English and German illustrates that it is possible in both languages to extrapose clausal material to the right periphery of a clause, as the following examples show:

Relative clause extraposition:

English: *A man just left who had come to ask a question.*

German: *Der Mann ist gerade weggegangen, der gekommen war, um eine Frage zu stellen.*

Infinitival clause extraposition:

English: *A decision was made to leave the country.*

German: *Eine Entscheidung wurde getroffen, das Land zu verlassen.*

Complement clause extraposition:

English: *A rumor has been circulating that he is ill.*

German: *Ein Gerücht ging um, dass er krank ist.*

Unlike obligatory movement phenomena such as Wh-movement, extraposition is subject to pragmatic variability. A widely-cited factor influencing extraposition is clausal heaviness; in general, extraposition of heavy clauses is preferred over leaving them in place. Consider the following example from the technical domain:

German: *Es werden Datenstrukturen verwendet, die für die Benutzer nicht sichtbar sind.*

English: *Data structures are used which are not visible to the user.*

This perfectly fluent sentence contains an extraposed relative clause. If the relative clause is left in place, as in the following example, the result is less fluent, though still grammatical:

? Es werden Datenstrukturen, die für die Benutzer nicht sichtbar sind, verwendet.
Data structures which are not visible to the users are used.

Table 1 presents a quantitative analysis of the frequency of extraposition in different corpora in both English and German. This analysis is based on automatic data profiling using the NLPWin system (Heidorn 2000). The technical manual corpus consists of 100,000 aligned English-German sentence pairs from Microsoft technical manuals. The Encarta corpora consist of 100,000 randomly selected sentences from the Encarta encyclopedia in both English and German. The output of the parser was post-processed to identify relative clauses (RELCL), infinitival clauses (INFCL), and complement clauses (COMPCL) that have been moved from a position adjacent to the term they modify. According to this data profile, approximately one third of German relative clauses are extraposed in technical writing, while only 0.22% of English relative clauses are extraposed in the corresponding sentence set. The high number of extraposed relative clauses in German is corroborated by numbers from the German hand-annotated NEGRA corpus. In NEGRA, 26.75% of relative clauses are extraposed. Uszkoreit et al. (1998) report 24% of relative clauses being extraposed in NEGRA, but their number is based on an earlier version of NEGRA, which is about half the size of the current NEGRA corpus.

We also used the NEGRA corpus to verify the accuracy of our data profiling with NLPWin. These results are presented in Table 2. We only took into account sentences that received a complete parse in NLPWin. Of the 20,602 sentences in NEGRA, 17,756 (86.19%) fell into that category. The results indicate that NLPWin is sufficiently reliable for the identification of relative clauses to make our conclusions noteworthy and to make learning from NLPWin-parsed data compelling.

Extraposition is so rare in English that a sentence realization module may safely ignore it and still yield fluent output. The fluency of sentence realization for German, however, will suffer from the lack of a good extraposition mechanism.

	German technical manuals	English technical manuals	German Encarta	English Encarta
RELCL	34.97%	0.22%	18.97%	0.30%
INFCL	3.2%	0.53%	2.77%	0.33%
COMPCL	1.50%	0.00%	2.54%	0.15%

Table 1: Percentage of extraposed clauses in English and German corpora

Relative clause identification overall		Identification of extraposed relative clauses		Identification of non-extraposed relative clauses	
Recall	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**
94.55	93.40	74.50	90.02	94.64	87.76

Table 2: NLPWin recall and precision for relative clauses on the NEGRA corpus

This evidence makes it clear that any serious sentence realization component for German needs to be able to produce extraposed relative clauses in order to achieve reasonable fluency. In the German sentence realization module, code-named Amalgam (Gamon et al. 2002, Corston-Oliver et al. 2002), we have successfully implemented both extraposition models as described here.

1 Two strategies for modeling extraposition

The linguistic and pragmatic factors involved in clause extraposition are inherently complex. We use machine learning techniques to leverage large amounts of data for discovering the relevant conditioning features for extraposition. As a machine learning technique for the problem at

hand, we chose decision tree learning, a practical approach to inductive inference in widespread use. We employ decision tree learning to approximate discrete-valued functions from large feature sets that are robust to noisy data. Decision trees provide an easily accessible inventory of the selected features and some indication of their relative importance in predicting the target features in question. The particular tool we used to build our decision trees is the WinMine toolkit (Chickering *et al.*, 1997, n.d.). Decision trees built by WinMine predict a probability distribution over all possible target values.

We consider two different strategies for the machine-learned modeling of extraposition. The two strategies are a series of movements versus a single reattachment.

1.1 Multi-step movement

In the multi-step movement approach, the question to model for each potential attachment site of an extraposable clause is whether the clause should move up to its grandparent (a "yes" answer) or remain attached to its current parent (a "no" answer). In other words, we have cast the problem as a staged classification task. At generation runtime, for a given extraposable clause, the movement question is posed, and if the DT classifier answers "yes", then the clause is reattached one level up, and the question is posed again. The final attachment site is reached when the answer to the classification task is "no", and hence further movement is barred. Figure 1 illustrates the multi-step movement of a clause (lower triangle) through two steps to a new landing site (the reattached clause is the upper triangle). Note that in both Figure 1 and Figure 2 linear order is ignored; only the hierarchical aspects of extraposition are represented.

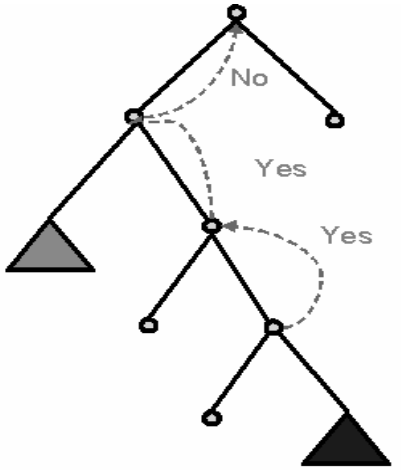

Figure 1: Multi-step movement

1.2 One-step movement

Modeling extraposition as a one-step movement involves a classification decision for each node in the parent chain of an extraposable clause. The classification task can be formulated as "should the extraposable clause move up to this target from its base position?". Figure 2 shows the one-step movement approach to extraposition in the same structural configuration as in Figure 1. In this example, out of the three potential landing sites, only one qualifies. At generation runtime, if more than one node in the parent chain qualifies as a target for extraposition movement, the node with the highest probability of being a target is chosen. In the event of equally likely target nodes, the target node highest in the tree is chosen.

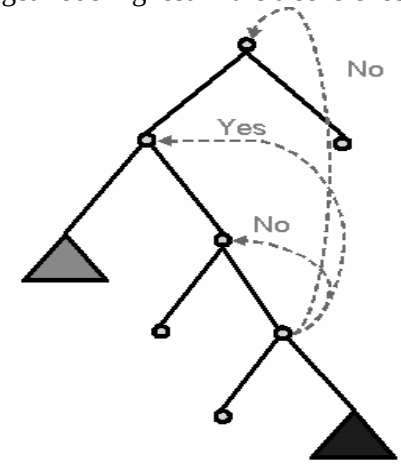

Figure 2: One-step movement

2 Data and features

We employed two different sets of data to build the models for German: the 100,000 sentence technical manual corpus, and the 100,000 sentence Encarta corpus. The data were split 70/30 for training and parameter tuning purposes, respectively. We extracted features for each data point, using the syntactic and semantic analysis provided by the Microsoft NLPWin system (see Gamon et al. 2002 for more details). We only considered sentences for feature extraction which received a complete spanning parse in NLPwin. 85.14% of the sentences in the technical domain, and 88.37% of the sentences in the Encarta corpus qualified. The following features were extracted:

- syntactic label of the node under consideration (i.e., the starting node for a single-step movement), its parent and grandparent, and the extraposable clause
- semantic relation to the parent node of the node under consideration, the parent and the grandparent, and the extraposable clause
- status of the head of the node under consideration as a separable prefix verb, the same for the parent and the grandparent
- verb position information (verb-second versus verb-final) for the node under consideration, the parent and grandparent
- all available analysis features and attributes in NLPWin (see Gamon et al. 2002 for a complete list of the currently used features and attributes) on the node under consideration, the parent and grandparent, and on the extraposable clause and its parent and grandparent
- two features indicating whether the extraposable node has any verbal ancestor node with verb-final or verb-second properties
- "heaviness" of extraposable clause as measured in both number of words and number of characters
- "heaviness" of the whole sentence as measured in both number of words and number of characters

A total of 1397 features were extracted for the multi-step movement model. For the single-step movement model, we extracted an additional 21 features. Those features indicate for each of the 21 labels for non-terminal nodes whether a node with that label intervenes between the parent of the extraposable clause and the putative landing site.

Another linguistic feature commonly cited as influencing extraposition is the length and complexity of the part of the structure between the original position and the extraposed clause. Since in the Amalgam generation module extraposition is applied before word and constituent order is established, length of intervening strings is not accessible as a feature.

For each training set, we built decision trees at varying levels of granularity (by manipulating the prior probability of tree structures to favor simpler structures) and selected the model with maximal accuracy on the corresponding parameter tuning data set.

Since the syntactic label of the extraposable clause is one of the extracted features, we decided to build one general extraposition model, instead of building separate models for each of the three extraposable clause types (complement clause *COMPCL*, infinitival clause *INFCL*, and relative clause *RELCL*). If different conditions apply to the three types of extraposition, the decision tree model is expected to pick up on the syntactic label of the extraposable clause as a predictive feature. If, on the other hand, conditions for extraposition tend to be neutral with respect to the type of extraposable clause, the modeling of INFCL and COMPCL extraposition can greatly benefit from the much larger set of data points in relative clause extraposition.

3 Comparison

To compare the one-step and multi-step models, we processed a new blind test set of 10,000 sentences from each domain, Microsoft technical manuals and Encarta, respectively. These sentences were extracted randomly from data in these domains that were neither included in the training nor in the parameter tuning set. For each extraposable clause, three different outputs were computed: the observed behavior, the prediction obtained by iteratively applying the multi-step model as described in Section 1.1, and the prediction obtained by applying the one-step

model. The values for these outputs were either "no extraposition" or a specific target node. If either the general extraposition prediction or the predicted specific target node did not match the observed behavior, this was counted as an error.

3.1 One-step versus multi-step in the technical domain

Accuracy data on a blind set of 10,000 sentences from the technical manuals domain are presented in Table 3.

	One-step	Multi-step	Baseline
RELCL	81.56%	83.87%	60.93%
INFCL	93.70%	92.02%	93.70%
COMPCL	98.10%	98.57%	94.29%
Overall	84.42%	86.12%	67.58%

Table 3: Accuracy numbers for the two models in the technical domain

The baseline score is the accuracy for a system that never extraposes. Both models outperform the overall baseline by a large margin; the multi-step movement model achieves an accuracy 1.7% higher than the one-step model. The baselines in INFCL and COMPCL extraposition are very high. In the test set there were only 15 cases of extraposed INFCLs and 12 cases of extraposed COMPCLs, making it impossible to draw definite conclusions.

3.2 One-step versus multi-step in the Encarta domain

Results from a blind test set of 10,000 sentences from the Encarta domain are presented in Table 4.

	One-step	Multi-step	Baseline
RELCL	87.59%	88.45%	80.48%
INFCL	97.73%	97.48%	95.72%
COMPCL	97.32%	97.32%	95.97%
Overall	89.99%	90.61%	84.15%

Table 4: Accuracy numbers for the two models in the Encarta domain

As in the technical domain, the multi-step model outperforms the one-step model, and both outperform the baseline significantly. Again, extraposed COMPCLs and INFCLs are rare in the dataset (there were only 17 and 6 instances, respectively), making the results on these types of clauses inconclusive.

3.3 Domain-specificity of the models

Since we have data from two very different domains we considered the extent to which the domain-specific models overlapped. This is a linguistically interesting question: from a linguistic perspective one would expect both universal properties of extraposition as well as domain specific generalizations to emerge from such a comparison.

3.3.1 Feature selection in the technical domain versus Encarta

Of the 1397 features that were extracted for the multi-step model, the best model for the technical domain was created by the WinMine tools by selecting 60 features. In the Encarta domain, 49 features were selected. 27 features are shared by the two models. This overlap in selected features indicates that the models indeed capture linguistic generalizations that are valid across domains. The shared features fall into the following categories (where *node* refers to the starting node for multi-step movement):

- features relating to verbal properties of the node
  - a separable prefix verb as ancestor node
  - tense and mood of ancestor nodes
  - presence of a verb-final or verb-second VP ancestor
  - presence of Modals attribute (indicating the presence of a modal verb) on ancestors
  - verb-position in the current node and ancestors
- "heaviness"-related features on the extraposable clause and the whole sentence:
  - sentence length in characters
  - number of words in the extraposable clause
- syntactic labels
- the presence of a prepositional relation
- the presence of semantic subjects and objects on the node and ancestors
- definiteness features
- the presence of modifiers on the parent
- person and number features

- some basic subcategorization features (e.g., transitive versus intransitive)

Interestingly, the features that are not shared (33 in the model for the technical domain and 27 in the model for the Encarta domain) fall roughly into the same categories as the features that are shared. To give some examples:

- The Encarta model refers to the presence of a possessor on the parent node, the technical domain model does not.
- The technical domain model selects more person and number features on ancestors of the node and ancestors of the extraposable clause than the Encarta model.

For the one-step model, 1418 total features were extracted. Of these features, the number of features selected as being predictive is 49 both in the Encarta and in the technical domain. Twenty-eight of the selected features are shared by the models in the two domains. Again, this overlap indicates that the models do pick up on linguistically relevant generalizations.

The shared features between the one-step models fall into the same categories as the shared features between the multi-step models.

The results from these experiments suggest that the categories of selected features are domain-independent, while the choice of individual features from a particular category depends on the domain.

3.3.2 Model complexity

In order to assess the complexity of the models, we use the simple metric of number of branching nodes in the decision tree. The complexity of the models clearly differs across domains. Table 5 illustrates that for both multi-step and one-step movement the model size is considerably smaller in the Encarta domain versus the technical domain.

	One-step	**Multi-step**
Encarta	68	82
Technical	87	116

Table 5: Number of branching nodes in the decision trees

We hypothesize that this difference in model complexity may be attributable to the fact that NLPWin assigns a higher percentage of spanning parses to the Encarta data, indicating that in general, the Encarta data may yield more reliable parsing output.

3.3.3 Cross-domain accuracy

The results in Table 3 and Table 4 above show that the models based on the Encarta domain achieve a much higher overall accuracy (89.99% and 90.61%) than the models based on the technical domain (84.42% and 86.12%), but they are also based on a much higher baseline of non-extraposed clauses (84.15% versus 67.58% in the technical domain). To quantify the domain specificity of the models, we applied the models across domains; i.e., we measured the performance of the Encarta models on the technical domain and vice versa. The results contrasted with the in-domain overall accuracy from Table 3 and Table 4 are given in Table 6.

	Encarta Model		**Technical Model**	
	1-step	**Multi**	**1-step**	**Multi**
On Enc.	89.99%	90.61%	84.42%	86.12%
On Tech.	79.39%	83.03%	88.54%	89.20%

Table 6: Cross-domain accuracy of the models

The results show that for both one-step and multi-step models, the models trained on a given domain will outperform the models trained on a different domain. These results are not surprising; they confirm domain-specificity of the phenomenon. Viewed from a linguistic perspective, this indicates that the generalizations governing clausal extraposition cannot be formulated independently of the text domain.

Conclusion

We have shown that it is possible to model extraposition in German using decision tree classifiers trained on automatic linguistic analyses of corpora. This method is particularly effective for extraposed relative clauses, which are pervasive in German text in domains as disparate as news, technical manuals, and encyclopedic text. Both one-step and multi-step models very clearly outperform the baseline in the two domains in which we experimented. This in itself is a significant result, given the complexity of the linguistic phenomenon of clausal extraposition. The machine learning

approach to extraposition has two clear advantages: it eliminates the need for hand-coding of complex conditioning environments for extraposition, and it is adaptable to new domains. The latter point is supported by the cross-domain accuracy experiment and the conclusion that extraposition is governed by domain-specific regularities.

We have shown that across domains, the multi-step model outperforms the one-step model. In the German sentence realization system code-named *Amalgam* (Corston-Oliver et al. 2002, Gamon et al. 2002), we have experimented with implementations of both the one-step and multi-step extraposition models, and based on the results reported here we have chosen the multi-step model for inclusion in the end-to-end system.

As we have shown, extraposed relative clauses outnumber other extraposed clause types by a large margin. Still, the combined model for clausal extraposition outperforms the baseline even for infinitival clauses and complement clauses, although the conclusions here are not very firm, given the small number of relevant data points in the test corpus. Since the syntactic label of the extraposed clause is one of the features extracted from the training data, however, the setup that we have used will adapt easily once more training data (especially for infinitival and complement clauses) become available. The models will automatically pick up distinctions between the generalizations covering relative clauses versus infinitival/complement clauses when they become relevant, by selecting the syntactic label feature as predictive.

Finally, evaluation of the types of features that were selected by the extraposition models show that besides the “heaviness” of the extraposed clause, a number of other factors from the structural context enter the determination of likelihood of extraposition. This, in itself, is an interesting result: it shows how qualitative inspection of a machine learned model can yield empirically based linguistic insights.

Acknowledgements

Our thanks go to Max Chickering for his assistance with the WinMine toolkit and to the anonymous reviewers for helpful comments.

References

Chickering D. M., Heckerman D. and Meek C. (1997). *A Bayesian approach to learning Bayesian networks with local structure.* In "Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference", D. Geiger and P. Punadlik Shenoy, ed., Morgan Kaufman, San Francisco, California, pp. 80-89.

Chickering, D. Max. nd. WinMine Toolkit Home Page. http://research.microsoft.com/~dmax/WinMine/Tooldoc.htm

Corston-Oliver S., Gamon M., Ringger E. and Moore R. (2002). *An overview of Amalgam: a machine-learned generation module.* To appear in Proceedings of the Second International Natural Language Generation Conference 2002, New York.

Gamon M., Ringger E., Corston-Oliver S.. (2002). *Amalgam: A machine-learned generation module.* Microsoft Technical Report MSR-TR-2002-57.

Heidorn, G. E. (2000): *Intelligent Writing Assistance.* In "A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text", R. Dale, H. Moisl, and H. Somers (ed.), Marcel Dekker, New York, pp. 181-207.

Uszkoreit, H., Brants T., Duchier D., Krenn B., Konieczny L., Oepen S. and Skut W. (1998). *Aspekte der Relativsatzextraposition im Deutschen.* Claus-Report Nr.99, Sonderforschungsbereich 378, Universität des Saarlandes, Saarbrücken, Germany.