Local Search for Creating Labelled Development Data in MT System Combination

Hannes Karlbom and Aaron Smith

Convertus AB & Uppsala University
hannes.karlbom@gmail.com, aaron.smith@convertus.se

Abstract

When developing a system that attempts to choose the best output from several machine translation engines for each input sentence, development data labelled with the correct answer is required. This data consists of input sentences, candidate outputs from each of the machine translation engines and labels to specify which of the outputs is best for each sentence. Previous research used an n-gram heuristic to compare each candidate output for each sentence to a reference translation in order to determine which of them contributes to the highest overall BLEU score. Here, we present an alternative method based on local search to choose the best candidate output for each sentence in the development data, and show that it leads to a higher BLEU on the labelled data itself and, more importantly, a higher BLEU on unseen test data for a combination of systems using a classifier trained on this new data.

1. Introduction

The objective of this work was to implement a machine translation (MT) system combining one rule-based (RBMT) and one statistical (SMT) system for the education domain for Swedish to English. The individual engines have both been developed at Convertus, a machine translation company based in Uppsala, Sweden. The rule-based system, known as the Convertus Syllabus Translator, has been deployed at leading Swedish universities for several years. This has led to the creation of a large corpus of translated course syllabuses through the post-editing of output from the Syllabus Translator, which was used in turn to train the SMT system. The two systems are known to have roughly equal performance on unseen course syllabus data. The hope was that a combined system might produce a better result than either of the individual systems.

2. Previous research

There is a wealth of research into merging MT engines, where consolidation can take place at various different stages in the translation process. Most often, *hybrid* systems refer to those where different elements within a single sentence can come from different engines (Costa-jussà and Fonollosa (2015) provide a review of these), whereas *combination* refers to translating each sentence separately and trying to choose the best candidate output (Zwarts and Dras, 2008; Cer et al., 2013). This work falls into the latter category; we treat the individual systems as black boxes, where the only information available is the source text, two translations (RBMT and SMT) and log files with information from the two systems.

A common method employed for system combination is to train a classifier to select which system will perform best for any new input sentence. The classifier can be trained on features from the input sentence and each of the candidate output translations. A crucial component in the training of the classifier is labelled data where the best output—that which contributes to the highest BLEU score for the whole data set—is known for each sentence. BLEU (Papineni et al., 2002), a measure of modified n-gram precision calculated by comparing a candidate translation to one or more reference translations, is the most widely used metric for evaluation of machine translation quality. It is designed to work at the document level, and is not appropriate for direct calculation at the sentence level. Meanwhile, for k sentences in the development data, there are 2^k possible combinations (in the case of two systems), and expounding each to calculate document-level BLEU is clearly not feasible.

To get around this problem, previous researchers such as Zwarts and Dras (2008) have used an n-gram heuristic to choose between potential outputs. The idea behind this heuristic is that matching n-grams between the candidate output and the reference translation contribute exponentially more to the overall BLEU for higher n. Candidate outputs are therefore chosen by first comparing 4-grams, then 3-grams and so on. As soon as one candidate matches more reference n-grams for a given value of $n \in (4, 3, 2, 1)$, it is declared the winner.

This heuristic will not necessarily lead to the best overall BLEU score for the development data used to train the classifier. It has been shown previously that idiosyncrasies related to BLEU itself mean that combining smaller chunks with optimal BLEU scores does not always lead to the best overall BLEU score (Chiang et al., 2008). Moreover, the BLEU score includes a length penalty which is not considered in the sentence-level heuristic. There may also be a tendency to produce many ties, where two outputs are different but cannot be distinguished by the heuristic, and a default must be chosen. To overcome some of these issues and produce more accurate labelled data for training the classifier, we propose a new method based on local search.

3. Local search

Local search algorithms such as simulated annealing are a common approach to intractable optimization problems (Hoos and Stützle, 1999). Local search has also been previously used in decoding for machine translation (Hardmeier

et al., 2012), and in investigating properties of the BLEU score (Smith et al., 2016).

Our idea here is as follows:

1. Start with a complete translation of the development data, comprised of output from one of the two MT systems for each sentence.
2. Calculate the BLEU score of this translation by comparison to reference translations.
3. Pick one or more sentences at random and switch their translation to the output of the other MT system.
4. Re-calculate BLEU for the whole translation: if it has gone up, the switch is accepted; otherwise, it is accepted with a certain probability.
5. Return to step 3 and keep looping until a certain stopping condition has been reached.

In step 4 we allow even changes that decrease BLEU to be accepted on occasion; this helps prevent the algorithm getting stuck in local minima. By lowering the probability of acceptance over time, this simulated annealing algorithm mimics the cooling that takes place during annealing of a physical material (Kirkpatrick et al., 1983).

There are several key issues that must be resolved before putting the above algorithm into practice: What is the best initial condition (choose output for all sentences from the SMT system or RBMT system or a combination thereof)? Which and how many sentences should be switched during step 3? What is the optimum stopping condition? These and other issues are explored in detail in Karlbom (2016).

4. Method

Our experiments consisted of two parts: firstly creating labelled data using the simulated annealing method described in Section 3; secondly training a classifier on this data to attempt to pick the best system for unseen test sentences. The data set was collected by web scraping various syllabuses from Stockholm University (not a Convertus customer) that could be found in both Swedish and English, before aligning sentences. In total, 731 parallel sentences were extracted: a small data set, largely because of time constraints. Due to the small size of the data set, k-fold cross-validation was employed for evaluation of the classifier. The Swedish texts were processed with Convertus' standard pipeline (cleaning, tokenisation, lowercasing etc.), before being sent as input to both the RBMT and SMT systems to produce two candidate translations for each sentence.

Our classifier was based on a support vector machine (SVM), with features such as sentence length, average token length, and average number of out-of-vocabulary words for the SMT system. These basic features were augmented with frequency statistics for part-of-speech tags for the input and two candidate output sentences, as well as additional features from parsing all three versions of the sentence in Universal Dependencies format (Nivre et al., 2016). Detailed information about the features selected can be found in Karlbom (2016).

System	BLEU
SMT	32.2
RBMT	32.3
n-gram heuristic	35.5
Local search	36.1

Table 1: *BLEU scores on our development data set for the individual SMT and RBMT systems, as well as the labelled data, as created by applying an n-gram heuristic or by local search.*

System	BLEU
SMT	31.2
RBMT	31.2
n-gram heuristic	32.1
Local search	33.1

Table 2: *BLEU scores on our test data set for the individual SMT and RBMT systems, as well as the combined system where the labelled data on which the classifier is trained is taken from n-gram heuristic or local search.*

5. Results

Results for the creation of labelled data are found in Table 1. The BLEU scores of the SMT and RBMT systems individually on our development data set were 32.2 and 32.3, respectively. Using the n-gram heuristic described in Section 2 to select the best output for each sentence, a BLEU score of 35.5 was obtained. Our proposed local search method, described in Section 3, meanwhile, achieved a higher BLEU of 36.1. Note that these scores are not the results of the classifier on unseen data, rather they are indications of the maximum BLEU score possible if the classifier were perfect, and are used to create the labelled development data on which the classifier is then trained.

The increase of 0.6 BLEU points on the development data is a promising sign for the final results of the combined SMT system, but without testing we do not know for sure that it will lead to an improvement on unseen test data. There is always a risk of overfitting to the development data: essentially picking candidate translations that happen to lead to a higher BLEU on the development data (perhaps simply because of their length), but do not provide better data for the training of the classifier for optimal performance on unseen data.

For this reason, we trained our classifier using two different sets of labelled data: one with the n-gram heuristic and one with our local search algorithm. The results are presented in Table 2. With the n-gram heuristic, a BLEU of 32.1 was obtained on our test data, an increase of 0.9 over the BLEU of 31.2 achieved by the SMT system on its own. Using the local search algorithm to create the labelled data meanwhile, the final BLEU score was 33.1, a much larger improvement of 1.9 BLEU points.

6. Discussion

It is interesting that there is such a big difference between using the labels created from the n-gram heuristic and those created with simulated annealing (SA) on the final combined system: a whole BLEU point, greater even than the improvement on the development data of 0.6 BLEU points.

It seems that the SA algorithm is able to find more complex patterns of translated sentences that together produce a higher BLEU and moreover are easier to find an underlying function for. According to the n-gram heuristic, almost half of the sentences in our data set are considered equal and the best system is therefore picked at random; this is in contrast to SA where every sentence has its place specifically chosen to maximise BLEU.

Note that the methods presented here could easily be applied to other MT metrics: future work will focus on verifying the results with a larger data set and further metrics. We will also look to further improve the performance of our classifier by adding more features from dependency parsing: the choice of Universal Dependencies allows the possibility to make direct comparisons between the input sentence and candidate translations.

References

D. Cer, C. Manning, and D. Jurafsky. 2013. Positive Diversity Tuning for Machine Translation System Combination. In *Proceedings of the Eight Workshop on Statistical Machine Translation*, pages 320–328.

D. Chiang, S. DeNeefe, Y. S. Chan, and H. T. Ng. 2008. Decomposability of Translation Metrics for Improved Evaluation and Efficient Algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619.

M. R. Costa-jussà and J. A. R. Fonollosa. 2015. Latest trends in hybrid machine translation and its applications. *Computer Speech and Language*, 32:3–10.

C. Hardmeier, J. Nivre, and J. Tiedemann. 2012. Document-Wide Decoding for Phrase-Based Statistical Machine Translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190.

H. H. Hoos and T. Stützle. 1999. *Stochastic Local Search: Foundations & Applications*.

H. Karlbom. 2016. Hybrid Machine Translation. Bachelor's Thesis, Uppsala University.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science*, 220(4598):671–680.

J. Nivre, M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

A. Smith, C. Hardmeier, and J. Tiedemann. 2016. Climbing Mount BLEU: The Strange World of Reachable High-BLEU Translations. *Baltic Journal of Modern Computing*, 4(2):269–281.

S. Zwarts and M. Dras. 2008. Choosing the Right Translation: A Syntactically Informed Classification Approach. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1153–1160.