Machine-learned contexts for linguistic operations in German sentence realization

Michael GAMON, Eric RINGGER, Simon CORSTON-OLIVER, Robert MOORE
Microsoft Research
Microsoft Corporation
Redmond, WA 98052
{mgamon, ringger, simonco, bobmoore}@microsoft.com

Abstract

We show that it is possible to learn the contexts for linguistic operations which map a semantic representation to a surface syntactic tree in sentence realization with high accuracy. We cast the problem of learning the contexts for the linguistic operations as classification tasks, and apply straightforward machine learning techniques, such as decision tree learning. The training data consist of linguistic features extracted from syntactic and semantic representations produced by a linguistic analysis system. The target features are extracted from links to surface syntax trees. Our evidence consists of four examples from the German sentence realization system code-named *Amalgam*: case assignment, assignment of verb position features, extraposition, and syntactic aggregation

1 Introduction

The last stage of natural language generation, sentence realization, creates the surface string from an abstract (typically semantic) representation. This mapping from abstract representation to surface string can be direct, or it can employ intermediate syntactic representations which significantly constrain the output. Furthermore, the mapping can be performed purely by rules, by application of statistical models, or by a combination of both techniques.

Among the systems that use statistical or machine learned techniques in sentence realization, there are various degrees of intermediate syntactic structure. Nitrogen (Langkilde and Knight, 1998a, 1998b) produces a large set of alternative surface realizations of an input structure (which can vary in abstractness). This set of candidate surface strings, represented as a word lattice, is then rescored by a word-bigram language model, to produce the best-ranked output sentence. FERGUS (Bangalore and Rambow, 2000), on the other hand, employs a model of syntactic structure during sentence realization. In simple terms, it adds a tree-based stochastic model to the approach taken by the Nitrogen system. This tree-based model chooses a best-ranked XTAG representation for a given dependency structure. Possible linearizations of the XTAG representation are generated and then evaluated by a language model to pick the best possible linearization, as in Nitrogen.

In contrast, the sentence realization system code-named *Amalgam* (*A Ma*chine *L*earned *G*ener*a*tion *M*odule) (Corston-Oliver et al., 2002; Gamon et al., 2002b) employs a series of linguistic operations which map a semantic representation to a surface syntactic tree via intermediate syntactic representations. The contexts for most of these operations in Amalgam are machine learned. The resulting syntactic tree contains all the necessary information on its leaf nodes from which a surface string can be read.

The goal of this paper is to show that it is possible to learn accurately the contexts for linguistically complex operations in sentence realization. We propose that learning the contexts for the application of these linguistic operations can be viewed as per-operation classification problems. This approach combines advantages of a linguistically informed approach to sentence realization with the advantages of a machine

learning approach. The linguistically informed approach allows us to deal with complex linguistic phenomena, while machine learning automates the discovery of contexts that are linguistically relevant and relevant for the domain of the data. The machine learning approach also facilitates adaptation of the system to a new domain or language. Furthermore, the quantitative nature of the machine learned models permits finer distinctions and ranking among possible solutions.

To substantiate our claim, we provide four examples from Amalgam: assignment of case, assignment of verb position features, extraposition, and syntactic aggregation.

2 Overview of Amalgam

Amalgam takes as its input a sentence-level semantic graph representation with fixed lexical choices for content words (the logical form graph of the NLPWin system – see (Heidorn, 2000)). This representation is first degraphed into a tree, and then gradually augmented by the insertion of function words, assignment of case and verb position features, syntactic labels, etc., and transformed into a syntactic surface tree. A generative statistical language model establishes linear order in the surface tree (Ringger et al., in preparation), and a surface string is generated from the leaf nodes. Amalgam consists of eight stages. We label these ML (machine-learned context) or RB (rule-based).

Stage 1 Pre-processing (RB):
- degraphing of the semantic representation
- retrieval of lexical information

Stage 2 Flesh-out (ML):
- assignment of syntactic labels
- insertion of function words
- assignment of case and verb position features

Stage 3 Conversion to syntactic tree (RB):
- introduction of syntactic representation for coordination
- splitting of separable prefix verbs based on both lexical information and previously assigned verb position features
- reversal of heads (e.g., in quantitative expressions) (ML)

Stage 4 Movement:
- extraposition (ML)
- raising, wh movement (RB)

Stage 5 Ordering (ML):
- ordering of constituents and leaf nodes in the tree

Stage 6 Surface cleanup (ML):
- lexical choice of determiners and relative pronouns
- syntactic aggregation

Stage 7 Punctuation (ML)

Stage 8 Inflectional generation (RB)

All machine learned components, with the exception of the generative language model for ordering of constituents (stage 5), are decision tree classifiers built with the WinMine toolkit (Chickering et al., 1997; Chickering, nd.). There are a total of eighteen decision tree classifiers in the system. The complexity of the decision trees varies with the complexity of the modeled task. The number of branching nodes in the decision tree models in Amalgam ranges from 3 to 447.

3 Data and feature extraction

The data for all of the models were drawn from a set of 100,000 sentences from technical software manuals and help files. The sentences are analyzed by the NLPWin system, which provides a syntactic and logical form analysis. Nodes in the logical form representation are linked to the corresponding syntactic nodes, allowing us to learn contexts for the mapping from the semantic representation to a surface syntax tree. The data is split 70/30 for training versus model parameter tuning. For each set of data we built decision trees at several different levels of granularity (by manipulating the prior probability of tree structures to favor simpler structures) and selected the model with the maximal accuracy as determined on the parameter tuning set. All models are then tested on data extracted from a separate blind set of 10,000 sentences from the same domain. For both training and test, we only extract features from sentences that have received a complete, spanning parse: 85.14% of the sentences in the training and parameter tuning set, and 84.59% in the blind test set fall into that category. Most sentences yield more than one training case.

We attempt to standardize as much as possible the set of features to be extracted. We exploit the full set of features and attributes available in the analysis, instead of pre-determining a small set of

potentially relevant features (Gamon et al., 2002b). This allows us to share the majority of code between the individual feature extraction tasks. More importantly, it enables us to discover new linguistically interesting and/or domain-specific generalizations from the data. Typically, we extract the full set of available analysis features of the node under investigation, its parent and its grandparent, with the only restriction being that these features need to be available at the stage where the model is consulted at generation run-time. This provides us with a sufficiently large structural context for the operations. In addition, for some of the models we add a small set of features that we believe to be important for the task at hand, and that cannot easily be expressed as a combination of analysis features/attributes on constituents. Most features, such as lexical subcategorization features and semantic features such as [Definite] are binary. Other features, such as syntactic label or semantic relation, have as many as 25 values. Training time on a standard 500MHz PC ranges from one hour to six hours.

4 Assignment of case

In German sentence realization, proper assignment of morphological case is essential for fluent and comprehensible output. German is a language with fairly free constituent order, and the identification of functional roles, such as subject versus object, is not determined by position in the sentence, as in English, but by morphological marking of one of the four cases: nominative, accusative, genitive or dative. In Amalgam, case assignment is one of the last steps in the Flesh-out stage (stage 2). Morphological realization of case can be ambiguous in German (for example, a feminine singular NP is ambiguous between accusative and nominative case). Since the morphological realization of case depends on the gender, number and morphological paradigm of a given NP, we chose to only consider NP nodes with unambiguous case as training data for the model[1]. As the target feature for this model is morphological case, it has four possible values for the four cases in German.

4.1 Features in the case assignment model

For each data point, a total of 712 features was extracted. Of the 712 features available to the decision tree building tools, 72 were selected as having predictive value in the model. The selected features fall into the following categories:

- syntactic label of the node, its parent and grandparent
- lemma (i.e., citation form) of the parent, and lemma of the governing preposition
- subcategorization information, including case governing properties of governing preposition and parent
- semantic relation of the node itself to its parent, of the parent to its grandparent, and of the grandparent to its great-grandparent
- number information on the parent and grandparent
- tense and mood on the parent and grandparent
- definiteness on the node, its parent and grandparent
- the presence of various semantic dependents such as subject, direct and indirect objects, operators, attributive adjuncts and unspecified modifiers on the node and its parent and grandparent
- quantification, negation, coordination on the node, the parent and grandparent
- part of speech of the node, the parent and the grandparent
- miscellaneous semantic features on the node itself and the parent

4.2 The case assignment model

The decision tree model for case assignment has 226 branching nodes, making it one of the most complex models in Amalgam. For each nominal node in the 10,000 sentence test set, we compared the prediction of the model to the

[1] Ideally, we should train the case assignment model on a corpus that is hand-disambiguated for case. In the absence of such a corpus, though, we believe that our approach is linguistically justified. The case of an NP depends solely on the syntactic context it appears in. Since we want to learn the syntactically determining factors for case, using unambiguously case marked NPs for training seems justified.

morphological case compatible with that node. The previously mentioned example of a singular feminine NP, for example, would yield a "correct" if the model had predicted nominative or accusative case (because the NP is morphologically ambiguous between accusative and nominative), and it would yield an "incorrect" if the model had predicted genitive or dative. This particular evaluation setup was a necessary compromise because of the absence of a hand-annotated corpus with disambiguated case in our domain. The caveat here is that downstream models in the Amalgam pipeline that pick up on case as one of their features rely on the absolute accuracy of the assigned case, not the relative accuracy with respect to morphological ambiguity. Accuracy numbers for each of the four case assignments are given in Table 1. Note that it is impossible to give precision/recall numbers, without a hand-disambiguated test set. The baseline for this task is 0.7049 (accuracy if the most frequent case (nominative) had been assigned to all NPs).

Table 1. Accuracy of the case assignment model.

Value	Accuracy
Dat	0.8705
Acc	0.9707
Gen	0.9457
Nom	0.9654
overall	0.9352

5 Assignment of verb position features

One of the most striking properties of German is the distributional pattern of verbs in main and subordinate clauses. Most descriptive accounts of German syntax are based on a topology of the German sentence that treats the position of the verb as the fixed frame around which other syntactic constituents are organized in relatively free order (cf. Eisenberg, 1999; Engel, 1996). The position of the verb in German is non-negotiable; errors in the positioning of the verb result in gibberish, whereas most permutations of other constituents only result in less fluent output.

Depending on the position of the finite verb, German sentences and verb phrases are classified as being "verb-initial", "verb-second" or "verb-final". In verb-initial clauses (e.g., in imperatives), the finite verb is in initial position. Verb-second sentences contain one constituent preceding the finite verb, in the so-called "pre-field". The finite verb is followed by any number of constituents in the "middle-field", and any non-finite verbs are positioned at the right periphery of the clause, possibly followed by extraposed material or complement clauses (the "post-field"). Verb-final clauses contain no verbal element in the verb-second position: all verbs are clustered at the right periphery, preceded by any number of constituents and followed only by complement clauses and extraposed material.

During the Flesh-out stage in Amalgam, a decision tree classifier is consulted to make a classification decision among the four verb positions: "verb-initial", "verb-second", "verb-final", and "undefined". The value "undefined" for the target feature of verb position is extracted for those verbal constituents where the local syntactic context is too limited to make a clear distinction between initial, second, or final position of the verb. The number of "undefined" verb positions is small compared to the number of clearly established verb positions: in the test set, there were only 690 observed cases of "undefined" verb position out of a total of 15,492 data points. At runtime in Amalgam, verb position features are assigned based on the classification provided by the decision tree model.

5.1 Features in the verb position model

For each data point, 713 features were extracted. Of those features, 41 were selected by the decision tree algorithm. The selected features fall into the following categories:

- syntactic label of the node and the parent
- subcategorization features
- semantic relations of the node to its parent and of the parent node to its parent
- tense and mood features
- presence of empty, uncontrolled subject
- semantic features on the node and the parent

5.2 The verb position model

The decision tree model for verb position has 115 branching nodes. Precision, recall and F-

measure for the model are given in Table 2. As a point of reference for the verb position classifier, assigning the most frequent value (second) of the target feature yields a baseline score of 0.4240.

Table 2. Precision, recall, and F-measure for the verb position model.

Value	Precision	Recall	F-measure
Initial	0.9650	0.9809	0.9729
Second	0.9754	0.9740	0.9743
Final	0.9420	0.9749	0.9581
Undefined	0.5868	0.3869	0.4663
Overall accuracy	0.9491		

6 Extraposition

In both German and English it is possible to extrapose clausal material to the right periphery of the sentence (extraposed clauses underlined in the examples below):

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 rumour has been circulating that he is ill.*

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

Extraposition is not obligatory like other types of movement (such as Wh-movement). Both extraposed and non-extraposed versions of a sentence are acceptable, with varying degrees of fluency.

The interesting difference between English and German is the frequency of this phenomenon. While it can easily be argued that English sentence realization may ignore extraposition and still result in very fluent output, the fluency of sentence realization for German will suffer much more from the lack of a good extraposition mechanism. We profiled data from various domains (Gamon et al. 2002a) to substantiate this linguistic claim (see Uszkoreit et al. 1998 for similar results). In the technical domain, more than one third of German relative clauses are extraposed, as compared to a meagre 0.22% of English relative clauses. In encyclopaedia text (Microsoft Encarta), approximately every fifth German relative clause is extraposed, compared to only 0.3% of English relative clauses. For complement clauses and infinitival clauses, the differences are not as striking, but still significant: in the technical and encyclopaedia domains, extraposition of infinitival and complement clauses in German ranges from 1.5% to 3.2%, whereas English only shows a range from 0% to 0.53%.

We chose to model extraposition as an iterative movement process from the original attachment site to the next higher node in the tree (for an alternative one-step solution and a comparison of the two approaches see (Gamon et al., 2002a)). The target feature of the model is the answer to the yes/no question "Should the clause move from node X to the parent of node X?".

6.1 Features in the extraposition model

The tendency of a clause to be extraposed depends on properties of both the clause itself (e.g., some notion of "heaviness") and the current attachment site. Very coarse linguistic generalizations are that a relative clause tends to be extraposed if it is sufficiently "heavy" and if it is followed by verbal material in the same clause. Feature extraction for this model reflects that fact by taking into consideration features on the extraposition candidate, the current attachment site, and potential next higher landing site. This results in a total of 1168 features. Each node in the parent chain of an extraposable clause, up to the actual attachment node, constitutes a single data point

During the decision tree building process, 60 features were selected as predictive. They can be classified as follows:

General feature:

- overall sentence length

Features on the extraposable clause:

- presence of verb-final and verb-second ancestor nodes
- "heaviness" both in number of characters and number of tokens

- various linguistic features in the local context (parent node and grandparent node): number and person, definiteness, voice, mood, transitivity, presence of logical subject and object, presence of certain semantic attributes, coordination, prepositional relations
- syntactic label
- presence of modal verbs
- prepositional relations
- transitivity

Features on the attachment site

- presence of logical subject
- status of the parent and grandparent as a separable prefix verb
- voice and presence of modal verbs on the parent and grandparent
- presence of arguments and transitivity features on the parent and grandparent
- number, person and definiteness; the same on parent and grandparent
- syntactic label; the same on the parent and grandparent
- verb position; the same on the parent
- prepositional relation on parent and grandparent
- semantic relation that parent and grandparent have to their respective parent node

6.2 The extraposition model

During testing of the extraposition model, the model was consulted for each extraposable clause to find the highest node to which that clause could be extraposed. In other words, the target node for extraposition is the highest node in the parent chain for which the answer to the classification task "Should the clause move from node X to the parent of node X?" is "yes" with no interceding "no" answer. The prediction of the model was compared with the actual observed attachment site of the extraposable clause to yield the accuracy figures shown in Table 3. The model has 116 branching nodes. The baseline for this task is calculated by applying the most frequent value for the target feature ("don't move") to all nodes. The baseline for extraposition of infinitival and complement clauses is very high. The number of extraposed clauses of both types in the test set (fifteen extraposed infinitival clauses and twelve extraposed complement clauses) is very small, so it comes as no surprise that the model accuracy ranges around the baseline for these two types of extraposed clauses.

Table 3. Accuracy of the extraposition model.

Extraposable clause	Accuracy	Baseline
RELCL	0.8387	0.6093
INFCL	0.9202	0.9370
COMPCL	0.9857	0.9429
Overall	0.8612	0.6758

7 Syntactic aggregation

Any sentence realization component that generates from an abstract semantic representation and strives to produce fluent output beyond simple templates will have to deal with coordination and the problem of duplicated material in coordination. This is generally viewed as a sub-area of aggregation in the generation literature (Wilkinson, 1995; Shaw, 1998; Reape and Mellish, 1999; Dalianis and Hovy, 1993). In Amalgam, the approach we take is strictly intra-sentential, along the lines of what has been called *conjunction reduction* in the linguistic literature (McCawley, 1988). While this may seem a fairly straightforward task compared to inter-sentential, semantic and lexical aggregation, it should be noted that the cross-linguistic complexity of the phenomenon makes it much less trivial than a first glance at English would suggest. In German, for example, position of the verb in the coordinated VPs plays an important role in determining which duplicated constituent can be omitted.

The target feature for the classification task is formulated as follows: "In which coordinated constituent is the duplicated constituent to be realized?". There are three values for the target feature: "first", "last", and "middle". The third value ("middle") is a default value for cases where neither the first, nor the last coordinated constituent can be identified as the location for the realization of duplicated constituents. At generation runtime, multiple realizations of a constituent in coordination are collected and the aggregation model is consulted to decide on the optimal position in which to realize that constituent. The constituent in that position is

retained, while all other duplicates are removed from the tree.

7.1 Features in the syntactic aggregation model

A total of 714 features were extracted for the syntactic aggregation model. Each instance of coordination which exhibits duplicated material at the semantic level without corresponding duplication at the syntactic level constitutes a data point.

Of these features, 15 were selected as predictive in the process of building the decision tree model:

- syntactic label and syntactic label of the parent node
- semantic relation to the parent of the duplicated node, its parent and grandparent
- part of speech of the duplicated node
- verb position across the coordinated node
- position of the duplicated node in premodifiers or postmodifiers of the parent
- coordination of the duplicated node and the grandparent of the duplicated node
- status of parent and grandparent as a proposition
- number feature on the parent
- transitivity and presence of a direct object on the parent

7.2 The syntactic aggregation model

The syntactic aggregation model has 21 branching nodes. Precision, recall and F-measure for the model are given in Table 4. As was to be expected on the basis of linguistic intuition, the value "middle" for the target feature did not play any role. In the test set there were only 2 observed instances of that value. The baseline for this task is 0.8566 (assuming "first" as the default value).

Table 4. Precision, recall, and F-measure for the syntactic aggregation model.

Value	Precision	Recall	F-measure
last	0.9191	0.9082	0.9136
first	0.9837	0.9867	0.9851
middle	0.0000	0.0000	0.0000
overall accuracy	0.9746		

8 Conclusion and future research

We have demonstrated on the basis of four examples that it is possible to learn the contexts for complex linguistic operations in sentence realization with high accuracy. We proposed to standardize most of the feature extraction for the machine learning tasks to all available linguistic features on the node, and its parent and grandparent node. This generalized set of features allows us to rapidly train on new sets of data and to experiment with new machine learning tasks. Furthermore, it prevents us from focusing on a small set of hand-selected features for a given phenomenon; hence, it allows us to learn new (and unexpected) generalizations from new data.

We have found decision trees to be useful for our classification problems, but other classifiers are certainly applicable. Decision trees provided an easily accessible inventory of the selected features and some indication of their relative importance in predicting the target features in question. Although our exposition has focused on the preferred value (the mode) predicted by the models, decision trees built by WinMine predict a probability distribution over all possible target values. For a system such as Amalgam, built as a pipeline of stages, this point is critical, since finding the best final hypothesis requires the consideration of multiple hypotheses and the concomitant combination of probabilities assigned by the various models in the pipeline to all possible target values. For example, our extraposition model presented above depends upon the value of the verb-position feature, which is predicted upstream in the pipeline. Currently, we greedily pursue the best hypothesis, which includes only the mode of the verb-position model's prediction. However, work in progress involves a search that constructs multiple hypotheses incorporating each of the predictions of the verb-position model and their scores, and likewise for all other models.

We have found the combination of knowledge-engineered linguistic operations with machine-learned contexts to be advantageous. The knowledge-engineered choice of linguistic operations, allows us to deal with complex linguistic phenomena. Machine learning, on the other hand, automates the discovery of general and domain-specific contexts. This facilitates

adaptation of the system to a new domain or even to a new language.

It should also be noted that none of the learned models can be easily replaced by a rule. While case assignment, for example, depends to a high degree on the lexical properties of the governing preposition or governing verb, other factors such as semantic relations, etc., play a significant role, so that any rule approaching the accuracy of the model would have to be quite complex.

We are currently adapting Amalgam to the task of French sentence realization, as a test of the linguistic generality of the system. Initial results are encouraging. It appears that much of the feature extraction and many of the linguistic operations are reusable.

Acknowledgements

Our thanks go to Max Chickering for assistance with the WinMine decision tree tools and to Zhu Zhang who made significant contributions to the development of the extraposition models.

References

S. Bangalore and O. Rambow 2000. Exploiting a probabilistic hierarchical model for generation. Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000). Saarbrücken, Germany. 42-48.

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

D. M. Chickering, D. Heckerman and C. Meek. 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.

S. Corston-Oliver, M. Gamon, E. Ringger, and R. Moore. 2002. *An overview of Amalgam: A machine-learned generation module*. To be presented at INLG 2002.

H. Dalianis and E. Hovy 1993 Aggregation in natural language generation. Proceedings of the 4th European Workshop on Natural Language Generation, Pisa, Italy.

P. Eisenberg 1999. Grundriss der deutschen Grammatik. Band2: Der Satz. Metzler, Stuttgart/Weimar.

U. Engel. 1996. Deutsche Grammatik. Groos, Heidelberg.

M. Gamon, E. Ringger, Z. Zhang, R. Moore and S. Corston-Oliver. 2002a. Extraposition: A case study in German sentence realization. To be presented at the 19th International Conference on Computational Linguistics (COLING) 2002.

M. Gamon, E. Ringger, S. Corston-Oliver. 2002b. Amalgam: A machine-learned generation module. Microsoft Research Technical Report, to appear.

G. E. Heidorn. 2002. 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.), Marce Dekker, New York.

I. Langkilde. and K. Knight. 1998a. The practical value of n-grams in generation. Proceedings of the 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, Canada. pp. 248-255.

I. Langkilde and K. Knight. 1998b. Generation that exploits corpus-based statistical knowledge. Proceedings of the 36th ACL and 17th COLING (COLING-ACL 1998). Montréal, Québec, Canada. 704-710.

J. D. McCawley. 1988 The Syntactic Phenomena of English. The University of Chicago Press, Chicago and London.

M. Reape. and C. Mellish. 1999. Just what is aggregation anyway? Proceedings of the 7th European Workshop on Natural Language Generation, Toulouse, France.

E. Ringger, R. Moore, M. Gamon, and S. Corston-Oliver. In preparation. *A Linguistically Informed Generative Language Model for Intra-Constituent Ordering during Sentence Realization.*

J. Shaw. 1998 Segregatory Coordination and Ellipsis in Text Generation. Proceedings of COLING-ACL, 1998, pp 1220-1226.

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

J. Wilkinson 1995 *Aggregation in Natural Language Generation: Another Look.* Co-op work term report, Department of Computer Science, University of Waterloo.