Can Machine Learning Algorithms Improve Phrase Selection in Hybrid Machine Translation?

Christian Federmann
Language Technology Lab
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, GERMANY
cfedermann@dfki.de

Abstract

We describe a substitution-based, hybrid machine translation (MT) system that has been extended with a machine learning component controlling its phrase selection. Our approach is based on a rule-based MT (RBMT) system which creates template translations. Based on the generation parse tree of the RBMT system and standard word alignment computation, we identify potential "translation snippets" from one or more translation engines which could be substituted into our translation templates. The substitution process is controlled by a binary classifier trained on feature vectors from the different MT engines. Using a set of manually annotated training data, we are able to observe improvements in terms of BLEU scores over a baseline version of the hybrid system.

1 Introduction

In recent years, the overall quality of machine translation output has improved greatly. Still, each technological paradigm seems to suffer from its own particular kinds of errors: statistical MT (SMT) engines often show poor syntax, while rule-based MT systems suffer from missing data in their vocabularies. Hybrid approaches try to overcome these typical errors by combining techniques from both (or even more) paradigms in an optimal manner.

In this paper we report on experiments with an extended version of the hybrid system we develop in our group (Federmann and Hunsicker, 2011; Federmann et al., 2010). We take the output from an RBMT engine as "translation template" for our hybrid translations and substitute noun phrases[1] by translations from one or several MT engines[2]. Even though a general increase in quality could be observed in previous work, our system introduced errors of its own during the substitution process. In an internal error analysis, these degradations could be classified in the following way:

- external translations were incorrect;
- the structure degraded through substitution;
- phrase substitution failed.

Errors of the first class cannot be corrected, as we do not have an easy way of knowing when the translation obtained from an external MT engine is incorrect. The other classes could, however, be eliminated by introducing additional steps for pre- and post-processing as well as by improving the hybrid substitution algorithm itself. So far, our algorithm relied on many, hand-crafted decision factors; in order to improve translation quality and processing speed, we decided to apply machine learning methods to our training data to train a linear classifier which could be used instead.

This paper is structured in the following way. After having introduced the topics of our work in Section 1, we give a description of our hybrid MT system architecture in Section 2. Afterwards we describe in detail the various decision factors we

[1] We are focusing on noun phrases for the moment as these worked best in previous experiments with substitution-based MT; likely because they usually form consecutive spans in the translation output.

[2] While this could be SMT systems only, our approach supports engines from all MT paradigms. If not all features inside our feature vectors can be filled using the output of some system X, we use defaults as fallback values.

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 113–118,
Avignon, France, April 23 - 27 2012.

have defined and how these could be used in feature vectors for machine learning methods in Section 3. Our experiments with the classifier-based, hybrid MT system are reported in Section 4. We conclude by giving a summary of our work and then provide an outlook to related future work in Section 5.

2 Architecture

Our hybrid machine translation system combines translation output from:

a) the Lucy RBMT system, described in more detail in (Alonso and Thurmair, 2003), and

b) one or several other MT systems, e.g. Moses (Koehn et al., 2007), or Joshua (Li et al., 2009).

The rule-based component of our hybrid system is described in more detail in section 2.2 while we provide more detailed information on the "other" systems in section 2.3.

2.1 Basic Approach

We first identify noun phrases inside the rule-based translation and compute the most probable correspondences in the translation output from the other systems. For the resulting phrases, we apply a factored substitution method that decides whether the original RBMT phrase should be kept or rather be replaced by one of the candidate phrases. As this shallow substitution process may introduce errors at phrase boundaries, we perform several post-processing steps that clean up and finalise the hybrid translation result. A schematic overview of our hybrid system and its main components is given in figure 1.

2.2 Rule-Based Translation Templates

We obtain the "translation template" as well as any linguistic structures from the RBMT system. Previous work with these structures had shown that they are usually of a high quality, supporting our initial decision to consider the RBMT output as template for our hybrid translation approach. The Lucy translation output can include markup that allows to identify unknown words or other phenomena.

The Lucy system is a transfer-based RBMT system that performs translation in three phases, namely *analysis*, *transfer*, and *generation*. Tree

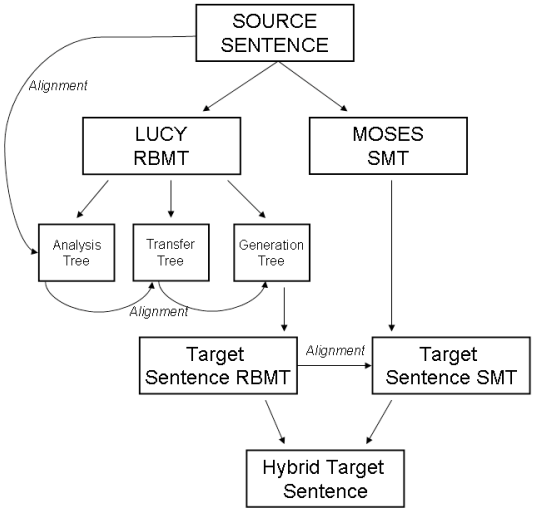

Figure 1: Schematic overview of the architecture of our substitution-based, hybrid MT system.

structures for each of the translation phases can be extracted from the Lucy system to guide the hybrid system. Only the 1-best path through the three phases is given, so no alternative translation possibilities can be extracted from the given data; a fact that clearly limits the potential for more deeply integrated hybrid translation approaches. Nonetheless, the availability of these 1-best trees already allowed us to improve the translation quality of the RBMT system as we had shown in previous work.

2.3 Substitution Candidate Translations

We use state-of-the-art SMT systems to create statistical, phrase-based translations of our input text, together with the bidirectional word alignments between the source texts and the translations. Again, we make use of markup which helps to identify unknown words as this will later be useful in the factored substitution method.

Translation models for our SMT systems were trained with lower-cased and tokenised Europarl (Koehn, 2005) training data. We used the LDC Gigaword corpus to train large scale language models and tokenised the source texts using the tokenisers available from the WMT shared task website[3]. All translations are re-cased before they are sent to the hybrid system together with the word alignment information.

[3]Available at `http://www.statmt.org/wmt12/`

The hybrid MT system can easily be adapted to support other translation engines. If there is no alignment information available directly, a word alignment tool is needed as the alignment is a key requirement for the hybrid system. For part-of-speech tagging and lemmatisation we used the TreeTagger (Schmid, 1994).

2.4 Aligning RBMT and SMT Output

We compute alignment in several components of the hybrid system, namely:

source-text-to-tree: we first find an alignment between the source text and the corresponding analysis tree. As Lucy tends to subdivide large sentences into several smaller units, it sometimes becomes necessary to align more than one tree structure to a source sentence.

analysis-transfer-generation: for each of the analysis trees, we re-construct the path from its tree nodes, via the transfer tree, to the corresponding generation tree nodes.

tree-to-target-text: similarly to the first alignment process, we find a connection between generation tree nodes and the corresponding translation output of the RBMT system.

source-text-to-tokenised: as the Lucy RBMT system works on non-tokenised input text and our SMT systems take tokenised input, we need to align the original source text with its tokenised form.

Given the aforementioned alignments, we can then correlate phrases from the rule-based translation with their counterparts from the statistical translations, both on source or target side. As our hybrid approach relies on the identification of such phrase pairs, the computation of the different alignments is critical to achieve a good system combination quality.

All tree-based alignments can be computed with a very high accuracy. However, due to the nature of statistical word alignment, the same does not hold for the alignment obtained from the SMT systems. If the alignment process produces erroneous phrase tables, it is very likely that Lucy phrases and their "aligned" SMT matches simply do not fit the "open slot" inside the translation template. Or put the other way round: the better the underlying SMT word alignment, the greater the potential of the hybrid substitution approach.

2.5 Factored Substitution

Given the results of the alignment process, we can then identify "interesting" phrases for substitution. Following our experimental setup from the WMT10 shared task, we again decided to focus on *noun phrases* as these seem to be best-suited for in-place swapping of phrases.

To avoid errors or problems with non-matching insertions, we want to keep some control on the substitution process. As the substitution process proved to be a very difficult task during previous experiments with the hybrid system, we decided to use machine learning methods instead. For this, we refined our previously defined set of decision factors into values $v \in \mathbb{R}$ which allows to combine them in feature vectors $x_i = v_1 \dots v_p$. We describe the integration of the linear classifier in more detail in Section 3.

2.6 Decision Factors

We used the following factors:

1. **frequency:** frequency of a given candidate phrase compared to total number of candidates for the current phrase;
2. **LM(phrase):** language model (LM) score of the phrase;
3. **LM(phrase)+1:** phrase with right-context;
4. **LM(phrase)-1:** phrase with left-context;
5. **Part-of-speech match?:** checks if the part-of-speech tags of the left/right context match the current candidate phrase's context;
6. **LM(pos)** LM score for part-of-speech (PoS);
7. **LM(pos)+1** PoS with right-context;
8. **LM(pos)-1** PoS with left-context;
9. **Lemma** checks if the lemma of the candidate phrase fits the reference;
10. **LM(lemma)** LM score for the lemma;
11. **LM(lemma)+1** lemma with right-context;
12. **LM(lemma)-1** lemma with left-context.

2.7 Post-processing Steps

After the hybrid translation has been computed, we perform several post-processing steps to clean up and finalise the result:

cleanup first, we perform some basic cleanup such as whitespace normalisation;

multi-words then, we take care of multi-word expressions. Using the tree structures from the RBMT system we remove superfluous whitespace and join multi-words, even if they were separated in the substituted phrase;

prepositions finally, prepositions are checked as experience from previous work had shown that these contributed to a large extent to the amount of avoidable errors.

3 Machine Learning-based Selection

Instead of using hand-crafted decision rules in the substitution process, we aim to train a classifier on a set of annotated training examples which may be better able to extract useful information from the various decision factors.

3.1 Formal Representation

Our training set D can be represented formally as

$$D = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^{n} \quad (1)$$

where each x_i represents the *feature vector* for sentence i while the y_i value contains the annotated class information. We use a binary classification scheme, simply defining 1 as "good" and -1 as "bad" translations. In order to make use of machine learning methods such as decision trees (Breiman et al., 1984), SVMs (Vapnik, 1995), or the Perceptron (Rosenblatt, 1958) algorithm, we have to prepare our training set with a sufficiently large number of annotated training instances. We give further details on the creation of an annotated training set in section 4.1.

3.2 Creating Hybrid Translations

Using suitable training data, we can train a *binary classifier* (using either a decision tree, an SVM, or the Perceptron algorithm) that can be used in our hybrid combination algorithm.

The *pseudo-code* in Algorithm 1 illustrates how such a classifier can be used in our hybrid MT decoder.

Algorithm 1 Decoding using linear classifier

```
good_candidates ← []
for all substitution candidates C_i do
    if CLASSIFY(C_i) == "good" then
        good_candidates ← C_i
    end if
end for
C_best ← SELECT-BEST(good_candidates)
SUBSTITUTE-IN(C_best)
```

We first collect all "good" translations using the `CLASSIFY()` operation, then choose the "best" candidate for substitution with `SELECT-BEST()`, and finally integrate the resulting candidate phrase into the generated translation using `SUBSTITUTE-IN()`. `SELECT-BEST()` could use system-specific confidences obtained during the tuning phase of our hybrid system. We are still experimenting on its exact definition.

4 Experiments

In order to obtain initial experimental results, we created a decision-tree-based variant of our hybrid MT system. We implemented a decision tree learning module following the CART algorithm (Breiman et al., 1984). We opted for this solution as decision trees represent a straightforward first step when it comes to integrating machine learning into our hybrid system.

4.1 Generating Training Data

For this, we first created an annotated data set. In a nutshell, we computed feature vectors and potential substitution candidates for all noun phrases in our training data[4] and then collected data from human annotators which of the substitution candidates were "good" translations and which should rather be considered "bad" examples. We used Appraise (Federmann, 2010) for the annotation, and collected 24,996 labeled training instances with the help of six human annotators. Table 1 gives an overview of the data sets characteristics.

	Translation Candidates		
	Total	"good"	"bad"
Count	24,996	10,666	14,330

Table 1: Training data set characteristics

[4]We used the WMT12 "newstest2011" development set as training data for the annotation task.

	Hybrid Systems		Baseline Systems			
	Baseline	+Decision Tree	Lucy	Linguatec	Moses	Joshua
BLEU	13.9	14.2	14.0	14.7	14.6	15.9
BLEU-cased	13.5	13.8	13.7	14.2	13.5	14.9
TER	0.776	0.773	0.774	0.775	0.772	0.774

Table 2: Experimental results comparing baseline hybrid system using hand-crafted decision rules to a decision-tree-based variant; both applied to the WMT12 "newstest2012" test set data for language pair English→German.

4.2 Experimental Results

Using the annotated data set, we then trained a decision tree and integrated it into our hybrid system. To evaluate translation quality, we created translations of the WMT12 "newstest2012" test set, for the language pair English→German, with a) a baseline hybrid system using hand-crafted decision rules and b) an extended version of our hybrid system using the decision tree.

Both hybrid systems relied on a Lucy translation template and were given additional translation candidates from another rule-based system (Aleksic and Thurmair, 2011), a statistical system based on the Moses decoder, and a statistical system based on Joshua. If more than one "good" translation was found, we used the hand-crafted rules to determine the single, winning translation candidate (implementing SELECT-BEST in the simplest, possible way).

Table 2 shows results for our two hybrid system variants as well as for the individual baseline systems. We report results from automatic BLEU (Papineni et al., 2001) scoring and also from its case-sensitive variant, BLEU-cased.

4.3 Discussion of Results

We can observe improvements in both BLEU and BLEU-cased scores when comparing the decision-tree-based hybrid system to the baseline version relying on hand-crafted decision rules. This shows that the extension of the hybrid system with a learnt classifier can result in improved translation quality.

On the other hand, it is also obvious, that the improved hybrid system was not able to outperform the scores of some of the individual baseline systems; there is additional research required to investigate in more detail how the hybrid approach can be improved further.

5 Conclusion and Outlook

In this paper, we reported on experiments aiming to improve the phrase selection component of a hybrid MT system using machine learning. We described the architecture of our hybrid machine translation system and its main components.

We explained how to train a decision tree based on feature vectors that emulate previously used, hand-crafted decision factors. To obtain training data for the classifier, we manually annotated a set of 24,996 feature vectors and compared the decision-tree-based, hybrid system to a baseline version. We observed improved BLEU scores for the language pair English→German on the WMT12 "newstest2012" test set.

Future work will include experiments with other machine learning classifiers such as SVMs. It will also be interesting to investigate what other features can be useful for training. Also, we intend to experiment with heterogeneous feature sets for the different source systems (resulting in large but sparse feature vectors), adding system-specific annotations from the various systems and will investigate their performance in the context of hybrid MT systems.

Acknowledgments

This work has been funded under the Seventh Framework Programme for Research and Technological Development of the European Commission through the T4ME contract (grant agreement no.: 249119). The author would like to thank Sabine Hunsicker and Yu Chen for their support in creating the WMT12 translations, and is indebted to Hervé Saint-Amand for providing help with the automated metrics scores. Also, we are grateful to the anonymous reviewers for their valuable feedback and comments.

References

Vera Aleksic and Gregor Thurmair. 2011. Personal translator at wmt2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 303–308, Edinburgh, Scotland, July. Association for Computational Linguistics.

Juan A. Alonso and Gregor Thurmair. 2003. The Comprendium Translator system. In *Proceedings of the Ninth Machine Translation Summit*, New Orleans, USA.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Christian Federmann and Sabine Hunsicker. 2011. Stochastic parse tree selection for an existing rbmt system. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 351–357, Edinburgh, Scotland, July. Association for Computational Linguistics.

Christian Federmann, Andreas Eisele, Yu Chen, Sabine Hunsicker, Jia Xu, and Hans Uszkoreit. 2010. Further experiments with shallow hybrid mt systems. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 77–81, Uppsala, Sweden, July. Association for Computational Linguistics.

Christian Federmann. 2010. Appraise: An open-source toolkit for manual phrase-based evaluation of translations. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL Demo and Poster Sessions*, pages 177–180, Jun.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit 2005*.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece, March. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. IBM Research Report RC22176(W0109-022), IBM.

F. Rosenblatt. 1958. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Toby Segaran. 2007. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly, Beijing.

V. N. Vapnik. 1995. *The nature of statistical learning theory*. Springer, New York.