Improving an Existing RBMT System by Stochastic Analysis

Christian Federmann, Sabine Hunsicker
DFKI – Language Technology Lab
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, GERMANY
E-mail: {cfedermann,sabine.hunsicker}@dfki.de

Abstract

In this paper we describe how an existing, rule-based machine translation (RBMT) system that follows a transfer-based translation approach can be improved by integrating stochastic knowledge into its analysis phase. First, we investigate how often the rule-based system selects the wrong analysis tree to determine the potential benefit from an improved selection method. Afterwards we describe an extended architecture that allows integrating an external stochastic parser into the analysis phase of the RBMT system. We report on the results of both automatic metrics and human evaluation and also give some examples that show the improvements that can be obtained by such a hybrid machine translation setup. While the work reported on in this paper is a dedicated extension of a specific rule-based machine translation system, the overall approach can be used with any transfer-based RBMT system. The addition of stochastic knowledge to an existing rule-based machine translation system represents an example of a successful, hybrid combination of different MT paradigms into a joint system.

Keywords: Machine Translation, Hybrid Machine Translation, Stochastic Parsing, System Combination

1. Introduction

Rule-based machine translation (RBMT) systems that employ a transfer-based translation approach, highly depend on the quality of their analysis phase as it provides the basis for its later processing phases, namely transfer and generation. Any parse failures encountered in the initial analysis phase will proliferate and cause further errors in the following phases. Very often, bad translation results can be traced back to incorrect analysis trees that have been computed for the respective input sentences. Consequently, any improvements that can be achieved for the analysis phase of some RBMT system lead to improved translation output, which makes this an interesting topic in the context of hybrid machine translation.

In this paper we describe how a stochastic parser can supplement the rule-based analysis phase of a commercial RBMT system. The system in question is the rule-based engine Lucy LT. This engine uses a sophisticated RBMT transfer approach with a long research history, as explained in detail in (Wolf et al., 2010). The output of its analysis phase is a forest containing a small number of tree structures. For this study we investigated if the existing rule base of the Lucy LT system chooses the best tree from the analysis forest and how the selection of this best tree out of the set of candidates can be improved by adding stochastic knowledge to the RBMT system.

The paper is structured in the following way: in Section 2 we describe the Lucy RBMT system and its transfer-based architecture. Afterwards, in Section 3, we provide details on the integration of a stochastic parser into the Lucy analysis phase of this rule-based system. Section 4 describes the experiments we performed and reports the results of both automated metrics and human evaluation efforts before Section 5 discusses some examples that show how the proposed approach has improved or degraded machine translation quality. Finally, in Section 6, we conclude and provide an outlook on future work in this area.

2. Lucy System Architecture

The Lucy LT engine is a renowned RMBT system that follows a classical, transfer-based translation approach. The system first analyses the given source sentence resulting in a forest of several analysis trees. One of these trees is then selected (as "best" analysis) and transformed in the transfer phase into a tree structure from which the target text can be generated.

It is clear that any errors that occur during the initial analysis phase proliferate and cause negative side effects on the quality of the resulting translation. As the analysis phase is of special importance, we describe it in more detail. The Lucy LT analysis consists of several phases:

1) The input is tokenised with regards to the source language lexicon.
2) The resulting tokens then undergo a morphological analysis, which identifies possible combinations of allomorphs for a token.
3) This leads to a chart which forms the basis for the actual parsing, using a head-driven strategy. Special treatment is performed for the analysis of multi-word expressions and also for verbal framing.

At the end of the analysis, there is an extra phase named phrasal analysis that is called whenever the grammar was not able to construct a legal constituent from all the elements of the input. This happens in several different scenarios:

- The input is ungrammatical according to the LT analysis grammar.
- The category of the derived constituent is not one of the allowed categories.
- A grammatical phenomenon in the source sentence is not covered.
- There are missing lexical entries for the input sentence.

During the phrasal analysis, the LT engine collects all partial trees and greedily constructs an overall interpretation of the chart. Based on our findings from experiments with the Lucy LT engine, phrasal analyses are performed for more than 40% of the sentences from our test sets and very often result in bad translations.

Each resulting analysis tree, independent of whether it is a grammatical or phrasal analysis, is also assigned an integer score by the grammar. The tree with the highest score is then handed over to the transfer phase, thus pre-defining the final translation output.

3. Adding Stochastic Analysis

An initial, manual evaluation of the translation quality based on the tree selection of the analysis phase showed that there is potential for improvement. For this, we changed the RBMT system to produce translations for all its analysis trees and ranked them according to their quality. In many cases, one of the alternative trees would have lead to a better translation.

Next to the assigned score, we examined the significance of two other features:

1) The size of the analysis trees themselves, and
2) The tree edit distance of each analysis candidate to a stochastic parse tree.

An advantage of stochastic parsing lies in the fact that parsers from this class can deal very well even with ungrammatical or unknown output, which we have seen is problematic for a rule-base parser. We decided to make use of the Stanford Parser as described in (Klein & Manning, 2003), which uses an unlexicalised, probabilistic context-free grammar trained on the Penn Treebank. We parse the original source sentence with this PCFG grammar to get a stochastic parse tree that can be compared to the trees from the Lucy analysis forest.

In our experiments, we compare the stochastic parse tree with the alternatives given by Lucy LT. Tree comparison is implemented based on the Tree Edit Distance, as originally defined in (Zhang & Shasha, 1989}. In analogy to the Word Edit or Levenshtein Distance, the distance between two trees is the number of editing actions that are required to transform the first tree into the second tree. The Tree Edit Distance knows three actions:

- Insertion
- Deletion
- Renaming (substitution in Levenshtein Distance)

We use a normalised version of the Tree Edit Distance to estimate the quality of the trees from the Lucy analysis forest. The integration of the stochastic selection has been possible by using an adapted version of the rule-based system, which allowed performing the selection of the analysis tree from an external process.

4. Experiments

Two test sets were used in our experiments. The first test set was taken from the WMT shared task 2008, consisting of a section of data from Europarl (Koehn, 2005). The second test set, which was taken from the WMT shared task 2010 contained news text. Phrasal analyses caused by unknown lexical items occurred more often in the news text, as that text sort tends to more often use colloquial expressions. In our experiments, we translated from English→German;

evaluation was performed using both automated metrics and human evaluation using an annotation tool similar to e.g. Appraise (Federmann, 2010).

First, only the Tree Edit Distance and internal score from the Lucy analysis phase were used and we select the tree with the lowest edit distance. If the lowest distance holds for two or more trees, the tree with the highest LT internal score is chosen. Later we added the size of the candidate trees as an additional feature, with a bias to prefer larger trees as they proved to create better translations in our experiments. Results from automatic scoring using BLEU (Papineni et al., 2001) and the derived NIST score are reported in Table 1 and Table 2 for test set #1 and test set #2, respectively. The BLEU scores for the new translations are a little bit worse, but still comparable to the quality of the original translations. The difference is not statistically significant.

Test set #1	BLEU	NIST
Baseline	0.1100	4.4059
Stochastic Selection	0.1096	4.3946

Table 1: Automatic scores for test set #1.

Test set #2	BLEU	NIST
Baseline	0.1529	5.5725
Stochastic Selection	0.1514	5.5469
Selection+Size	0.1511	5.5341

Table 2: Automatic scores for test set #2.

We also manually evaluated a sample of 100 sentences. For this, we created all possible translations for each phrasal analysis and had human annotators judge on their quality. Then, we checked whether our stochastic selection mechanism returned a tree that led to the best translation. In case it did not, we investigated the reasons for this. Sentences for which all trees created the same translation were skipped.

Table 3 shows the error rate of our stochastic analysis component that chose the optimal tree for 56% of the sentences, while Table 4 shows the selection reasons that resulted in the selection of a non-optimal tree. We also see that the minimal tree edit distance seems to be a good feature to use for comparisons, as it holds for 71% of the trees, including those examples where the best tree was not scored highest by the LT engine. This also means that additional features for choosing the tree out of the group of trees with the minimal edit distance are required.

Best translation?	Yes (56%)	No (44%)
Minimal distance?	Yes (71%)	No (29%)

Table 3: Error rate of the stochastic analysis.

More than 50 tokens in source	36.4%
Time-out before best tree is reached	29.5%
Chosen tree had minimal distance	34.1%

Table 4: Reasons for erroneous tree selection.

Even for the 29% of sentences, in which the optimal tree was not chosen, little quality was lost: in 75.86% of those cases, the translations didn't change at all (obviously the trees resulted in equal translation output). In the remaining cases the translations were divided evenly between slight degradations and equal quality.

In cases when the best tree was not chosen, the first tree (which is the default tree) was selected in 70.45% . This is due to a combination of robustness factors that are implemented in the RBMT system and have been beyond our control in the experiments. The LT engine has several different indicators that may each throw a time-out exception, if, for example, the analysis phase takes too long to produce a result. To avoid getting time-out errors, only sentences with up to 50 tokens are treated by our stochastic selection mechanism. Additionally, the component itself checks the processing time and returns intermediate results, if this limit is reached. We are currently working on eliminating this time-out issue as it prevents us from driving our approach to its full potential.

As with the internal score, we see that the Tree Edit Distance on its own is a good indicator of the quality of the analysis, but that additional features are required to prevent suboptimal decisions to be taken. As such, we included the size of the trees. Here the bigger trees are preferred to smaller ones as experimental results have confirmed that these are more likely to produce better translations.

The manual evaluation shows results that are similar to the automated metrics. We are currently investigating in more detail what happened in case of the degradations to improve that misbehaviour. It seems as if additional features might be needed to more broadly improve the rule-based machine translation engine using our stochastic selection mechanism.

5. Examples

We now provide some examples from our experiments that illustrate how the stochstic selection mechanism changed the translation output of the rule-based system. For example, the analysis of the following sentence is now correct:

Source: "They were also protesting against bad pay conditions and alleged persecution."

Translation A: "Sie protestierten auch gegen schlechte Soldbedingungen und behaupteten Verfolgung."

Translation B: "Sie protestierten auch gegen schlechte Soldbedingungen und angebliche Verfolgung."

Translation A is the default translation. The analysis tree associated with this translation contains a node for the adjective "alleged" which is wrongly parsed as a verb.

The next example shows how an incorrect word order problem is fixed:

Source: "If the finance minister can't find the money elsewhere, the project will have to be aborted and sanctions will be imposed, warns Janota."

Translation A: "Wenn der Finanzminister das Geld nicht anderswo finden kann, das Projekt abgebrochen werden müssen wird und Sanktionen auferlegt werden werden, warnt Janota."

Translation B: "Wenn der Finanzminister das Geld nicht anderswo finden kann, wird das Projekt abgebrochen werden müssen und Sanktionen werden auferlegt werden, warnt Janota."

Lexical items are associated with a domain area in the lexicon of the rule-based system. Items that are contained within a different domain area than the input text are still accessible, but items in the same domain are preferred. In the following example, this leads to an incorrect disambiguation of multi-word expressions:

Source: "Apparently the engine blew up in the rocket's third phase."

Translation A: "Offenbar blies der Motor hinauf die dritte Phase der Rakete in."

Translation B: "Offenbar flog der Motor in der dritten Phase der Rakete in die Luft."

Again, the stochastic selection allows choosing a better tree, which leads to the correct idiomatic translation. Something similar happens in the following case:

Source: "As of January, they should be paid for by the insurance companies and not compulsory."

Translation A: "Ab Januar sollten sie für von den Versicherungsgesellschaften und nicht obligatorisch bezahlt werden."

Translation B: "Ab Januar sollten sie von den Versicherungsgesellschaften und nicht obligatorisch gezahlt werden."

These changes remain at a rather local scope, but we also have observed instances where the sentence improves globally:

Source: "In his new book, 'After the Ice', Alun Anderson, a former editor of New Scientist, offers a clear and chilling account of the science of the Arctic and a gripping glimpse of how the future may turn out there."

Translation A: "In seinem neuen Buch bietet Alun Anderson, ein früherer Redakteur von Neuem Wissenschaftler, 'Nach dem Eis' einen klaren und kalten Bericht über die Wissenschaft der Arktis und einen spannenden Blick davon an, wie die Zukunft sich hinaus dort drehen kann."

Translation B: "In seinem neuen Buch, 'Nach dem Eis', bietet Alun Anderson, ein früherer Redakteur von Neuem Wissenschaftler, einen klaren und kalten Bericht über die Wissenschaft der Arktis und einen spannenden Blick davon an, wie die Zukunft sich hinaus dort drehen kann."

In translation A, the name of the book, "After the Ice", has been moved to an entirely different place in the sentence, removing it from its original context.

6. Conclusion and Outlook

The analysis phase proves to be crucial for the quality of the translation in rule-based machine translation systems. In this paper, we have shown that it is possible to improve the analysis results of such a rule-based engine by introducing a better selection method for the trees created by the grammar. Our experiments show that the selection itself is not a trivial task and requires fine-grained selection criteria.

While the work reported on in this paper is a dedicated extension of a specific rule-based machine translation system, the overall approach can be used with any transfer-based RBMT system. Future work will concentrate on the circumvention of e.g. the time-out errors that prevented a better performance of the stochastic selection mechanism. Also, we will more closely investigate the issue of decreased translation quality and experiment with additional decision factors that may help to alleviate the negative effects.

The addition of stochastic knowledge to an existing rule-based machine translation system represents an example of a successful, hybrid combination of different MT paradigms into a joint system.

7. Acknowledgements

This work was also supported by the EuroMatrixPlus project (IST-231720) that is funded by the European Community under the Seventh Framework Programme for Research and Technological Development.

8. References

Federmann, C. (2010). Appraise: An open-source toolkit for manual phrase-based evaluation of translations. In Proceedings of the Seventh conference on International Language Resources and Evaluation. European Language Resources Association (ELRA).

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the ACL, pages 423–430.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In Proceedings of the MT Summit 2005.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. IBM Research Report RC22176 (W0109-022), IBM.

Wolf, P., Alonso, J., Bernardi, U., and Llorens, A. (2010). EuroMatrixPlus WP2.2: Study of Example-Based Modules for LT Transfer.

Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput., 18:1245–1262.