Choosing the Right Translation: A Syntactically Informed Classification Approach

Simon Zwarts
Centre for Language Technology
Macquarie University
Sydney, Australia
szwarts@ics.mq.edu.au

Mark Dras
Centre for Language Technology
Macquarie University
Sydney, Australia
madras@ics.mq.edu.au

Abstract

One style of Multi-Engine Machine Translation architecture involves choosing the best of a set of outputs from different systems. Choosing the best translation from an arbitrary set, even in the presence of human references, is a difficult problem; it may prove better to look at mechanisms for making such choices in more restricted contexts.

In this paper we take a classification-based approach to choosing between candidates from syntactically informed translations. The idea is that using multiple parsers as part of a classifier could help detect syntactic problems in this context that lead to bad translations; these problems could be detected on either the source side—perhaps sentences with difficult or incorrect parses could lead to bad translations—or on the target side—perhaps the output quality could be measured in a more syntactically informed way, looking for syntactic abnormalities.

We show that there is no evidence that the source side information is useful. However, a target-side classifier, when used to identify particularly bad translation candidates, can lead to significant improvements in Bleu score. Improvements are even greater when combined with existing language and alignment model approaches.

1 Introduction

It is fairly safe to say that whenever there are multiple approaches to solving a problem in Artificial Intelligence, the idea of trying to find a better solution by combining those approaches has been proposed: blackboard architectures, ensemble methods for machine learning, and so on.

In Machine Translation (MT), there is a long tradition of combining multiple machine translations, as through a Multi-Engine MT (MEMT) architecture; the origins of this are generally credited to Frederking and Nirenburg (1994). One way of dividing up such systems is into those that take the whole output from multiple systems and judge between them to select the best candidate, and those that combine elements of the outputs to construct a best candidate.

Deciding between whole sentence level outputs looks like a classical classification problem. Of course, deciding between MT outputs in the general case is a problem that currently has no good solution, and is unlikely to in the near future: Bleu (and similar metrics) require one or more reference texts to distinguish between candidate outputs with the level of accuracy that they achieve, and even then they are open to substantial criticism (Callison-Burch et al., 2006). However, there are reasons to think that there is some promise in considering this as a classification problem. Corston-Oliver et al. (2001) build a classifier to distinguish between human and machine translations with an 80% accuracy. Several other later systems have some success in distinguishing between MT outputs using language models, alignment models, and voting schemes. In addition, while the problem of deciding between arbitrary MT outputs is difficult, it may

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 1153–1160
Manchester, August 2008

be feasible in specific cases. The classifier constructed by Corston-Oliver et al. (2001) takes advantage of characteristic mistakes found in the output of the particular MT system used.

In general, we are interested in MT where syntax is involved. The first part of the main idea of this paper is that there are two ways in which problematic translations might be detected. One is on the source side: perhaps sentences with difficult or incorrect parses could lead to bad use of syntax and hence bad translations, and this could be detected by a classifier. The other is that on the target side, perhaps the output quality could be measured in a more syntactically informed way, looking for syntactic abnormalities.

As to the particular system, in this paper we look at a specific type of MT, the output of systems that use syntactic reordering as preprocessing (Collins et al., 2005; Wang et al., 2007; Zwarts and Dras, 2007). In these systems, the source language is reordered to mirror the syntax of the target language in certain respects, leading to an improvement in the aggregate quality of the output over the baseline, although it is not always the case that each individual sentence in the reordered version is better. This could then be framed as an MEMT, where the reordered candidate is considered the default one, backing off to the baseline where the reordered one is worse, based on the decision of a classifier. Given the 'unnatural' order of the preprocessed source side, there is reason to expect that bad or unsuccessful reordered translations might be detectable.

The second part of the main idea of the paper is that a classifier could use a combination of multiple parsers, as in Mutton et al. (2007), to indicate problems. In that work, designed to assess fluency of output of generation systems, metrics were developed from various parsers—log probability of most likely parse, number of tree fragments, and so on—that correlated with human judgements, and that could be combined in a classifier to produce a better evaluation metric. We take such an approach as a starting point for developing classifiers to indicate problematic source and target sides within a reordering MT system.

In Section 2 we review some related work. In Section 3 we investigate the potential gain in correctly choosing the better translation candidate in our context. In Section 4 we build a classifier using an approximation to fairly standard language and alignment model features, mostly for use as a comparator, while Sections 5 and 6 present our models based on source and target language sides respectively. Section 7 concludes.

2 Related work

In this section we briefly review some relevant work on deciding between translation candidates in 'sentence-level' MEMT.

Most common is the use of language models, or voting which may be based on some kind of alignment, or a combination. Callison-Burch and Flournoy (2001) use a trigram language model (LM) on MT outputs to decide the best candidate, looking at nine systems across four language directions and domains, and treating them as black boxes; evaluation is by human judges and on a fairly small data set. Akiba et al. (2002) score MT outputs by a combination of a standard LM and an alignment model (here IBM 4), and then use statistical tests to determine rankings of MT system outputs. Eisele (2005) uses a heuristic voting scheme based on n-gram overlap of the different outputs, and adds an LM to make decisions; the LM reportedly achieves further improvement. Rosti et al. (2007) look at sentence-level combinations (as well as word- and phrase-level), using reranking of n-best lists and confidence scores derived from generalised linear models with probabilistic features from n-best lists. Huang and Papineni (2007) propose a hierarchical model for word, phrase and sentence level combination; they use LMs and interestingly find that incorporating rudimentary linguistic information like Part-of-Speech is helpful. Riezler and Maxwell (2006) combine transfer-based and statistical MT; they back off to the SMT translation when the grammar is inadequate, analysing the grammar to determine this.

Other work, like ours, uses a classifier. The goal of Corston-Oliver et al. (2001) is slightly different, in that it aims to distinguish human translations from MT output. The classifier uses syntactic features derived from a manual error analysis, taking advantage of character-

istics specific to their MT system and parser. Nomoto (2003) uses a LM and an IBM-based alignment model, and then constructs separate SVMs for regression based on these, each with a single feature (i.e. the LM value or the alignment model value); the SVM is thus not strictly used as a classifier, but as a regression tool. Nomoto (2004) extends this by deciding on the best LM through a voting scheme. Other related work not in an MEMT context that uses parsers to distinguish better from worse translations are on syntax-based language models (Charniak et al., 2003) and on syntactically informed reranking (Och et al., 2003). Both use only single parsers and work only with candidate translations generated inside an SMT system (either all candidates or n-best).

3 Potential Gain

The type of system we focus on in this paper operates in two stages. First, syntactically based reordering takes place to make the source sentence more similar in structure to the syntax of the target language. This is then passed to a Phrase-based SMT (PSMT) component (Pharaoh (Koehn, 2004) in the cited work). For German to English (Collins et al., 2005) and Dutch to English (Zwarts and Dras, 2007) this reordering involves moving some long-distance dependencies closer together, such as clause-final participles and verb-second auxiliaries. This improves translation quality by compensating for the weakness in PSMT of long-distance word reordering: Collins et al. (2005) report a 1.6 Bleu percentage point improvement, Zwarts and Dras (2007) a 1.0 Bleu percentage point improvement.

However, individual sentences translated from the original non-reordered source sentences are sometimes better than their reordered equivalent; examples are given in both Collins et al. (2005) and Zwarts and Dras (2007). (We refer to these in the rest of the paper as non-reordered translations and reordered translations respectively.) For there to be a point to constructing an MEMT-style system where the reordered translation is the default translation and the non-reordered translation the fallback, it is necessary for the non-reordered version to be better a reasonable proportion of the time, allowing scope for a Bleu improvement across the system.

To determine if this is the case, we construct an approximate oracle to choose the better of each pair of reordered and non-reordered translation sentences. While Bleu is a reasonable choice for evaluating the quality of the overall composite set of translation sentences, it is not suitable for sentence-level decisions. However, in line with Nomoto (2003)'s motivation for developing m-precision as an alternative to Bleu, we make the following observation.

The Bleu score (ignoring brevity) is an harmonic mean between the different n-gram components:

$$exp(\sum_{n=1}^{N} \log p_n)$$

Here p_n is the precision for the different n-gram overlap counts of a candidate sentence with a gold standard sentence. If we want to globally optimise this score for an optimal Bleu document score, we need to pick for each sentence the n-gram counts that contribute most to the overall score. For example, if we have to pick between sentence A and sentence B, where A has 2 unigram counts and 1 bigram count, and B has 2 unigram counts only, A is clearly preferred; however, for sentences C and D, where C has 4 unigram counts and D has 2 unigram counts and 1 bigram count, we do not know which eventually will lead to the global maximum Bleu.

However we observe that because it is an harmonic mean, small values are weighted exponentially heavier, due to the log operator. Our heuristic to achieve the highest score is to have the most extreme possible small values. Since we know that an n-gram is always less frequent than an $(n-1)$-gram we concentrate on the higher n-grams first. The decision process between sentences is therefore to choose the candidate with higher n-gram counts for the maximum value of n, then $n-1$-gram counts, and so on down to unigrams.

Here we will work with the Dutch–English data used by Zwarts and Dras (2007). We use the portions of the Europarl corpus (Koehn, 2003) that were used for training in that work; and Bleu with 1 reference with n-grams up to length 4. We then use our heuristic to select between the reordered and non-reordered

Not-BLEU comparable	
Identical	179,327
Undecidable	119,725
Total	299,052
BLEU comparable	
Non-Reordered better	128,585
Reordered better	163,172
Total	291,757
Overall Total	590,809

Table 1: Comparing translation quality

Learner	Baseline	Accuracy
English → Dutch		
SVM - Polynomial	56.0%	56.6%
SVM - Polynomial	50.0%	51.2%
Maximum Entropy	50.0%	51.0%
Dutch → English		
SVM - Polynomial	50.0%	51.4%

Table 2: Results for internal language decider

translation candidates of Zwarts and Dras (2007) for the language direction Dutch to English. Selecting the reordered translation as default and backing off leads to a 1.1 BLEU percentage point improvement over the 1.0 already mentioned. Results for English to Dutch are similar.

In Table 1 we see the breakdown of the entire corpus we work with. Some sentences are identical, and some are different but with no indication by our heuristic as to which of the two is better. In the cases where we do have an indication we see a sizeable 44% of the non-reordered translations are better.

4 Internal Indicators

Before looking at our syntax-related approaches, it would be useful to have a comparison based on the approaches of previous work. As noted in Section 2, these generally use language models and alignment models, as usual to estimate fluency and fidelity of candidate translations.

Because our two candidate solutions are both ultimately produced by Pharaoh (Koehn, 2004), our quick-and-dirty solution can use Pharaoh's own final translation probabilities, which capture language and alignment model information. We build a classifier

Learner	Baseline	Accuracy
English → Dutch		
SVM - Polynomial	50.0%	50.1%
SVM - Radial	50.0%	49.7%
Maximum Entropy	50.0%	50.2%

Table 3: Results for Source language decider

that attempts to distinguish the better of a pair of reordered and non-reordered translations. Denoting the non-reordered translation T_n, and the reordered T_r, we take as features $\log(P(T_n))$, $\log(P(T_r))$, and $\log(P(T_n))$-$\log(P(T_r))$. In addition, because the sentences do not always have equal length and we do not want to penalise longer sentences, we also have three features describing the perplexity: $e^{\log(P(T_n))/\text{length}(T_n)}$, $e^{\log(P(T_r))/\text{length}(T_r)}$, and the difference between these two. Here *length* is the function returning the length of a sentence in tokens. Our training data we get by partitioning the sentences according to whether reordering is beneficial as measured by our heuristic from Section 3. As machine learners we used SVM-light[1] (Joachims, 1998) and the MaxEnt decider from the Stanford Classifier[2] (Manning and Klein, 2003).

Table 2 shows the results the classifier produces on this data set. While the accuracy rates for the classifiers are all statistically significantly different (at a 95% confidence level) from the baseline (using a standard test of proportions), the results are not promising.

5 Source Language Indicators

5.1 All Data

The finding that almost half of the reordered translations degrade the actual translation quality raises the question of why. Our initial hypothesis is that because we use more linguistic tools, this is likely to introduce new errors. We hypothesise that one of the problems of reordering is either the parser getting it wrong, or the rules getting it wrong because of parse complexity. Our idea for estimating the wrongness of a parse, or the complexity of a parse that might lead to incorrect reordering rule application, is to use 'side-effect' informa-

[1] http://svmlight.joachims.org

[2] http://nlp.stanford.edu/software/classifier.shtml

Top	Correct	Accuracy
10	5	50%
50	23	46%
100	48	48%
200	100	50%
500	240	48%
1000	490	49%

Table 4: Accuracy range for Source Side Extreme Predictions

tion from multiple parsers, in a modification of an idea taken from Mutton et al. (2007).[3] For example, the parser of Collins (1999), in addition to the actual parse, gives a probability for the most likely parse; if this most likely parse is not at all likely, this may be because the parser is having difficulty. The Link Parser (Grinberg et al., 1995) produces dependency-style parses, and gives an unlinked fragment count where a complete parse cannot be made; this unlinked fragment count may be indicative of parse difficulty. For this part, we therefore look only at translations with English as source side and Dutch as target, in order to be able to use multiple parsers on the source side sentences.

Again, we construct a machine learner to predict which is the better of the reordered and non-reordered translations. Our training data is as in Section 4.

As a feature set we use: character and token length of the sentence, probability values as supplied by the Collins parser, and the unlinked fragment count as supplied by the Link Parser. We used machine learners as in Section 4. Both the SVM and the features are similar to Mutton et al. (2007).

The results are calculated on 39k examples, split 30k training, 9k testing. Table 3 shows the results for different learning techniques with different settings. The accuracy scores show selection no different from random: none of the differences are statistically significant. With such poor results, we do not bother to calculate the BLEU effect of using the classifier as a decider here.

[3]Similar work is that of Albrecht and Hwa (2007); however this requires human references unavailable here.

Learner	Baseline	Accuracy
Dutch → English		
SVM - Polynomial	50.0%	52.3%
Maximum Entropy	50.0%	52.9%

Table 5: Results for target language decider

5.2 Thresholding

Because our MEMT uses the non-reordered translations as a back-off, even if the classifier is not accurate over the whole set of sentences, it could still be useful to identify the poorest reordered translations and back off only in those cases. SVM-light gives prediction scores as part of its classification; data points that are firmly within the positive (negative) classification spaces are higher positive (negative) values, while border-line cases have a value very close to 0. Here we interpret these as an estimate of the magnitude of the difference in quality between reordered and non-reordered translations. We calculated the accuracy over the n most extreme predictions for different values of n. The results in Table 4 show that the 'extreme range' does not have a higher accuracy either.

6 Target Language Indicators

6.1 All Data

We now consider our second approach, trying to classify syntactic abnormality of the translations. Inspecting the sentences by hand, we found that there are some sentences with markedly poor grammaticality, even by the standards of MT output. Examples of often reoccurring problems include verb positioning (often still sentence-final), positioning of modals in the sentence, etc. Most are in the realm of problems the reordering rules actually try to target.

Here we use the multiple-parser approach in a way more like that of Mutton et al. (2007), as an estimate of the fluency of the sentence with a focus on syntactic characteristics. As in Section 5, we construct a classifier using multiple parser outputs to distinguish the better of a pair of reordered and non-reordered translations. Similarly, we use as features the most likely parse probability of the Collins parser (Collins, 1999) and unlinked fragment count

Learner	Baseline	Accuracy
SVM - Complete	50.0%	52.3%
SVM - LargeDiff	50.0%	52.9%
SVM - HugeDiff	50.0%	51.2%

Table 6: Varying BLEU training data

from the Link parser (Grinberg et al., 1995). We combine these with the sentences lengths in both character count and token count of the two candidate sentences.

Our translation direction in this section, Dutch to English, is the opposite of Section 5, for the same reason that we want to use multiple parsers on the target side. The reordering on the Dutch language is done on the results of the Alpino (Bouma et al., 2000) parser. The rules for reordering are found in Zwarts and Dras (2006). Our training data is again as in Section 4.

Table 5 shows the accuracy, calculated on a 38k examples, split 30k training, 8k testing. The accuracy again is close to baseline performance, although it is clearly better than our LM and alignment classifier of Section 4. Here all the improvements are statistically significant on a 95% confidence level. This is surprising as Mutton et al. (2007) on a somewhat similar task was much more successful. Their performance is expressed as a correlation with human judgement rather than accuracy, but compared to our performance where the improvement in accuracy is only a couple of times the standard error, their approach performed much better. A possible explanation could be that the data we work on has much subtler differences than their work. We know both translations are ultimately generated from the same input, which makes our both candidates very close.

6.2 Varying Training Data

In particular in (Mutton et al., 2007) the training data used human sentences as positive exemplars and very simple bigram-generated sentences as negative ones, so that there was a big difference in quality between them. So perhaps there are too many borderline cases in the training data here.

Therefore we retrained the classifier of Section 6.1, selecting only those sentence pairs where the difference was more distinct. For the LargeDiff set the difference was at least 4 or more unigrams or 3 or more bigrams; for the HugeDiff set the difference was at least 6 or more unigrams or 5 or more bigrams.

Table 6 shows the results; all accuracy scores are better than the baseline with 95% confidence. For LargeDiff, there is an improvement over using the complete data set. Surprisingly, for the HugeDiff training data the gain is not only gone, but this decider performs statistically significantly worse than using all the data.

We therefore conclude that the nature of mistakes made when using reordering as a preprocessing step is of a very subtle kind. Very big mistakes are made as part of translation process completely independent of reordering, while the improvement due to reordering is only where subtly a small set of words, compared to the reference, has been changed for the better. The training size however is only reduced to three quarters of the complete training size. It is therefore very unlikely this sudden drop in performance is due to data sparsity.

Top	Correct	Accuracy
10	9	90%
20	19	95%
50	40	80%
100	79	79%
200	145	72.5%
500	300	66.6%
1000	538	53.8%

Table 7: Accuracy of Prediction in the extreme range

6.3 Thresholding

As in Section 5.2, we look at the cases where our SVM gives a higher prediction score that indicates a greater difference in quality of the non-reordered translation over the reordered one. Here we use as training data the LargeDiff set from Section 6.2.

Results are in Table 7, which unlike the thresholded results of Section 5.2 are quite promising. There is a clear pattern here, with very high accuracy scores in the top range, slowly dropping to around overall performance

System	Bleu
Baseline	0.208
Reordered	0.221
SVM-pick	0.238

Table 8: Bleu results for the different selections

Features	Accuracy
SVM - all	52.3%
SVM - length only	49.8%
SVM - length and Link	50.5%
SVM - length and Collins	50.1%

Table 9: Contribution of Parsers

after 1000 samples. This 1000 mark is out of 3461 negative samples in the test set range, roughly marking the first third mark before accuracy scores have reached average performance.

Predictions with an extreme score on the other side of the scale hardly show an improvement. Because this subset of sentences shows a higher accuracy, it is worthwhile to calculate Bleu scores over the sentences in the test set belonging to the top 500 SVM-predictions positive (reordered translation is better) and the 500 SVM predictions negative (non-reordered is better). Table 8 shows the improvement of Bleu scores.[4]

The first interesting thing which can be seen in the table is that this subset of sentences already has higher improvement than is seen in the whole data set simply by choosing the reordered only, because the SVM is already used to pick the most discriminating sentences. We note that on this subset of sentences our technique of picking the right sentence actually scores an improvement equal to the use of reordering by itself.

6.4 Parser Contribution

In Table 9 we show the effects of individual parsers, taking as the starting point the SVM of Table 5. Clearly, combining parsers leads to a much better decider.

Learner	Baseline	Accuracy
Dutch → English		
SVM Polynomial	50.0%	60.5%

Table 10: Combining internal features with target side features

Top	Reordering	Non-reordered
10	9 90%	10 100%
20	18 90%	18 90%
50	33 66%	43 86%
100	61 61%	77 77%
200	114 57%	148 74%
500	289 58%	383 76%
1000	564 56%	748 75%

Table 11: Accuracy of the Combined model

6.5 Combining Models

As the results of classifying translation outputs using features derived from multiple parsers are promising, we next look at whether it is useful to combine this information with the language and alignment model information from Section 4. Remarkably, as can be seen in Table 10, the combination of these two features has a much greater effect that the two features sets individually. Comparing these scores against 80% accuracy achieved in distinguish MT output from human output in the work of Corston-Oliver et al. (2001), this 60% on a dataset with much more subtle differences is quite promising.

Furthermore Table 11 shows the accuracy ranking of the SVM for the combining model for the extreme SVM-predictions, similar to Tables 4 and 7. The last column of Table 11 matches previous tables, but now we also show an improvement in correct prediction for the reordered cases.

7 Conclusion

In this paper we have looked at a restricted MEMT scenario, where we choose between a syntactic-reordering-as-preprocessing translation candidate, in the style of (Collins et al., 2005), and a baseline PSMT candidate. We have shown that using a classifier built around outputs of multiple parsers, to decide

[4] Baseline here is the same baseline from Zwarts and Dras (2007), which is the parser read-off of the tree.

whether to back off to the baseline candidate, can be successful in selecting the right candidate. There is no indication that classifying information on the source side—looking to see whether sentences with difficult or incorrect parses could lead to bad reorderings and hence bad translations—is useful; however, applying such a classifier to the target side—looking to see whether the output quality could be measured in a syntactically informed way, looking for syntactic abnormalities—is successful in detecting particularly bad translation candidates, and leads to an improvement in BLEU score over the reordered translations equal to the improvement gained by the reordering approach over the baseline. Multiple parsers clearly improve the results over single parsers. The target-side classifier can also be usefully combined with language and alignment model features, improving its accuracy substantially; continuing with such an approach looks like a promising direction. As a further step, the results are sufficiently positive to extend to other sorts of syntactically informed SMT.

References

Akiba, Yasuhrio, Taro Watanabe, and Eiichiro Sumita. 2002. Using Language and Translation Models to Select the Best among Outputs from Multiple MT systems. In *Proc. of Coling*, pages 8–14.

Albrecht, Joshua S. and Rebecca Hwa. 2007. Regression for Sentence-Level MT Evaluation. In *Proc. of ACL*, pages 296–303.

Bouma, Gosse, Gertjan van Noord, and Robert Malouf. 2000. Alpino: Wide Coverage Computational Analysis of Dutch. In *Computational Linguistics in the Netherlands (CLIN)*.

Callison-Burch, Chris and Raymond S. Flournoy. 2001. A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In *Proc. MT Summit*, pages 63–66.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In *Proc. of EACL*, pages 249–256.

Charniak, Eugene, Kevin Knight, and Kenju Yamada. 2003. Syntax-based Language Models for Statistical Machine Translation. In *Proc. of MT Summit*, pages 40–46.

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL*, pages 531–540, Ann Arbor, Michigan, June.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Corston-Oliver, Simon, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proc. of ACL*, pages 148–155.

Eisele, Andreas. 2005. First steps towards multi-engine machine translation. In *Proc. of the ACL Workshop on Building and Using Parallel Texts*, pages 155–158.

Frederking, Robert and Sergei Nirenburg. 1994. Three Heads are Better than One. In *Proc. of the ACL Conference on Applied Natural Language Processing*, pages 95 – 100.

Grinberg, Dennis, John Lafferty, and Daniel Sleator. 1995. A robust parsing algorithm for link grammars. In *Proc. of the International Workshop on Parsing Technologies*.

Huang, Fei and Kishore Papineni. 2007. Hierarchical System Combination for Machine Translation. In *Proc. of EMNLP*, pages 277–286.

Joachims, T. 1998. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, A. Smola, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Koehn, Philipp. 2003. Europarl: A Multilingual Corpus for Evaluation of Machine Translation Philipp Koehn, Draft, Unpublished.

Koehn, Philipp. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *Proc. of AMTA*, pages 115–124.

Manning, Christopher and Dan Klein. 2003. Optimization, Maxent Models, and Conditional Estimation without Magic. *Tutorial at HLT-NAACL 2003 and ACL 2003*.

Mutton, Andrew, Mark Dras, Stephan Wan, and Robert Dale. 2007. Gleu: Automatic evaluation of sentence-level fluency. In *Proc of ACL*, pages 344–351.

Nomoto, Tadashi. 2003. Predictive Models of Performance in Multi-Engine Machine Translation. In *Proc. of MT Summit*, pages 269–276.

Nomoto, Tadashi. 2004. Multi-Engine Machine Translation with Voted Language Model. In *Proc. of ACL*, pages 494–501.

Och, Franz Josef, Daniel Gildea, Sanjeev Khundanpur, Anoop Sarkar, Kenju Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jainand Zhen Jin, and Dragomir Radev. 2003. Final report of Johns Hopkins 2003 summer workshop on syntax for statistial machine translation.

Riezler, Stefan and John Maxwell, III. 2006. Grammatical machine translation. In *Proc of NAACL*, pages 248–255.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and B Bonnie J. Dorr. 2007. Combining Outputs from Multiple Machine Translation Systems" in Human Language. In *Proc. of NAACL*, pages 228–235.

Wang, Chao, Michael Collins, and Philipp Koehn. 2007. Chinese Syntactic Reordering for Statistical Machine Translation. In *Proc of EMNLP*, pages 737–745.

Zwarts, Simon and Mark Dras. 2006. This Phrase-Based SMT System is Out of Order: Generalised Word Reordering in Machine Translation. In *Proc. of the Australasian Language Technology Workshop*, pages 149–156.

Zwarts, Simon and Mark Dras. 2007. Syntax-Based Word Reordering in Phrase-Based Statistical Machine Translation: Why Does it Work? In *Proc. of MT Summit*, pages 559–566.