Using Machine Learning for System-Internal Evaluation of Transferred Linguistic Representations

Michael Gamon, Hisami Suzuki, and Simon Corston-Oliver

Microsoft Research
One Microsoft Way
Redmond, WA 98052
USA
{mgamon, hisamis, simonco}@microsoft.com

Abstract

We present an automated, system-internal evaluation technique for linguistic representations in a large-scale, multilingual MT system. We use machine-learned classifiers to recognize the differences between linguistic representations generated from transfer in an MT context from representations that are produced by "native" analysis of the target language. In the MT scenario, convergence of the two is the desired result. Holding the feature set and the learning algorithm constant, the accuracy of the classifiers provides a measure of the overall difference between the two sets of linguistic representations: classifiers with higher accuracy correspond to more pronounced differences between representations. More importantly, the classifiers yield the basis for error-analysis by providing a ranking of the importance of linguistic features. The more salient a linguistic criterion is in discriminating transferred representations from "native" representations, the more work will be needed in order to get closer to the goal of producing native-like MT. We present results from using this approach on the Microsoft MT system and discuss its advantages and possible extensions.

Keywords

Machine translation, evaluation, machine learning, logical form, decision tree.

1. Introduction

The evaluation of MT systems falls into two broad categories: cross-system evaluation, and system-internal evaluation. Cross-system evaluations tend to be performed infrequently, and require system-independent evaluation metrics. System-internal evaluation, on the other hand, is customized for a particular MT architecture and needs to be performed on a regular basis, in order to provide feedback to system developers and linguists. Given the high cost in time and money that is associated with human evaluation, automating the evaluation process is crucial in system-internal evaluation (for related efforts, see also Corston-Oliver et al. (2001), Ringger et al. (2001), Bangalore et al (2000), Alshawi et al. (1998) and Su et al. (1992)). Ideally, an automated evaluation procedure should provide two kinds of information: raw numbers that can be used for quantitative analysis over time, and information that helps in qualitative error analysis.

In this paper, we propose an automated system-internal evaluation procedure for transferred semantic representations that fits these desiderata. We present results from using this evaluation procedure on the multilingual Microsoft MT system, and show how this approach can be used for error analysis.

In an MT system that transfers linguistic representations from a source language to a target language, it is important to ensure that the transferred linguistic representations are as similar as possible to the representations of the target language. In the Microsoft Natural Language Processing System (Heidorn 2000) we use logical form representations (LFs) in semantic transfer. These representations are graphs, representing the predicate argument structure and major semantic relations within a sentence. The nodes in the graph are identified by the lemma of a content word. The edges are directed, labelled arcs, indicating the semantic relationship between nodes. An example of an LF graph is given in Figure 1. Note that each node in the graph also carries attributes and features that are not shown in Figure 1.

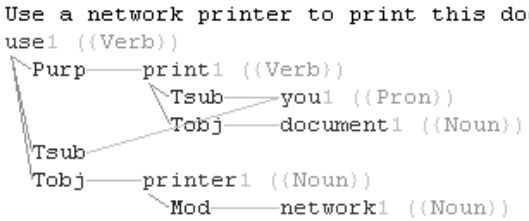

Figure 1: Logical form representation

The logical form representations should be as language-neutral as possible, but some typological differences between languages leave their mark. For example, the lack of overt markers for definiteness in Japanese leads to underspecification of definiteness features in Japanese logical forms and hence in logical forms transferred from Japanese into English. Another example involves the lack of complete resolution of all syntactic relations into semantic primitives, where that kind of analysis proves extremely hard: a German "NP + genitive NP" construction like "die Anordnung der Tabelle" is analyzed at logical form in terms of a Possessor relation, whereas the English counterpart "the position of the table" is currently analyzed as an unspecified prepositional relation.

In our MT architecture, alignments between logical form subgraphs of source and target language are identified in a training phase using aligned corpora (Menezes & Richardson 2001). These aligned subgraphs are stored in an example base. During the translation process, a sentence in the source language is analyzed and its logical

form is mapped against the example base of logical form mappings into the target language. From the subgraphs retrieved from the example base, a target logical form is constructed which serves as input into a generation component to produce a target string.

For our evaluation experiment, we automatically construct classifiers that distinguish two sets of logical forms. In one scenario, we compare logical forms from two different languages to assess the extent to which the LFs converge on similar representations. In the second scenario, we compare transferred logical forms to native logical forms in the target language to assess the contribution of the transfer component. An automated evaluation method is particularly important given that the specifics of the LF representations are continually evolving.

The classification accuracy gives an indication of how different or distant two LFs are. By inspecting the classifiers we can learn where improvements can be made. The machine learning approach that we use in this paper is a decision tree model. The reason for this choice is purely a pragmatic one: decision trees are easy to construct and easy to inspect. Nothing in our methodology, however, hinges on this particular choice.

2. Data

Our data consist of five aligned sets of 10,000 sentences from published computer software manuals and online help documents in five languages (English, French, German, Japanese, and Spanish). For the purpose of our evaluation experiments, we split the data 70/30 for training of the classifiers versus testing against held out data.

From these five sets of raw sentence data, we extracted features in the following way: First, we ran each set of sentences through the linguistic analysis component of our system, extracting linguistic features for "native" logical form representations. We then processed each of the datasets in our MT system, producing sets of features of the transferred logical forms. These two sets of extracted features are used in two comparison schemes: the native-to-native comparison scheme and the transferred-to-target scheme.

3. Features

The features used for these comparison schemes fall into two broad categories:

(a) Features that characterize various aspects of the LF graph. The 22 features in this category include:

- Boolean-valued features: e.g. `NonModalinModals` (indicating the presence of an item in the Modals attribute that is not lexically marked as a modal or auxiliary), `EmptyVP` (existence of verbal nodes with an empty subject and with no other semantic dependents)
- Integer-valued features: e.g. `Xpredcounter` (number of zero subjects), `Nodecounter` (number of nodes in the LF)
- Floating point-valued features: e.g. `BitsperNoun` (average number of features per nominal node), `Attributesperverb` (average number of semantic relations per verbal node), `Connectivity` (number of arcs divided by number of nodes).

(b) Features that capture the fit between the part of speech (POS) and the semantic relations (semrels). We currently use a simple combination of 9 POSs and 38 semrel features: for example, the LF in Figure 1 can be characterized by the POS_semrel co-occurrence features such as `Verb_Tsub`, `Verb_Tobj`, `Verb_Purp`, `Noun_Mod`, and semrel_POS co-occurrence features such as `Tsub_Pron`, `Tobj_Noun`, `Mod_Noun`, and `Purp_Verb`. Unusual pairings of POS and semrels are often indicative of ill-formed logical forms. There are currently 138 POS_semrel and 161 semrel_POS features.

These features are extracted automatically by traversing the native and transferred LF graphs. As the system develops, a different set of features might become more informative in discriminating two sets of LF representations. A machine-learning approach is therefore particularly advantageous, as the discovery of distinguishing features can be done automatically once a candidate set of features is given to the learner. Not all features are selected for all models by the decision tree learning tools.

4. The Decision Tree Models

We used a set of automated tools to construct decision trees (Chickering et al. 1997) based on the features extracted from logical forms. To avoid overfitting, we specified that nodes in the decision tree should not be split if they accounted for fewer than fifty cases. For each set of data we built decision trees at varying levels of granularity (by manipulating the prior probability of tree structures to favor simpler structures) and selected the tree with maximal accuracy. Since all datasets contain equal numbers of logical forms from each of the two categories being compared, the baseline accuracy for comparison is 50%.

4.1 Constructing the Models

We built a total of 16 different models for the following 22 logical form comparisons:

Ten models to compare native logical forms:

German versus English
Spanish versus English
French versus English
Japanese versus English
German versus Spanish
German versus French
German versus Japanese
Spanish versus French
Spanish versus Japanese
French versus Japanese

Six models to compare transferred logical forms to target logical forms:

(Japanese → English) versus English
(French → English) versus English
(German → English) versus English
(Spanish → English) versus English
(English → Japanese) versus Japanese
(English → Spanish) versus Spanish

The comparison between native logical forms is useful in two respects. First, it provides a baseline against which to compare the transferred logical forms. If the accuracy of the classifier distinguishing native LFs of language A and language B is 80%, and the accuracy of the classifier distinguishing LFs transferred from A to B to native LFs in B is only 70%, the difference of 10% accuracy can be interpreted as a gain achieved by transfer from A to B. Secondly, the comparison between native LFs helps identify areas where the logical form analysis components can be improved, i.e. can be brought closer to the goal of maximally language-independent representations (modulo typological differences).

Comparing transferred LFs to target LFs shows directly how closely the transferred LFs converge on native-like qualities. In the ideal case, the accuracy should equal the baseline of 50%, meaning that the transferred LFs are indistinguishable from native LFs. Features of high salience in the classifier indicate potential areas for improvement in alignment and transfer.

4.2 Evaluating the Decision Trees

The accuracy numbers achieved by the classifiers can be interpreted as a measure of the distance between LFs. Somewhat paradoxically, dissimilar sets of logical forms yield high classification accuracy while similar sets of logical forms yield low classification accuracy, i.e. as the LF representations converge, it becomes increasingly difficult for the classifier to distinguish them.

To facilitate interpretation of the data, we have converted accuracy percentages into a more intuitive distance measure using the following formula:

$$\text{distance} = 2*(\text{accuracy}-50)$$

Minimal accuracy of 50% (baseline) translates into 0 distance, maximal accuracy of 100% translates into a distance of 100.

Comparison of Native LFs

Figure 2 shows the results of our native-native comparison scheme. The representations for English and German are most similar (distance=50.30), while Japanese differs substantially from all other languages. Note that this should not be interpreted as a measure of typological differences among languages. Rather it reflects a combination of typological differences and the current implementation of our system.

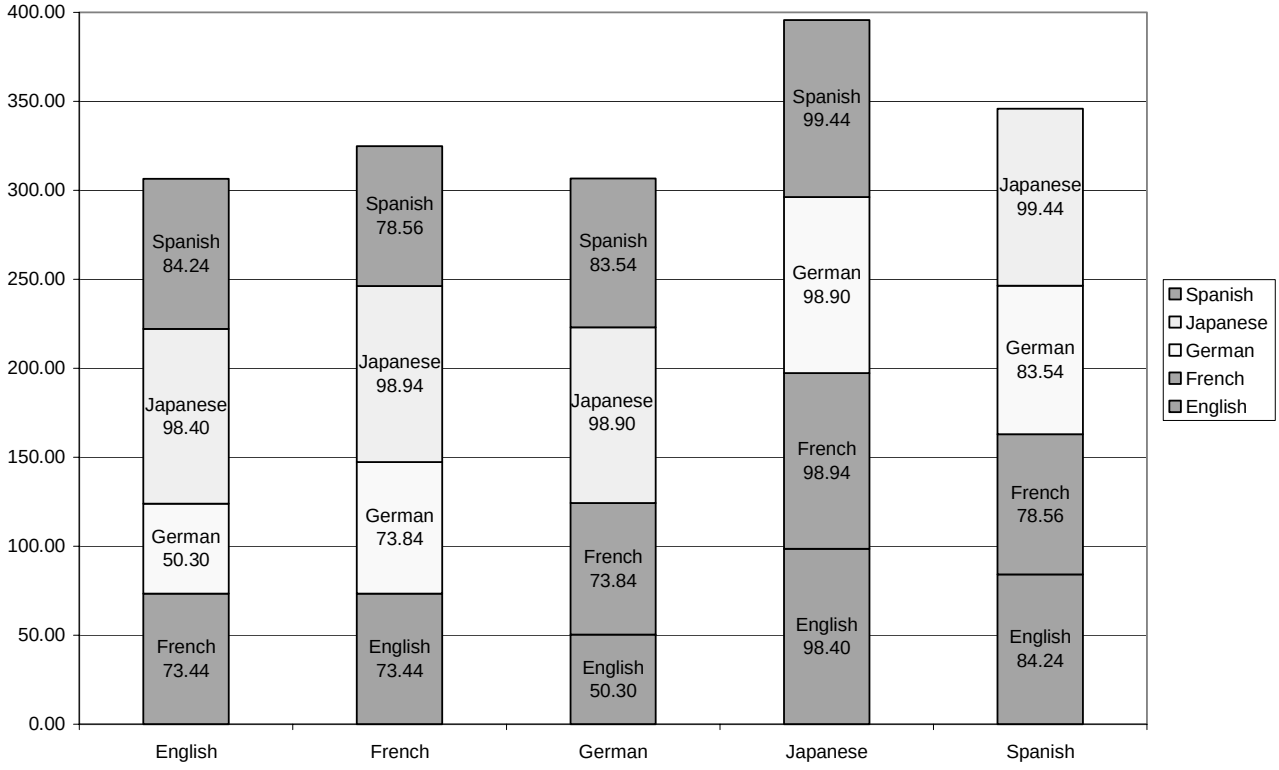

Figure 2: Comparing native logical forms

In order to disentangle these two influences we successively removed the top ranked feature and retrained the classifier. Figure 3 shows the effect of the removal of the ten most salient features that distinguish English and Japanese. We see that the measured distance between the two languages decreases sharply when the feature `BitsperNoun` is eliminated. This reflects a typological fact, namely that Japanese nouns are usually not marked for categories such as number and definiteness. The second feature `Verb_LTopic` is a system-internal feature whose assignment differs between English and Japanese.

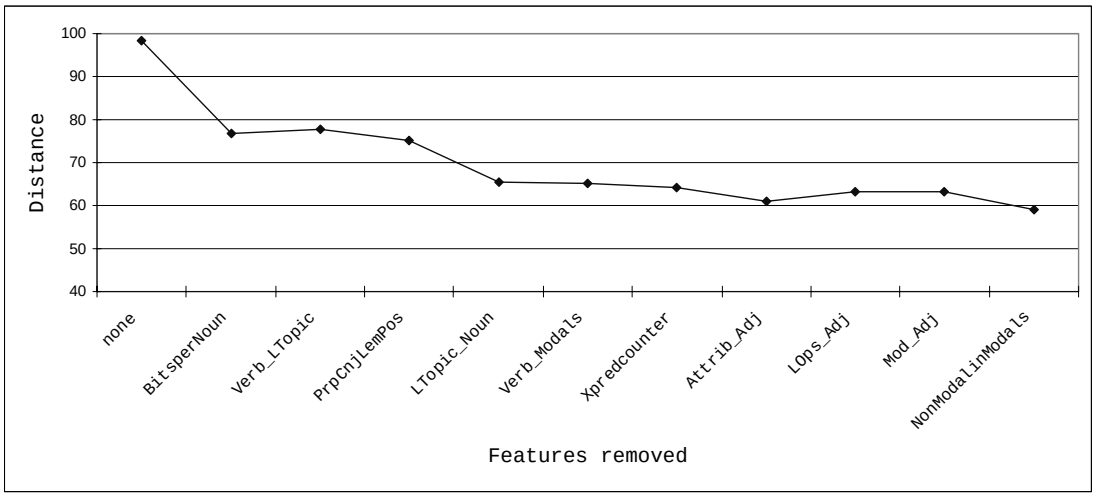

Figure 3: Elimination of features in the native Japanese/English comparison

Comparison of Transferred and Native LFs

In order to measure the effect of the transfer module we used a subset of the features described above in section 3. Two kinds of features were eliminated. The features that count bits overwhelmed other features and were mostly indicative of typological characteristics of the languages rather than areas requiring system modification. The second set of eliminated features were those referring to attributes that are deliberately excluded from the transfer process. In Figure 4 we compare the native-native LFs and transferred vs. native LFs for the six language pairs for which transfer has been implemented. The difference illustrated reflects the impact of the transfer module. For example, consider the Spanish-English language pair, which has received the most attention. The distance between the native LFs is 68.50. When Spanish LFs are transferred to English and compared to native English LFs, the distance decreases to 46.00. A very small adverse effect (-1.10) in the case of German-English is most likely due to overfitting.

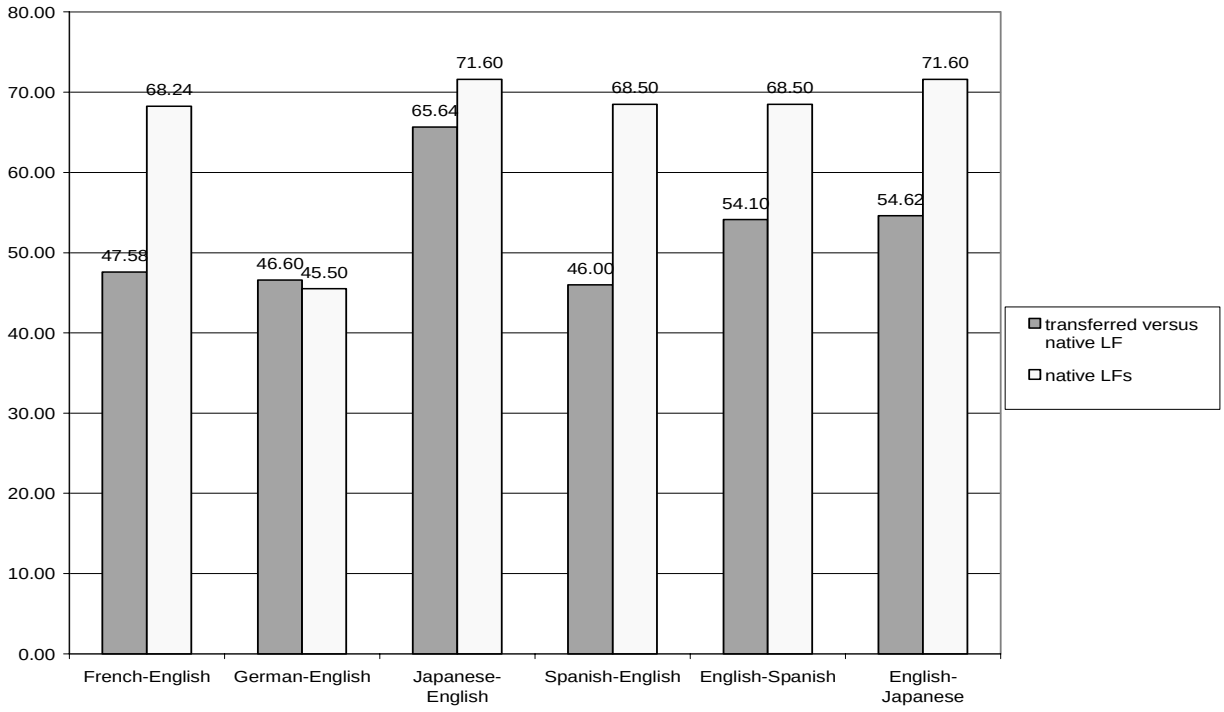

Figure 4: Measuring the contribution of transfer

4.3 Inspecting the Decision Trees

We use a modified version of the Dnetviewer tool (Heckerman et al. 2000) to visually inspect the decision trees. This tool allows a linguist to view the sentences that are covered by a particular leaf node in the decision tree, i.e. that exhibit the properties identified by the features on the path from root node to leaf node. Figures 5 and 6 show different views in the Dnetviewer tool. Figure 5 shows a partial view of the decision tree, displaying some of the most salient features distinguishing Spanish and Japanese. The leaf nodes display a probability distribution over the possible states of the target feature. For the binary classification shown in Figure 5, the dark portion of the rectangle indicates p(Japanese) of the logical forms covered by that leaf node, while the light gray indicates p(Spanish) of those LFs. As Figure 6 shows, clicking on the rectangle under a leaf node (the circled leaf node in Figure 5) displays the sentences covered by that leaf node.

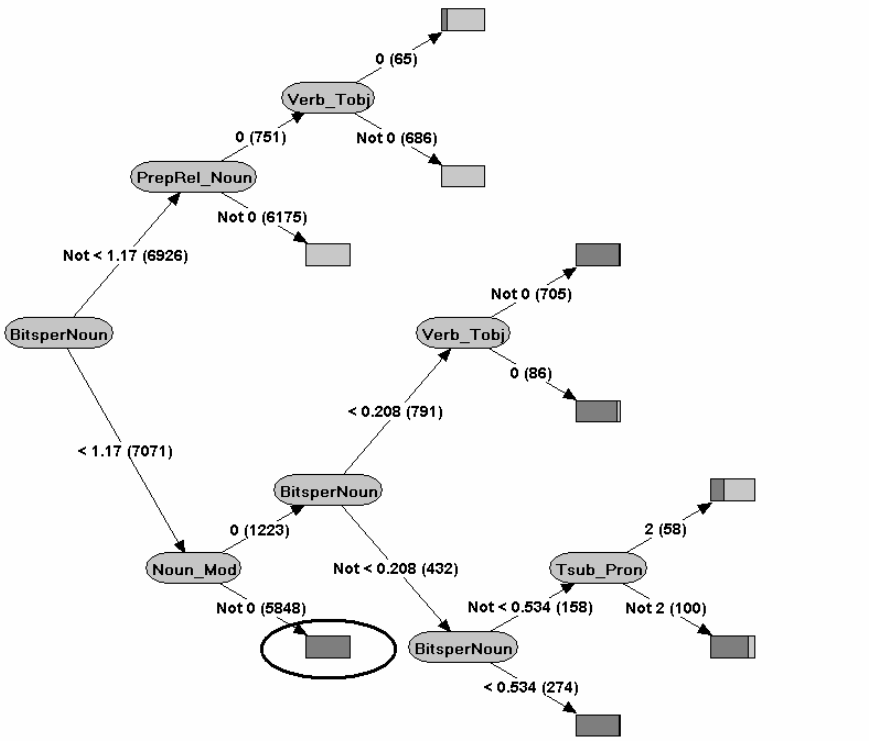

Figure 5: Dnetbrowser view of the decision tree distinguishing native Spanish and Japanese

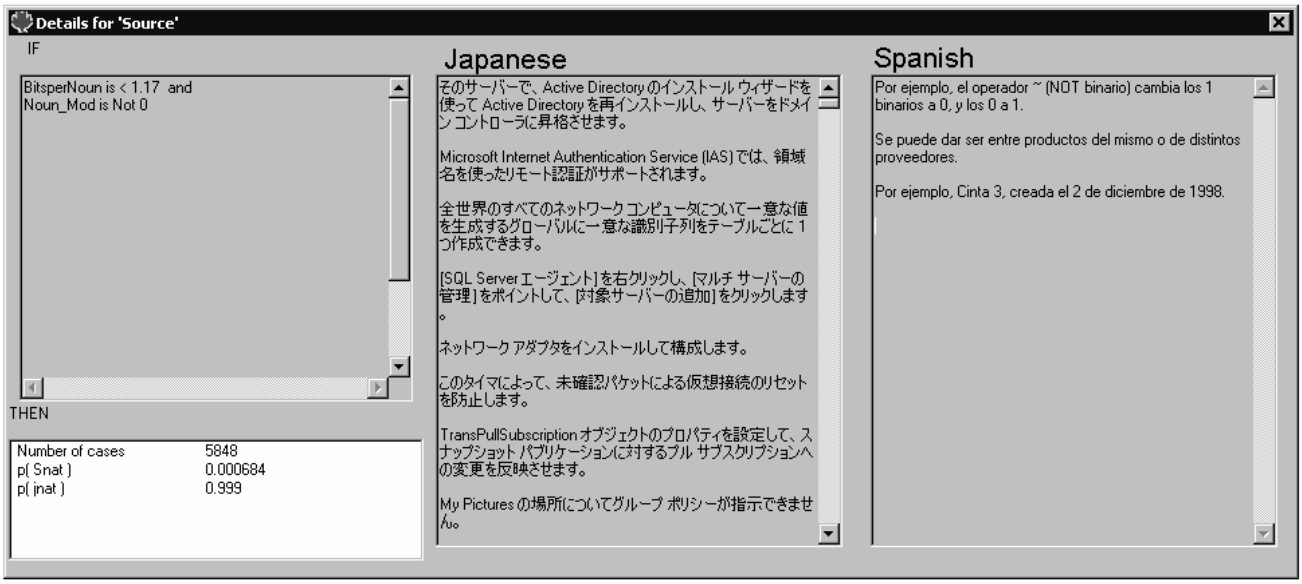

Figure 6: Dnetbrowser view of the sentences under a leaf node of the decision tree

5. Conclusion

The approach that we have described has a number of significant advantages. First, this evaluation technique can be fully automated once the set of features has been determined. For example, it could be run on a daily or weekly basis to give reports to measure progress. Secondly, this approach is completely customizable. Any reasonable selection of linguistic features can be used to build a model, and the evaluation process itself is independent of the framework of a given MT system or the specifics of the representations used. Finally, we have shown that the level of error categorization that the decision tree models goes beyond purely quantitative assessment of the quality of the transfer process. While this cannot replace detailed human error analysis for debugging, it provides a high-level first categorization of problem areas.

A possible future use of our method is to evaluate individual logical forms. Once the classifiers have been trained, they can assign confidence scores to individual transferred LFs. Those confidence scores can be used, for example, to trigger repair strategies on representations that are very different from what an LF in the target language should typically look like.

Acknowledgements

Our thanks go to Eric Ringger and Max Chickering for programming assistance with the decision tree tools and to members of the NLP group at Microsoft Research for valuable feedback.

References

Alshawi, H., S Bangalore, & S Douglas. (1998). Automatic acquisition of hierarchical transduction models for machine translation. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, (Vol I pp. 41-47). Montreal, Canada.

Bangalore, S., O Rambow, & S. Whittaker. (2000). Evaluation Metrics for Generation. In Proceedings of the International Conference on Natural Language Generation (INLG 2000) (pp. 1-13). Mitzpe Ramon, Israel.

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

Corston-Oliver, S., M. Gamon & C. Brockett. (2001): A machine learning approach to the automatic evaluation of machine translation. To be presented at ACL 2001.

Heckerman, D., D. M. Chickering, C. Meek, R. Rounthwaite & C. Kadie. (2000). Dependency networks for inference, collaborative filtering and data visualization. Journal of Machine Learning Research 1,49-75.

Heidorn, G.E. (2000). Intelligent writing assistance. In R. Dale, H. Moisl and H. Somers (Eds.). Handbook of Natural Language Processing (pp. 181-207). New York: Marcel Dekker.

Menezes, A. & Richardson, S. (2001). A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In review.

Ringger, E., M. Corston-Oliver & R. Moore. (2001). Using Word-Perplexity for Automatic Evaluation of Machine Translation. In review.

Su, K., M. Wu, & Chang, J. (1992). A new quantitative quality measure for machine translation systems. In Proceedings of COLING-92 (pp. 433-439). Nantes, France.