Combining Machine Learning and rule-based approaches in Spanish and Japanese sentence realization

Maite Melero
Microsoft Research
One Microsoft Way
Redmond, WA 98008, USA
maitem@microsoft.com

Takako Aikawa
Microsoft Research
One Microsoft Way
Redmond, WA 98008, USA
takakoa@microsoft.com

Lee Schwartz
Microsoft Research
One Microsoft Way
Redmond, WA 98008, USA
leesc@microsoft.com

Abstract

In this paper we describe two parallel experiments on the integration of machine learning (ML) methods into the Spanish and Japanese rule-based sentence realization modules developed at Microsoft Research. The paper explores the use of decision trees (DT) for the lexical selection of the copula in Spanish and the insertion of a locative postposition in Japanese. We show that it is possible to machine-learn the contexts for these two non-trivial linguistic phenomena with high accuracy.

1 Introduction

The two experiments described in this paper were carried out in the framework of the Spanish and Japanese sentence generation modules that are part of MSR-MT, the multilingual Machine Translation system developed at Microsoft Research. MSR-MT is a hybrid system that uses hand-written, rule-based linguistic components for analysis and generation, and example-based, statistical components for transfer (Richardson et al., 2001). The output of the analysis, as well as the input to generation is an annotated predicate-argument structure or logical form (LF) (Heidorn, 2000). Transfer takes place between source LF and target LF using an automatically generated knowledge base known as MindNet, built by aligning logical forms of bilingual text. As described in (Aikawa et al., 2001), the rule-based generation module generates the surface string in the target language from the transferred LF.

Here we explore the integration of a machine learning technique into two generation components in order to deal with two different sentence realization problems: the selection of the copula in Spanish and the insertion of a locative postposition in Japanese. As shown by (Gamon et al. 2002) among others, many linguistic operations can be viewed as classification tasks, thus lending themselves to statistical methods such as decision tree classifiers.

Following the questions raised by (Bangalore et al., 2001) on the impact that the type of corpus has on the quality of the stochastic generation components, we wanted to perform our experiments using two very different types of texts. For this purpose we built two different models for each experiment: one using text coming from the Encarta encyclopedia and another using text from technical and computer manuals.

Our goals can be summarized as follows:

- To integrate a ML approach for a well-defined linguistic operation into an otherwise totally hand-coded rule-based generation module;
- To evaluate the usefulness of such an approach vs. hand-coded rules;
- To evaluate the impact of the type of the training data on the accuracy of the model.

To build the statistical models, we used the WinMine toolkit (Chickering et al., 1997) which has been used to build a machine-learned generation module (Corston-Oliver et al., 2002). As training data, we used logical forms produced by analyzing text in the languages of interest, Spanish and Japanese, respectively. The data was automatically split 70/30 for training and parameter tuning by the WinMine toolkit, which then built different decision trees with different degrees of granularity, by manipulating the prior probability of tree structures to favor simpler ones. The best model was chosen and then evaluated using a different blind set of sentences. We also performed an evaluation across text types.

2 Selection of the Spanish copula

2.1 Description of the problem

Spanish has two different copulas, *ser* and *estar*, which are both translated into English as ‘to be.’ *Ser* is used to express permanence, identity or inherent quality and *estar* is used for temporary conditions and location. The correct generation of the copula is a specific instance of the general problem of lexical selection. In the context of our MT system, this problem is generally solved by transfer, which can make decisions that are context sensitive. However, as pointed out in (Aikawa et al., 2000), the generation component, being ultimately responsible for the fluency and grammaticality of the output, re-evaluates some of the decisions made by transfer.

The main uses of *ser* and *estar*, following (Porroche, 1988), are summarized in Table 1, leaving out the auxiliary uses.

	Predicative function	Attributive function	Identity function	Stative passive
SER	**Existential (case 1)** *La reunión es a las 6* (The meeting is at 6) *La fiesta es en mi casa* (The party is at my place)	**With nouns (case 3)** *Juan es (un) médico* (Juan is a doctor) **With adjectival phrases (case 4)** *Juan es guapo* (Juan is handsome) *El globo es de colores* (The balloon is multicolored)	**(case 6)** *Juan es el médico* (Juan is the doctor)	
ESTAR	**Locative (case 2)** *Él está en casa.* (He is at home) *El libro está sobre la mesa.* (The book is on the table)	**With adjectival phrases (case 5)** *María está muy guapa* (Mary looks very pretty) *Mi jefe está de vacaciones* (My boss is on vacation)		**(case 7)** *La casa está construida* (The house is built)

Table 1: Uses of *ser* and *estar*

Each of these cases presents a different degree of difficulty from a generation perspective. Thus, cases 3, 6 and 7 can be easily addressed using basic morphosyntactic information: only *ser* can take an NP argument, and only *estar* can appear as a main verb with a past participle. The distinction between cases 1 and 2 is more challenging and involves properties of the subject as well as of the predicate. Cases 4 and 5 are the hardest to predict and entail aspectual interpretations sometimes difficult to deduce from context.

The use of *ser* in the <copula+AJP> constructions implies that the attribute is an inherent quality of the subject, while the use of *estar* implies that the condition expressed by the attribute is accidental or circumstantial. Some attributes can be used with both verbs, provided that the nature of these attributes allows for the two aspectual interpretations. Thus, *La nieve es fría* and *La nieve está fría* (both translate as ‘The snow is cold’) are both possible. Other attributes do not have this flexibility. For example, *disponible* ‘available’ can only go with *estar* and *eterno* ‘eternal’ can only go with *ser*. Many of the attributes that have a strong preference for *ser* could also go with *estar* in very specific contexts, and then only if the subject is able to experience change, as *rojo* ‘red’ in *El semáforo está rojo* ‘The traffic light is red’.

The problem of selecting the right copula is complex because it has to take many different types of information into account. Nonetheless, it can be easily mapped into a classification problem. For these reasons, it is a good candidate for machine learning techniques.

2.2 Experiment design and evaluation

2.2.1 Decision Tree model for selecting the copula

We built two different DT models: Model A, using 131K sentences from the Encarta encyclopedia; and Model B, using 55K sentences from technical and computer manuals. All the sentences used for the two models contained at least one instance of *ser* or *estar*. The target feature, i.e. feature we wanted to predict, was expressed in terms of the copula being *estar* (the less frequent value) or not. This translates into the Boolean values "no" for *ser* and "yes" for *estar*[1].

We parsed the sentences up to their logical form and then automatically extracted 290 *variables* from each sentence (or clause containing *ser* or *estar*). A variable is the combination of a position or node in the LF structure we want the DT to consider, and a linguistic attribute or feature that may be present in this node. Thus, for instance: *Anim(Tsub)* means "presence of the feature Anim(ate) in the (logical) subject"; *Time(Tobj)* means "presence of a Time attribute in the (logical) object". Most of these variables were binary, with 1 representing presence of the corresponding feature and 0 representing absence. In a few cases, we used the actual value of the attribute: namely, syntactic category of the logical object and lemma[2] of the preposition in the prepositional complement.

Although we manually selected the positions in the LF to be considered by the DT, we did not perform any manual selection of features but rather let WinMine choose the best predictors among them. The number of *predictors*, or variables that are predictive of the target feature, selected by the decision tree algorithm was the same in both models: 55 out of the original 290. The set of predictive variables in both models was very similar, and as expected, the strongest predictors were the same in both cases. They can be grouped in the following way:

- semantic relation of the argument of the copula (object, prepositional complement, locative)
- morphological properties of the argument (past participle)
- lexical semantic features of the subject (animate, proper name, count/mass)
- lexical semantic features of the argument (color, animate, count/mass, location)
- presence of a modifier (manner, time, means)
- lemma of the preposition of the prepositional complement

Most of the predictors have intuitive linguistic relevance to the problem, but some of them were not expected, as:

- presence of an intensifier, classifier or operator on the argument (only in Model A)
- coordination in the main node or in an argument

The overall accuracy of each model, as well as the values for precision and recall, are measured on the 30% part of the training data that is held out for the purpose of selecting the best tree. Those values, as well as the size of the two models, measured by the number of their branching nodes, are summarized in Tables 2 and 3.

Model	A (Encarta)	B (technical)
#Branching nodes	109	117
Baseline[3]	82.85%	68.21%[4]
Accuracy	95.10%	90.36%

Table 2: Size, baseline and overall accuracy for the two models

Case	Precision (%)		Recall (%)		F-measure[5] (%)	
Model	A	B	A	B	A	B
Ser	95.78	89.82	98.43	96.84	97.09	93.20
Estar	91.23	91.86	79.07	76.45	84.71	83.45

Table 3: Precision and recall for *ser*/*estar*

[1] There is some noise in the corpus, as sentences where *ser* or *estar* are auxiliaries have not been excluded. However, the frequency of this use is proportionally low.

[2] A lemma of a word is its lexeme or citation form, e.g. the infinitive form of a verb.

[3] The baseline represents the overall accuracy if the most frequent value (i.e. *ser*) would have been selected in all cases.

[4] There is a big difference in the value of the baseline (82.85% for Encarta and 68.21% for technical manuals), indicating that the preference of *ser* over *estar* is much more pronounced in Encarta than in the manuals

[5] F-measure is the harmonic mean of precision and recall.

Once we built the decision trees, we wanted to see whether their accuracy varied across different domains. Finally, we were interested in evaluating their performance against a hand-coded rule.

2.2.2 Evaluation of the models

Even though our main interest is to use the result of this experiment in an application environment such as MT, we used Spanish texts for evaluation purposes. It may seem that evaluating the results using Spanish data constitutes an artificial environment: after all, we are generating Spanish sentences from structures resulting from the analysis of the same Spanish sentences. Nonetheless, this enables us to perform an automatic evaluation of the results. The procedure is the following: we analyze and regenerate the Spanish sentences (with the right copula in them) and we create a master file with the results; we then run regression testing against this file by removing the copula and recalculating it using the model. The number of changes equals the number of regressions[6] .

We used two blind testing sets of 10K sentences each, one for each type of text (Encarta and technical manuals). Since we were interested in evaluating the usefulness of the ML approach with respect to encoding the information in the form of a rule, we also measured the accuracy of a not-too-complex-but-not-too-dumb hand-coded rule that uses some of the linguistic insights revealed by the inspection of the models. Table 4, which gives the number of errors in the generation of the copula and the accuracy as a percentage, summarizes the results of our evaluation on a blind corpus.

	Model A (Encarta)	Model B (Technical)	Hand-coded rule
Encarta text	447/10k (95.53%)	753/10k (92.47%)	959/10k (90.41%)
Technical manuals	1022/10k (89.78%)	966/10k (90.34%)	1383/10k (86.17%)

Table 4: Accuracy of the two models vs the hand-coded rule

[6] If there is more than one copula in a sentence and there is more than one regression in this sentence, we will only be able to count one regression. However, we consider that the evaluation set is large enough to account for that noise.

From these results we observe that there is an expected correlation between the type of text and the type of model: Model A is the best model for the Encarta text and Model B is the best model for the technical text. Interestingly, the model trained on technical data increases its accuracy when tested on text from Encarta. This is consistent with the fact that all three methods have better results on text from Encarta. The reasons are not clear but one possible explanation is that, as seen with the values for the baseline above (82% vs. 68%), in this type of text the copula insertion is “easier” to predict. The hand-coded rule does a poorer job overall. Error analysis shows that the rule is slightly more biased towards *estar* than the model. The formulation of the contextual constraints is necessarily simpler in the rule than in the models (which each have over a hundred branching conditions). Both the models and the rule perform poorly on <copula+AJP> constructions (cases 4 and 5 above) defaulting to *ser* most of the time.

2.2.3 Enriched models using the lemma of the attribute

In order to provide a solution for the <copula+AJP> cases, we built a version of the models that looks at the lemma of the argument. The DT is able to cluster lemmas on a statistical basis, obviating the need to encode this sort of selectional information in the dictionary. The expected improvement is hardly noticeable in Model A. However, in the case of Model B, the enriched version (Model B’) is much smaller (72 vs. 117 branching nodes) and its overall accuracy jumps to 97% (notable, especially if we consider that the baseline for this type of text is 68%).

Model	A	A’	B	B’
Text type	Encarta	Encarta	technical	technical
Lemma	no	yes	no	yes
#Predictors	55	41	55	35
#Branching	109	101	117	72
Overall accuracy (%)	95.10	95.40	90.36	97.04

Table 5: Comparison of models according to text type and use of lemma of the adjective

We wanted to evaluate how well Model B' would do in the blind set used in our previous evaluation. The result, shown in Table 6, with blind data from the technical domain is predictably good, but a more surprising result is the 95.10% accuracy on the sentences from Encarta.

	Model B'(Technical trained model using lemmas)
Encarta text (10K sentences)	490 (95.10%)
Technical manuals (10K sentences)	306 (96.94%)

Table 6: Evaluation of Model B' on the blind set.

2.2.4 Integration of the DT Model in an MT system

The generation rule that predicts the lemma of the copula calls the DT model by invoking a function that returns a Boolean value. This function takes as parameters the DT model, the target feature we are trying to predict (*estar* in our case), and the LF node we are considering (in our case the node of the copula).

The Spanish generation grammar in the context of which this experiment has been performed is currently being used to generate the Spanish output of an MT system that has English as input. In this MT system, all lexical selections are, in principle, performed by transfer. Transfer rules are automatically extracted from parsed aligned corpora (Menezes & Richardson, 2001). Thus, the lemma of the copula is also computed by transfer rules, with a varying degree of accuracy. We wanted to perform a second evaluation of our best DT model, this time in an MT environment. We picked the model that had been trained on technical text and used information about the lemma of the adjective (i.e. Model B'). We had two goals in mind:

- prove that a model trained on a monolingual Spanish corpus could be used on structures coming from transfer;
- compare the degree of accuracy of the model vs the transfer component in the task of copula selection.

We took about 9K English sentences from computer manuals and processed them with our English-Spanish MT system, keeping the copula that transfer had found. We then kept these results in a master file. We included a rule in the generation grammar that removed the lemma of the copula and recalculated it using the DT model, and then ran regressions on the previous master file. We obtained 154 differences. Those were the cases for which transfer and DT predicted a different copula. Since we were only looking at the differences we were in fact ignoring the cases where transfer and DT were both right or both wrong. We reviewed all the differences manually and obtained the results shown in Table 7.

	#differences
DT was better	116/154 (73.00%)
Transfer was better	22/154 (14.20%)
Neither[7]	16/154 (10.30%)

Table 7: Comparison of transfer vs. DT results on the task of copula selection

The DT model beat the copula selection performed by transfer in 116 cases, versus 22 where transfer was right and the model was not.

3 Selection of a locative postposition in Japanese

3.1 Description of the problem

The Japanese experiment deals with the use of the two postpositions for location nouns: (i) *de* ('at') and (ii) *ni* ('in'). The choice between the two depends on the type of eventuality that a sentence denotes: if the sentence denotes an event, *de* is used for the location noun; if the sentence is stative, *ni* is used. The following examples illustrate this difference.

(1) a. ジョンは、ここに　住んでいる。
John -wa　koko-ni　sunde-iru
John -Top　this place-in　live-ing
"John lives here."　(stative -> "ni")

[7] Those were cases were the output was too ill-formed to consider correctness of the copula.

(LF)

```
住む ({live} +Pres +Prog)
 \Locn—ここ ({this place} {に(in)} +Place)
  \Tsub—ジョン({John}+PrprN +Humn)
```

b. ジョンは、 ここで 食べる。
John -wa koko-de taberu.
John -Top this place-at eat
"John eats here." (event -> "de")

(LF)

```
食べる ({eat} +Pres)
 \Locn—ここ({this place}{で(at)} +Place)
  \Tsub—ジョン({John}+PrprN +Humn)
```

The predicate in (1a), *sunde-iru* ('to live'), is stative and hence, the location noun *koko* ('this place') is marked by *ni*. On the other hand, the predicate in (1b), *taberu* ('to eat'), denotes the event of John's eating and the location noun is therefore marked by *de*. In the LFs above, the location nouns are indicated as Locn and the postpositions are provided. Thus, in generating surface strings using native Japanese LFs, the sentence realization component has no problem; i.e., *de*/*ni* is given and hence, no decision is necessary. However, when the Japanese generation module takes as input a transferred LF (as in the MT scenario), the correct postposition is not always provided. For instance, the following is the transferred LF from the English sentence, 'John eats lunch in that room'. Here, the transferred LF provides the wrong postposition *ni* to indicate the place of John's action of eating lunch.

(2) (Transferred LF of 'John eats lunch in the room.')

```
食べる ({eat} +Pres)
 \Locn—部屋 ({room}{に(in)} +Def)
  \Tsub—ジョン ({John} +Pers3 +PrprN)
   \Tobj—昼食 ({lunch})
```

Such mistakes are common in transferred LFs. The Japanese generation component thus needs to have an independent mechanism to handle this phenomenon. However, predicting which postposition (*de* or *ni*) occurs in which contexts is a difficult task. As mentioned above, the choice between *de* and *ni* depends on the type of eventuality that a sentence denotes. Thus, predicting the correct choice between these two postpositions requires fine-grained lexical-semantic coding on all the verbs in Japanese. Furthermore, the choice sometimes is contingent upon other factors. For instance, both (3a) and (3b) below have the same predicate (i.e., *aru* 'to exist'). However, the location noun *Tookyoo* 'Tokyo' in (3a) is marked by *de* whereas in (3b), it is marked by *ni*.

(3) a. 東京で ロボットの 展示会が ある。
Tookyoo-de robotto-no tenjikai-ga aru
Tokyo-in robot-Gen exhibit-Nom exist
"There is a robot exhibit in Tokyo."

b. 東京に フランス料理店がたくさん ある。
Tookyoo-ni furansu-ryooriten-ga takusan aru
Tokyo-in French restaurants many exist
"There are many French restaurants in Tokyo."

The contrast above can be reduced to the difference in types between the two subjects: the subject in (3a) (i.e., *robotto-no tenjikai* 'the robot exhibit') is an event nominal and hence, *Tookyoo* is marked by *de* whereas the subject in (3b) (i.e., *furansu ryooriten* 'French restaurants') is not an event nominal and hence, *Tookyoo* is marked by *ni*. Given the complication of the linguistic phenomena involved in choosing between *de* and *ni* and the limited amount of subcategorization information or semantic information available for verbs and nouns in the dictionary, it is almost impossible for linguists to write rules to determine this choice. We believe that this is exactly one of the situations in which machine-learning approaches such as DT can be utilized.

3.2 Experiment Design and Evaluation

3.2.1 Decision Tree model for predicting the insertion of the locative postposition

Like the Spanish experiment discussed in Section 2, two types of models were built for the Japanese experiment; one model was trained on Encarta (76.7K sentences) and the other on technical and computer documents (27.4K sentences). The variables selected for the Japanese experiment involve: (i) the lemma of the parent predicate of a location noun and its linguistic features and (ii) linguistic features associated with the location noun and the subject of the predicate. Table 8 provides the number of branching nodes, the overall accuracy and the baseline for both these models.

Model	A(Encarta)	B (technical)
# of Branching nodes	878	102
Baseline	62.27%	79.40%
Accuracy	79.10%	90.44%

Table 8: Size, baseline and accuracy of the models

In Model A (Encarta Model), 69 variables were selected and 160 variables were rejected. Features selected in Model A include the following:

- lemma of the parent predicate of a location noun
- subcategorization features of the predicate (e.g., intransitive; transitive; ditransitive; unaccusative; etc.).
- voice information for the predicate (i.e., passive or not)
- presence of modifier(s) of the predicate (e.g., time; prepositional modifier; coordination; etc.)
- presence of modifier(s) of the location noun (e.g., possessor; prepositional modifier; appositive; etc.)
- lexical semantic features of the subject of the parent predicate.

In Model B (Technical Model), 30 variables were selected and 138 variables were rejected. As for Model A, the predictors selected in Model B predominantly involve the lemmas of the parent verbs of location nouns and their subcategorization features. Precision and recall information for Model A and Model B are provided in Table 9 below.

Case	Precision (%)		Recall (%)		F-measure (%)	
Model	A	B	A	B	A	B
de	75.08	77.93	66.78	74.79	70.69	76.33
ni	81.13	93.53	86.57	94.50	83.76	94.01

Table 9: Precision and recall for *de*/*ni* in the two models

3.2.2 Evaluation

Parallel to the Spanish experiment, we used two types of blind test data; one from Encarta (1K sentences) and the other from technical documents (1K sentences). Using the DT model, we regenerated the test data and compared the regenerated strings with the original sentences to find out how many sentences were the same as the original sentences with respect to the assignment of *de*/*ni* for location nouns.

We did the same using the hand-coded rule. Our hand-coded rule for the choice between *de* and *ni* used the subcategorization features of the predicates available in our Japanese dictionary. Basically, the hand-code rule assigned *ni* to a location noun if the parent verb belongs to one of the following types of verbs: (i) directional motion verbs (e.g., *iku* ‘to go’); (ii) verbs that require a locative argument (e.g., *oku* ‘to put’); and (iii) existential verbs (e.g., *aru* or *iru* ‘to be/to exist’). For other types of verbs, the rule assigned *de* to a location noun. Table 10 gives the number of errors in the generation of the postposition *de*/*ni* in our test data sets and the accuracy as a percentage.

	Model A (Encarta)	Model B (Technical)	Hand-coded rule[8]
Encarta text	291/1K (70.90 %)	421/1 K (57.90%)	305/1 K (69.50%)
Technical manuals	440/1 K (56.00%)	215/1 K (78.50%)	240/1 K (76.00%)

Table 10: Comparison between the DT models and the hand-coded rule on blind data

In the Japanese experiment, the two models performed slightly better than the hand-coded rule with respect to the same domain test set. However, with respect to the different domain test set, they performed worse than the hand-coded rule. This means that, unlike the Spanish experiment described in Section 2, both Japanese models are sensitive to the domain of the test sets: Model A (Encarta Model) achieves 70.90% accuracy for the Encarta test data but its accuracy sharply drops for the technical data (56.00%). Model B (Technical Model) achieves 78.50% for the technical test data but its accuracy drops sharply again for the Encarta data (57.90%). That the two models are sensitive to the domain of the test sets makes sense: the types of predicates used in Encarta data are quite different from those used in technical documents.

[8] The hand-coded rule here examines the subcategorization information of the predicate.

Thus, it is reasonable to assume that the set of distinguishing features selected by Model A may not work for technical documents and vice versa.

4 Conclusion

The results of the two experiments presented in this paper show that it is possible to machine learn the contexts for non-trivial linguistic phenomena such as the selection of the copula in Spanish and *de*/*ni*-assignment in Japanese.

Particularly, in the case of the selection of the Spanish copula, the complexity of the task gives a clear advantage to the statistical approach over the hand-written rule, especially when the lemma of the adjective is included in the model.

As for sensitivity of the models to type of text, the results for the Spanish experiment show that the model trained on data coming from technical manuals performed better across different text types than the model trained on Encarta, whereas in the case of the Japanese experiment, the models were highly sensitive to the type of data on which they were trained.

Using the model for copula selection in Spanish, we have shown how the models can be used in the context of an application such as MT. With this experiment, we have also demonstrated that a model that has been trained on a monolingual (Spanish) corpus can be used on logical form structures coming from transfer.

Whether DTs are used directly in the code, or the information they provide is used to write a more accurate rule or to encode information in the dictionary, they seem to be a useful tool for addressing complex linguistic phenomena, such as the two addressed in this paper.

Acknowledgements

Our special thanks go to Michael Gamon, Simon Corston-Oliver, and Max Chickering for assistance and advice while conducting these experiments. We also are grateful to the members of the NLP group at Microsoft Research for their useful comments during the process of writing this paper.

References

Aikawa, T., Melero, M., Schwartz, L. and Wu, A. (2001). Generation for Multilingual MT. In *Proceedings of the VIIIth MT-Summit*, Santiago de Compostela (Spain), September 2001.

Bangalore, S., Chen, J. and Rambow, O. (2001). Impact of Quality and Quantity of Corpora on Stochastic Generation . In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, Pennsylvania, 2001.

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 B. (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. & Moore, R. (2002). Machine-learned contexts for linguistic operations in German sentence realization. To appear in *Proceedings of the Association for Computational Linguistics 2002*, Pennsilvania.

Heidorn, G. E. (2000). Intelligence Writing Assistance. In Dale R., Moisl H., and Somers H. (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker, New York, 1998 (published in August 2000), pages 181-207.

Menezes A. and Richardson S. (2001). A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the Association for Computational Linguistics 2001*, Toulouse, France.

Porroche Ballesteros, M. (1988). *Ser, estar y verbos de cambio*. Arco/Libros, Madrid.

Richardson, S., Dolan, W., Menezes, A. & Pinkham J. (2001). Achieving commercial-quality translation with example-based methods. In *Proceedings of the VIIIth MT-Summit*, Santiago de Compostela (Spain), September 2001, pp. 293-298.