High quality machine translation using a machine-learned sentence realization component

Martine Smets, Michael Gamon, Jessie Pinkham, Tom Reutter and Martine Pettenaro
Microsoft Research
Redmond, U.S.A.
{martines, mgamon, jessiep, treutter, martinep}@microsoft.com

Abstract

We describe the implementation of two new language pairs (English-French and English-German) which use machine-learned sentence realization components instead of hand-written generation components. The resulting systems are evaluated by human evaluators, and in the technical domain, are equal to the quality of highly respected commercial systems. We comment on the difficulties that are encountered when using machine-learned sentence realization in the context of MT.

1 Introduction

Recently, statistical and machine-learning approaches have been applied to the sentence realization phase of natural language generation. The Nitrogen system, for example, uses a word bigram language model to score and rank a large set of alternate sentence realizations (Langkilde and Knight, 1998a, 1998b). Other recent approaches use syntactic representations. FERGUS (Bangalore and Rambow, 2000), HALogen (Langkilde 2000, Langkilde-Geary 2002) and Amalgam (Corston-Oliver et al., 2002) use syntactic trees as an intermediate representation to determine the optimal string output.

Amalgam, the sentence realization system introduced into our machine translation system maps a semantic representation to a surface syntax tree via intermediate syntactic representations. The mappings are performed with linguistic operations, the contexts for which are primarily machine-learned. The resulting syntax tree contains all the necessary information on its leaf nodes from which a surface string can be read. Our sentence realization system was first applied to German, although its architecture was designed to be language-independent. It was subsequently adapted to French, as described in (Smets et al., 2003), and is currently being ported to English.

The purpose of this paper is to present a full scale machine translation system using this sentence realization module and to report the quality of the resulting translations. We also review the results of testing only the sentence realization system (i.e., with native, not transferred input) and point out the issues that arise in the translation context. Work in progress to address the issues specific to translation is briefly outlined.

2 Overview of the Machine Translation system

The machine translation system described here operates from English as source language to German or French. It uses broad-coverage analyzers (the NLPWin system (Heidorn, 2000)) for those languages, large multi-purpose monolingual dictionaries, automatically generated bilingual dictionaries (English-French and English-German), the Amalgam sentence realization system, and a transfer component. The transfer component consists of transfer patterns automatically acquired from sentence-aligned bilingual corpora using an alignment grammar and algorithm described in detail in (Menezes & Richardson, 2001). Training takes place on aligned sentence pairs which have been analyzed by the source and target NLPWin analysis systems to yield sentence-level semantic dependency graphs (Logical Forms or LFs). The LFs have fixed lexical choices for content words and represent the predicate-argument structure of a sentence. LFs include semantic information concerning relations between nodes of the graph (Heidorn, 2000). The LF structures, when aligned, allow the extraction of lexical and structural translation correspondences which are stored for use at runtime in the transfer database. The transfer database can also be thought of as an example base of conceptual structure representations. The transfer database for English-French is trained on approximately 1.8 million aligned sentence pairs; the database for English-German is trained on 900,000 aligned sentence pairs. The training corpus is extracted from manually translated computer manuals and help files. This machine translation system architecture was previously implemented for French-English, German-English, Spanish-English, English-Spanish, English-Japanese, and Japanese-English. However, in each of these language pairs, the generation components for English, Spanish, or Japanese was created by hand. The novelty in the implementations of the systems for English-French and English-German is the use of the Amalgam generation component which uses machine learning extensively. In what follows, we will give an overview of the Amalgam sentence realization system for German and for French, then describe our evaluation methodology and results, and finally examine some issues in the systems. Our conclusion is that even without adaptation to the special conditions of MT, our machine-learned realization system allows us to produce output which is of comparable quality to current commercial MT systems.

3 Overview of the Amalgam sentence realization system

Amalgam takes as its input an LF graph. An example of a French logical form is given in Figure 2. Amalgam first degraphs the logical form into a tree and then augments it by the insertion of function words, assignment of various syntactic features, syntactic labels, etc., to produce an unordered syntax tree. Amalgam then establishes intra-constituent order. After syntactic aggregation, insertion of punctuation, morphological inflection, and capitalization, an output string is read off the leaf nodes. The contexts for most of these linguistic operations are machine-learned (Gamon et al. 2002a).

Amalgam is based on a division of labor between linguistically motivated, knowledge-engineered linguistic operations, and their machine-learned contexts. A linguist decides which stages and operations are necessary for sentence realization. The contexts or "triggering environments" in which these operations apply are machine-learned as classifiers on events extracted from LFs and corresponding syntax trees. Even the ordering component is viewed as a classification task: daughter constituents of any given parent node are ordered from left-to-right with a classifier determining which of the yet unordered daughter nodes is most likely to occur "next", given the set of already ordered nodes[1]. Training proceeds from a monolingual corpus which is parsed by the analysis system to LFs. Features are then extracted from LF nodes and their corresponding syntactic nodes. These extracted features comprise the data for machine learning of mapping operations from LF to a syntax tree. All machine-learned components employ decision trees for classification and for probability distribution estimation (Gamon et al., 2002b). The decision trees are built with the WinMine toolkit (Chickering, 2002).

In determining the set of necessary operations, we have taken care to be as language- and corpus-independent as possible. In this way, adapting the sentence realization to a different corpus requires only retraining on new data (an advantage of machine-learned systems), and the sentence realization system can be easily adapted to a different language (Smets et al., 2003).

Amalgam was first developed for German, then ported to French, and is currently being ported to English. The specifics for the German and French Amalgam modules are detailed in the next sections.

3.1 Overview of German Amalgam

Figure 1 lists the eight stages in German Amalgam: the label *ML* denotes that the context for the operation is machine-learned, *Proc* denotes the procedural (knowledge-engineered) character of the operation.

Stage 1 Pre-processing (Proc):
- 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 / verb position features.

Stage 3 Conversion to syntax tree (Proc):
- Introduction of syntactic representation for coordination.
- Splitting of separable prefix verbs based on both lexical information and previously assigned verb position features.

Stage 4 Movement:
- Raising, wh movement (Proc).

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

Stage 6 Extraposition (ML)

Stage7 Surface clean-up (ML):
- Lexical choice of det. and relative pronouns.
- Syntactic aggregation.

Stage 8 Punctuation (ML)

Stage 9 Inflectional generation (Proc)

Figure 1 : The stages of German Amalgam

[1] This sorting approach to ordering has outperformed language-model-based ordering for French, English and German. We report on the details of these results in work in progress (Ringger et al. 2003).

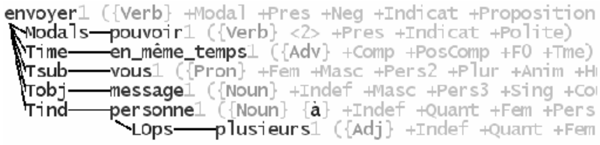
There are a total of twenty-one decision trees in the German system. The complexity of the decision trees varies with the complexity of the modeled task: the number of branching nodes in the decision tree models in the German system ranges from just 4 to 7,876 in the order model.

3.2 French Amalgam

French Amalgam re-uses the architecture of the German system. Indeed, sentence realization on the basis of a semantic graph must undergo many of the same transformations regardless of the language: pre-processing of the logical form, fleshing-out, conversion to syntax tree, etc. We outline below the stages of the French system, and compare them to the German system.

Stage 1, the pre-processing of the data, involves language-neutral transformations from a graph representation to a tree representation, and can be reused without alteration by the French system.

The fleshing out of the logical form in Stage 2 required changes for French. French does not need a machine-learned model for case. On the other hand French requires a model for clitic insertion which does not exist in German.



```
envoyer1 ({Verb} +Modal +Pres +Neg +Indicat +Proposition
 Modals—pouvoir1 ({Verb} <2> +Pres +Indicat +Polite)
 Time———en_même_temps1 ({Adv} +Comp +PosComp +F0 +Tme)
 Tsub———vous1 ({Pron} +Fem +Masc +Pers2 +Plur +Anim +H
 Tobj———message1 ({Noun} +Indef +Masc +Pers3 +Sing +Co
 Tind———personne1 ({Noun} {à} +Indef +Quant +Fem +Pers
             LOps———plusieurs1 ({Adj} +Indef +Quant +Fem
```

Figure 2 : French logical form

Because French does not have separable prefix verbs, the lexical operation that splits prefixes in German is not needed in Stage 3. French uses a head-switching operation for verb phrases headed by modal verbs, because of the status of French modals: although they share the semantic characteristics of modal verbs, they behave syntactically as main verbs (see Smets et al., 2003). Figure 2 is the logical form of the sentence in (1) which illustrates a modal construction. The modal (*pouvoir*, 'must') is the syntactic head, but an attribute of its complement (*envoyer*, 'send') in the logical form.

(1) Vous ne pouvez pas envoyer un message à plusieurs personnes en même temps.
'You cannot send a message to several people at the same time'.

Stage 4 (raising and *Wh*-movement) is identical for both languages.

In stage 5, both German and French use a left-to-right model of constituent order. For each language, the model is a decision tree representing the probability distributions involved in ordering.

Extraposition of relative and complement clauses, which is common in German (Gamon et al., 2002c), is rare in the French technical software manuals: there were too few examples of extraposition in the French data to train an extraposition model for Stage 6.

Stage 7 (clean-up) uses language-specific information, especially in the realization of lexical forms of function words.

Finally, stage 8, the realization of inflection, is completely language-specific.

Figure 3 provides a summary of the French sentence realization system.

There are twenty-one decision trees in the French system, and as with German the complexity of the decision trees varies with the complexity of the task modeled. The number of branching nodes in the decision tree models in the French system ranges from 10 (for the *subconj* model, which decides whether to insert the subordinate conjunctions *que* ('that'), *si* ('whether') or nothing) to 1,040 (the label model), except for the order model for verb phrases which has 5,456 branching nodes.

Stage 1 Pre-processing (Proc):
- Degraphing of the logical form.
- Retrieval of lexical information.

Stage 2 Flesh-Out (ML):
- Assignment of syntactic labels.
- Insertion of function words.
- Insertion of clitics.
- Assignment of case (Proc).

Stage 3 Conversion to syntax tree (Proc):
- Introduction of syntactic representation for coordination.
- Head-switching (ML).

Stage 4 Movement:
- Raising, wh movement (Procedural).

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

Stage 6 Surface clean-up (ML):
- Lexical choice of determiners and relative pronouns.
- Syntactic aggregation.

Stage 7 Punctuation (ML)
Stage 8 Inflectional generation (Proc)

Figure 3: The stages of French Amalgam

There are a number of differences between the French and German systems, some concerning models that are language-specific, others relating to features relevant only for one language. Most of the differences are in feature extraction and in the linguistic operations relying on the information provided by the models. See (Smets et al., 2003) for a discussion of these differences.

4 Data and feature extraction

The data for all models are automatically extracted from a set of 100,000 sentences drawn from software manuals. Depending on the linguistic phenomenon to be modeled, between 30,000 and one million cases are extracted from these sentences. The sentences are analyzed in the NLPWin system (Heidorn, 2000), which provides a syntactic and logical form analysis. Nodes in the logical form representation are linked to the corresponding syntax nodes, allowing us to learn contexts for the mapping from the semantic representation to the surface syntax representation. The data is split 70/30 for training versus model parameter tuning. For each set of data we build decision trees at several levels of granularity and select the model with the maximal accuracy as determined on the parameter tuning set.

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 for each model. This allows us to share the majority of code among the individual feature extraction tasks and among languages. Typically, we extract the full set of available linguistic 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 yields approximately six hundred features that provide a sufficiently large structural context for the operations. There are three types of features: lexical, syntactic and morpho-syntactic. The decision tree learner selects appropriate features for a particular task. For example, for the insertion of infinitive markers in French, the selected features fall in the following categories (in decreasing order of importance, according to the learner):

- Grammatical function of the infinitive clause, e.g., whether it is a purpose clause or an object.
- Is there already a governing preposition?
- Subcategorization features of the parent
- Category of the parent
- Semantic features of the node, parent and grandparent
- Subcategorization features of the node itself
- Other arguments of the parent
- Is there a preposition introducing the parent?
- Function of the parent
- Nominal features of the parent
- Arguments and agreement features of the grandparent

The top features selected by the decision tree learner correspond to linguistic intuition: the choice of an infinitive marker depends on the function of the clause, on whether the infinitive clause is already introduced by a preposition, and on subcategorization features of the parent. Other features, however, are less intuitive (for example, agreement features of the grandparent), but are judged less significant by the learner.

In addition to these standard features, for some of the models we add a small set of specially computed linguistic features that we believe to be important for the task at hand. For example, the model which inserts negation in French must choose between inserting "*ne pas*" or "*ne*". The first value is the default as in (2), while the second value is chosen if a negative quantifier is present among the arguments of the verb, as in (3).

(2) Assurez-vous que les périphériques **ne** bougent **pas** ou **ne** vibrent **pas**
"Make sure the devices are not moving or vibrating"

(3) Dans ce cas, **aucune** modification **n'**est observée
"In this case, no change is seen"

In order to learn the correct context for each form of the negation, the decision tree has to take into account the presence of negative quantifiers in the clause. Because this information is not readily available on the node under consideration (the verb), its parents or its grandparent, we define specific functions to compute that feature. Similarly, we compute special features which assess "heaviness" in terms of character and token length for the models of clausal extraposition in German, and for punctuation insertion in the Amalgam modules for all languages.

5 Evaluation of Amalgam as a stand-alone module (German to German and French to French)

The evaluations performed for Amalgam as a stand-alone component and the evaluations performed for machine translation follow the same model. Each evaluator sees the output that the system produces, and also sees the reference that is considered perfect. In the case of a French-to-French evaluation, the reference is the original sentence. For the evaluation of translation, the reference is the reference translation. We use approximately five raters for each evaluation, and all data is blind and distinct from training data. Raters evaluate around 500 sentences taken randomly from our test corpus (a subset of Microsoft technical manuals).

For the evaluation of Amalgam, 545 test sentences in isolation from a blind technical software corpus were analyzed with our analysis system, giving a logical form representation. Our sentence realization system then generated the 545 sentences on the basis of that representation. We did not control for noise introduced into the data by the analysis phase (in about 15% of the sentences). Nevertheless, this experiment gives us a good indication of the performance of our system.

All the raters assigned an integer score between 1 and 4, comparing each sentence to the reference using the scoring system in Table 1.

1	"Unacceptable". Absolutely not comprehensible and/or little or no information transferred accurately
2	Possibly Acceptable: Possibly comprehensible (given enough context and/or time to work it out); some information transferred accurately
3	"Acceptable": Not perfect (stylistically or grammatically odd), but definitely comprehensible, AND with accurate transfer of all important information
4	**"**Ideal**"**: Not necessarily a perfect translation, but grammatically correct, and with all information accurately transferred

Table 1 : the rating system for evaluation

The score of a sentence is the average of the scores given by the five raters. The resulting score is the average of the scores of all of the sentences by all of the raters. The results for both the German and French sentence realization systems appear in Table 2. Note the very high quality of both systems. German, which was created first, rates higher in absolute quality than French.

Pair	Date	Score
FF	11/02	2.92 +/- 0.19
GG	1/03	3.25 +/- 0.16

Table 2 : results of French-to-French and German-to-German evaluation

Poor scores can be due to the analysis system (for the evaluation, a sentence is first analyzed into a logical form, from which Amalgam realizes the sentence again). If we start from a faulty logical form, the realized sentence is most likely of poor quality. This occurs mainly with long sentences. Problems of the sentence realization systems often involve word order (if the word order is incorrect, a sentence can be unintelligible, even with all the correct content words), absence of syntactic aggregation (some constituents are redundant). Examples of problematic sentences for the French system are given below. The first sentence suffers mainly from word order and syntactic aggregation issues, and received a score of 2. The second sentence received a score of 1, and has a problem of word order and negation insertion (a negation is not inserted when it should be inserted).

> Ces propriétés ainsi que d'autres peuvent être définies et modifiées directement à partir du diagramme de base de données[2].
>
> Peut ces propriétés ainsi qu' autre sont directement modifiée à partir de le diagramme de base de données et peut ces propriétés ainsi qu' autre sont directement définie à partir de le diagramme de base de données.
>
> Si un serveur n'est pas trouvé, le programme d'installation affiche un message indiquant que ce serveur n'est pas joignable[3].
>
> S' un serveur est ne trouvé le programme d' installation affiche un message indiquant que ce serveur est joignable.

6 Evaluation and Results for English-French and English-German

We performed evaluations of both translation systems in June of 2003. For the evaluation of translation, 500 sentences were analyzed by the English analysis system to give logical forms, which were transferred into logical forms of the target language. Amalgam realized sentences of the target language on the basis of these transferred logical forms.

For each language pair, the evaluators evaluated our system concurrently with a highly respected commercial system (the competition), which serves as a baseline for performance. In both cases, we found that our system, which in the case of French was first assembled in January 2003, equals the level of absolute quality of the competition.

The competition for English-French is the latest Systran system for EF; the competition for English-German is the latest Systran system for EG. These competitor systems are run with the

[2] 'These properties and others can be defined and directly modified from the database diagram'.

[3] 'If no server is found, the installation program displays a message stating that the server cannot be found'.

available domain dictionaries (computer dictionaries). Results appear in Table 3 below.

Pair	Date	Our system	Competition
EF	6/7/03	2.43 +/- 0.17	2.26 +/- 0.16
EG	6/16/03	2.43 +/- 0.22	2.14 +/- 0.28

Table 3 : evaluation results against competitor system

BLEU scores (Papineni et al. 2001) for the E-G and E-F systems are given in Table 4. The scores were computed on a set of 20K translations of held-out source language data, about one month after the human evaluations were conducted.

Pair	BLEU scores
EF	0.382
EG	0.323

Table 4 : BLEU scores

7 Discussion of issues for sentence realization in the context of MT

The results in the previous section show significant differences in the scores of monolingual sentence realization (French to French or German to German), and translation from English into French or German. The reason for this gap is the noise introduced by the transfer component: transferred LFs are not native LFs. This section presents some problems encountered in English-French translation.

7.1 Issues in French transferred LFs

A main issue in EF translation is the discrepancy between the use of determiners in English and in French: determiners in English are omitted in a number of cases (indefinite plurals, indefinite mass, etc), while in French determiners are always used (except in some types of collocations). A mismatch between definiteness/indefiniteness features in the source and target language is detrimental to the performance of our realization system. An example of a translation problem relating to that issue is given in Table 5.

This will prevent rich-text information from being sent along with the message.
Cela empêchera information au format RTF d' être envoyée avec le message.

Table 5 : translation problem resulting from different use of determiners in English and French

In the French sentence above, *information* should be preceded by a determiner. However, the LF transferred from the English LF does not contain a definiteness feature. Because our sentence realization system is trained on native French LFs, it expects that information to be present when the decision is made whether to insert determiners.

This problem never arises in French-to-French sentence realization, as the relevant information is present in the native French LF.

More severe problems arise: the type of complements required by an English verb is not always the same as the complement required by its French translation. For example, in our corpus, a translation is learned between *NP may receive a message* and *il se peut que NP recevoir un message* ('it is possible that NP receive a message'). The information transferred on *recevoir* is the information present on *receive* in the English LF. But *receive* is a base verbal form, while *il se peut* requires a subjunctive complement clause. The verb *recevoir*, inheriting the features of *receive*, is realized as an infinitive. As a result, there is no subordinate conjunction, nor expressed subject of the infinitive. Amalgam uses verbal features such as tense and mood (among other features) to decide whether or not to insert subordinate conjunctions, and also whether to express the subjects of verbs. French infinitives do not head *that*-clauses, and do not have their own subject: this explains the translation in Table 6.

These examples illustrate the problems we encountered when using an automatic sentence realization system. The set of transferred features is not always what is expected by the sentence realization component. Some features are missing, others are not appropriate in the target language for the constituent they are supposed to characterize. This problem of "noisy" transferred logical forms is very common, and explains the differences of scores between translation and native generation.

When you try to browse this type of folder, you may receive an error message similar to the following:
Quand vous essayez de parcourir ce type de dossier il se peut recevoir un message d'erreur similaire au suivant:

Table 6 : translation problem resulting from differences in complement structure in English and French

The obvious long-term approach to these problems is to incrementally improve the transfer mechanism to a point where transferred LFs are native-like. It is important to point out, however, that issues of over- and underspecification will always be a major challenge and that sentence realization may also be employed in situations where the data are inherently noisy (speech). For the latter reasons we have decided to experiment with a machine-learning approach to noisy features in linguistic representations. The next section briefly discusses this approach in the context of the Amalgam sentence realization system.

7.2 Noisy data and pre-generation: future work

An interesting difference exists between a machine-learned sentence realization system like Amalgam and a hand-coded generation component: Amalgam mercilessly picks up on distributional regularities of linguistic features, but of course lacks the linguistic understanding of these features that a human generation grammarian possesses and utilizes for maximal grammatical accuracy. The result is that the machine-learned models may pick up on features and feature combinations that seem counterintuitive to a grammarian, although they are solid indicators on native semantic representations in the set of training data provided. For instance, instead of a grammatical generalization of the form "if the part of speech of X is noun", a model may learn that "if X has the 3rd person feature". The latter is true for the majority of nouns and pronouns, of course, and may capture a very similar generalization, but does not correspond to the judgment of a grammarian. In our experience, this can make Amalgam more susceptible to noisy transfer. There is no way we can convey to a machine-learning component that certain features and feature combinations are to be more trusted than others and are more "solid" indicators - the machine-learned component can (and should) only rely on the information present in the training data.

The result is that correct feature transfer becomes a larger issue in the transfer component, and that it also may prove useful to have a post-transfer and pre-generation component which specifically deals with feature noise, underspecification, and overspecification of linguistic features. We are currently experimenting with a machine-learned approach to this "clean-up" component (see Corston-Oliver and Gamon 2003). The basic idea is to learn a set of rules which set or delete linguistic features based on function word lemmas, the presence of other features in the structural neighborhood, parts-of-speech and semantic relations. Initial results indicate that this approach results in a statistically significant improvement in translation quality.

8 Conclusion

It is well known that the development of each component in the creation of a quality MT system can be labor-intensive and time-consuming. In order to reduce the time and effort, we have incorporated the Amalgam sentence realization system, trained only on the target language, in our overall machine translation architecture. Porting Amalgam to French has proven to be feasible in a matter of 10 person weeks. Even at this early stage in our experimentations, we found the preliminary results of human evaluations of the translation quality to be comparable to existing commercial systems, and this is very promising.

9 Acknowledgments

We thank Simon Corston-Oliver and Eric Ringger for their collaboration on the Amalgam project, David Rojas for his contributions to French and German Amalgam and the Butler Hill Group for conducting the human evaluations.

10 Bibliographical References

Bangalore S. and Rambow O. (2000) "Exploiting a probabilistic hierarchical model for generation". In *Proceedings of COLING 2000*, Saarbrücken, Germany, pp. 42-48.

Chickering D. M. (2002) *The WinMine Toolkit.* Microsoft Technical Report MSR-TR-2002-103.

Corston-Oliver S., Gamon M., Ringger E. and Moore R. (2002) "An overview of Amalgam: a machine-learned generation module". In *Proceedings of INLG 2002*, New York, pp.33-40.

Corston-Oliver S. and Gamon M. (2003) "Combining decision trees and transformation-based learning to correct transferred linguistic representations". To appear in *Proceedings of MT Summit IX.*

Gamon M., Ringger E., Corston-Oliver S., Moore R. (2002a) "Machine-learned contexts for linguistic operations in German sentence realization". In *Proceedings of ACL 2002*, pp. 25-32.

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

Gamon M., Ringger E., Zhang Z., Moore R. and S. Corston-Oliver (2002c) “Extraposition: A case study in German sentence realization”. In: *Proceedings of COLING 2002*, pp. 301-307.

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

Langkilde I. (2000) “Forest-Based Statistical Sentence generation”. In *Proceedings of NAACL 2000*, pp. 170-177.

Langkilde-Geary I. (2002) “An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator”. In *Proceedings of INLG 2002*, New York, pp.17-24.

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

Langkilde I. and Knight K. (1998b) “Generation that exploits corpus-based statistical knowledge”. In *Proceedings of the 36th ACL and 17th COLING*. Montreal, Canada pp. 704-710.

Menezes, A. and Richardson S. 2001. A Best-First Alignment Algorithm for Automatic Extraction of Transfer Mappings from Bilingual Corpora. In Proceedings of the Data-Driven MT workshop, ACL 2001.

Papineni K., Roukos S., Ward T. and Zhu W.-J. (2001). “Bleu: a Method for Automatic Evaluation of Machine Translation”. IBM Research Report RC 22176.

Ringger E., Gamon M., Smets M., Corston-Oliver S. and Moore R. (in preparation) “Linguistically informed statistical models of constituent structure for ordering in sentence realization”.

Smets M., Gamon M., Corston-Oliver S. and Ringger E. (2003) “The adaptation of a machine-learned sentence realization system to French”, *Proceedings of the 10th conference of the European Chapter of the Association for Computational Linguistics*, 2003. Budapest, Hungary pp 323-330.