An Empirical Evaluation of Stop Word Removal in Statistical Machine Translation

Chong Tze Yuang
School of Computer Engineering, Nanyang Technological University, 639798 Singapore
tychong@ntu.edu.sg

Rafael E. Banchs
Institute for Infocomm Research, A*STAR, 138632, Singapore
rembanchs@i2r.a-star.edu.sg

Chng Eng Siong
School of Computer Engineering Nanyang Technological University, 639798 Singapore.
aseschng@ntu.edu.sg

Abstract

In this paper we evaluate the possibility of improving the performance of a statistical machine translation system by relaxing the complexity of the translation task by removing the most frequent and predictable terms from the target language vocabulary. Afterwards, the removed terms are inserted back in the relaxed output by using an *n*-gram based word predictor. Empirically, we have found that when these words are omitted from the text, the perplexity of the text decreases, which may imply the reduction of confusion in the text. We conducted some machine translation experiments to see if this perplexity reduction produced a better translation output. While the word prediction results exhibits 77% accuracy in predicting 40% of the most frequent words in the text, the perplexity reduction did not help to produce better translations.

1 Introduction

It is a characteristic of natural language that a large proportion of running words in a corpus corresponds to a very small fraction of the vocabulary. An analysis of the Brown Corpus has shown that the hundred most frequent words account for 42% of the corpus, while only 0.1% in the vocabulary. On the other hand, words occurring only once account merely 5.7% in the corpus but 58% in the vocabulary (Bell et al. 1990). This phenomenon can be explained in terms of Zipf's Law, which states that the product of word ranks and their frequencies approximates a constant, i.e. word-frequency plot is close to a hyperbolic function, and hence the few top ranked words would account for a great portion of the corpus. Also, it appears that the top ranked words are mainly function words. For instance, the eight most frequent words in the Brown Corpus are *the*, *of*, *and*, *to*, *a*, *in*, *that* and *is* (Bell et al. 1990).

It is a common practice in Information Retrieval (IR) to filter the most frequent words out from processed documents (which are referred to as stop words), as these function words are semantically non-informative and constitute weak indexing terms. By removing this great amount of stop words, not only space and time complexities can be reduced, but document content can be better discriminated by the remaining content words (Fox, 1989; Rijsbergen, 1979; Zou et al., 2006; Dolamic & Savoy 2009).

Inspired by the concept of stop word removal in Information Retrieval, in this work we study the feasibility of stop word removal in Statistical Machine Translation (SMT). Different from Information Retrieval, that ranks or classifies documents; SMT hypothesizes sentences in target language. Therefore, without explicitly removing frequent words from the documents, we proposed to ignore such words in the target language vocabulary, i.e. by replacing those words with a null token. We term this process as "relaxation" and the omitted words as "relaxed words".

Relaxed SMT here refers to a translation task in which target vocabulary words are intentionally omitted from the training dataset for reducing translation complexity. Since the most frequent words are targeted to be relaxed, as a result, there will be vast amount of null tokens in the output text, which later shall be recovered in a post processing stage. The idea of relaxation in SMT is motivated by one of our experimental findings, in which the perplexity measured over a test set decreases when most frequent words are relaxed. For instance, a 15% of perplexity reduction is observed when the twenty most frequent words are relaxed in the English EPPS dataset. The reduction of perplexity allows us to conjecture

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 30–37, Avignon, France, April 23 - 27 2012.

about the decrease of confusion in the text, from which a SMT system might be benefited.

After applying relaxed SMT, the resulting null tokens in the translated sentences have to be replaced by the corresponding words from the set of relaxed words. As relaxed words are chosen from the top ranked words, which possess high occurrences in the corpus, their *n*-gram probabilities could be reliably trained to serve for word prediction. Also, these words are mainly function words and, from the human perspective, function words are usually much easier to predict from their neighbor context than content words. Consider for instance the sentence *the house of the president is very nice*. Function words like *the*, *of*, and *is*, are certainly easier to be predicted than content words such as *house*, *president*, and *nice*.

The rest of the paper is organized into four sections. In section 2, we discuss the relaxation strategy implemented for a SMT system, which generates translation outputs that contain null tokens. In section 3, we present the word prediction mechanism used to recover the null tokens occurring in the relaxed translation outputs. In section 4, we present and discuss the experimental results. Finally, in section 5 we present the most relevant conclusion of this work.

2 Relaxation for Machine Translation

In this paper, relaxation refers to the replacement of the most frequent words in text by a null token. In the practice, a set of frequent words is defined and the cardinality of such set is referred to as the relaxation order. For example, lets the relaxation order be two and the two words on the top rank are *the* and *is*. By relaxing the sample sentence previously presented in the introduction, the following relaxed sentence will be obtained: *NULL house of NULL President NULL very beautiful.*

From some of our preliminary experimental results with the EPPS dataset, we did observe that a relaxation order of twenty leaded to a perplexity reduction of about a 15%. To see whether this contributes to improving the translation performance, we trained a translation system by relaxing the top ranked words in the vocabulary of the target language. In this way, there will be a large number of words in the source language that will be translated to a null token. For example: *la* (*the* in Spanish) and *es* (*is* in Spanish) will be both translated to a null token in English.

This relaxation of terms is only applied to the target language vocabulary, and it is conducted after the word alignment process but before the extraction of translation units and the computation of model probabilities. The main objective of this relaxation procedure is twofold: on the one hand, it attempts to reduce the complexity of the translation task by reducing the size of the target vocabulary while affecting a large proportion of the running words in the text; on the other hand, it should also help to reduce model sparseness and improve model probability estimates.

Of course, different from the Information Retrieval case, in which stop words are not used at all along the search process, in the considered machine translation scenario, the removed words need to be recovered after decoding in order to produce an acceptable translation. The relaxed word replacement procedure, which is based on an *n*-gram based predictor, is implemented as a post-processing step and applied to the relaxed machine translation output in order to produce the final translation result.

Our bet here is that the gain in the translation step, which is derived from the relaxation strategy, should be enough to compensate the error rate of the word prediction step, producing, in overall, a significant improvement in translation quality with respect to the non-relaxed baseline procedure.

The next section describes the implemented word prediction model in detail. It constitutes a fundamental element of our proposed relaxed SMT approach.

3 Frequent Word Prediction

Word prediction has been widely studied and used in several different tasks such as, for example, augmented and alternative communication (Wandmacher and Antoine, 2007) and spelling correction (Thiele et al., 2000). In addition to the commonly used word *n*-gram, various language modeling techniques have been applied, such as the semantic model (Luís and Rosa, 2002; Wandmacher and Antoine, 2007) and the class-based model (Thiele et al., 2000; Zohar and Roth, 2000; Ruch et al., 2001).

The role of such a word predictor in our considered problem is to recover the null tokens in the translation output by replacing them with the words that best fit the sentence. This task is essentially a classification problem, in which the most suitable relaxed word for recovering a given null token must be selected. In other words, $w_i = \max_{v_i \in R} P_{\text{sentence}}(\dots w_{i-1} v_i w_{i+1} \dots)$, where $P_{\text{sentence}}(\cdot)$ is the probabilistic model, e.g. *n*-

gram, that estimates the likelihood of a sentence when a null token is recovered with word v_i, drawn from the set of relaxed words R. The cardinality $|R|$ is referred to as the relaxation order, e.g. $|R| = 5$ implies that the five most frequent words have been relaxed and are candidates to be recovered.

Notice that the word prediction problem in this task is quite different from other works in the literature. This is basically because the relaxed words to be predicted in this case are mainly function words. Firstly, it may not be effective to predict a function word semantically. For example, we are more certain in predicting *equity* than *for* given the occurrence of *share* in the sentence. Secondly, although class-based modeling is commonly used for prediction, its original intention is to tackle the sparseness problem, whereas our task focuses only on the most frequent words.

In this preliminary work, our predicting mechanism is based on an *n*-gram model. It predicts the word that yields the maximum a posteriori probability, conditioned on its predecessors. For the case of the trigram model, it can be expressed as follows:

$$w_i = \max_{v_i \in R} P(v_i | w_{i-2} w_{i-1}) \quad (1)$$

Often, there are cases in which more than one null token occur consecutively. In such cases predicting a null token is conditioned on the previous recovered null tokens. To prevent a prediction error from being propagated, one possibility is to consider the marginal probability (summed over the relaxed word set) over the words that were previously null tokens. For example, if v_{i-1} is a relaxed word, which has been recovered from earlier predictions, then the prediction of v_i should no longer be conditioned by v_{i-1}. This can be computed as follows:

$$w_i = \max_{v_i \in R} \bigcup_{v_{i-1} \in R} P(v_i | w_{i-2} v_{i-1}) = \max_{v_i \in R} \sum_{v_{i-1} \in R} P(v_i | w_{i-2} v_{i-1}) \quad (2)$$

The traditional *n*-gram model, as discussed previously, can be termed as the forward *n*-gram model as it predicts the word ahead. Additionally, we also tested the backward *n*-gram to predict the word behind (i.e. on the left hand side of the target word), which can be formulated as:

$$w_i = \max_{v_i \in R} P(v_i | w_{i+1} w_{i+2}) \quad (3)$$

and the bidirectional *n*-gram to predict the word in middle, which can be formulated as follows:

$$w_i = \max_{v_i \in R} P(v_i | w_{i-1}, w_{i+1}) \quad (4)$$

Notice that the backward *n*-gram model can be estimated from the word counts as:

$$P(w_i | w_{i+1} w_{i+2}) = \frac{C(w_i w_{i+1} w_{i+2})}{C(w_{i+1} w_{i+2})} \quad (5)$$

or, it can be also approximated from the forward *n*-gram model, as follows:

$$P(w_i | w_{i+1} w_{i+2}) = \frac{P(w_{i+2} | w_i w_{i+1}) P(w_{i+1} | w_i) P(w_i)}{P(w_{i+2} | w_{i+1}) P(w_{i+1})} \quad (6)$$

Similarly, the bidirectional *n*-gram model can be estimated from the word counts:

$$P(w_i | w_{i-1} w_{i+1}) = \frac{P(w_{i+1} | w_{i-1} w_i) P(w_i | w_{i-1}) P(w_{i-1})}{\sum_{v_i \in V} P(w_{i+1} | w_{i-1} v_i) P(v_i | w_{i-1}) P(w_{i-1})} \quad (7)$$

or approximated from the forward model:

$$P(w_i | w_{i-1} w_{i+1}) = \frac{P(w_{i+1} | w_{i-1} w_i) P(w_i | w_{i-1}) P(w_{i-1})}{\sum_{v_i \in V} P(w_{i+1} | w_{i-1} v_i) P(v_i | w_{i-1}) P(w_{i-1})} \quad (8)$$

The word prediction results of using the forward, backward, and bidirectional *n*-gram models will be presented and discussed in the experimental section.

The three *n*-gram models discussed so far predict words based on the local word ordering. There are two main drawbacks to this: first, only the neighboring words can be used for prediction as building higher order *n*-gram models is costly; and, second, prediction may easily fail when consecutive null tokens occur, especially when all words conditioning the prediction probability are recovered null tokens. Hence, instead of predicting words by maximizing the local probability, predicting words by maximizing a global score (i.e. a sentence probability in this case), may be a better alternative.

At the sentence level, the word predictor considers all possible relaxed word permutations and searchs for the one that yields the maximum a posteriori sentence probability. For the trigram model, a relaxed word that maximizes the sentence probability can be predicted as follows:

$$w_i = \max_{v_i \in R} \prod_{i=1}^{N} P(v_i | w_{i-2} w_{i-1}) \quad (9)$$

where, N is the number of words in the sentence.

Although the forward, backward, and interpolated models have been shown to be applicable for local word prediction, they make no difference at sentence level predictions as they produce identical sentence probabilities. It is not hard to prove the following identity:

$$\prod_{i=1}^{N} P(w_i|w_{i-2}w_{i-1}) = \prod_{i=1}^{N} P(w_i|w_{i+1}w_{i+2}) \quad (10)$$

4 Experimental Results and Discussion

In this section, we first highlight the Zipfian distribution in the corpus and the reduction of perplexity after removing the top ranked words. The *n*-gram probabilities estimated were then used for word prediction, and we report the resulting prediction accuracy at different relaxation orders. The performance of the SMT system with a relaxed vocabulary is presented and discussed in the last subsection of this section.

4.1 Corpus Analysis

The data used in our experiments is taken from the EPPS (Europarl Corpus). We used the version available through the shared task of the 2007's ACL Workshops on Statistical Machine Translation (Burch et al., 2007). The training set comprises 1.2M sentences with 35M words while the development set and test sets contains 2K sentences each.

From the train set, we computed the twenty most frequent words and ranked them accordingly. We found them to be mainly function words. Their counts follow closely a Zipfian distribution (Figure 1) and account for a vast proportion of the text (Figure 2). Indeed, the 40% of the running words is made up by these twenty words.

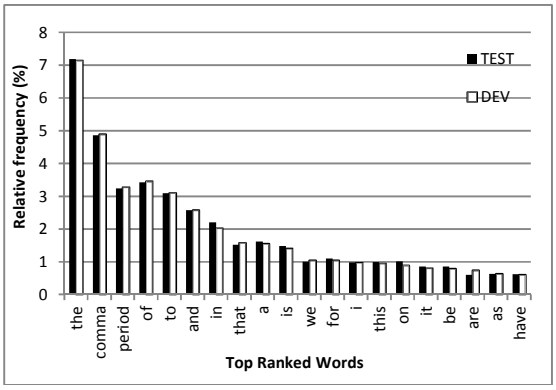

Figure 1. The twenty top ranked words and their relative frequencies

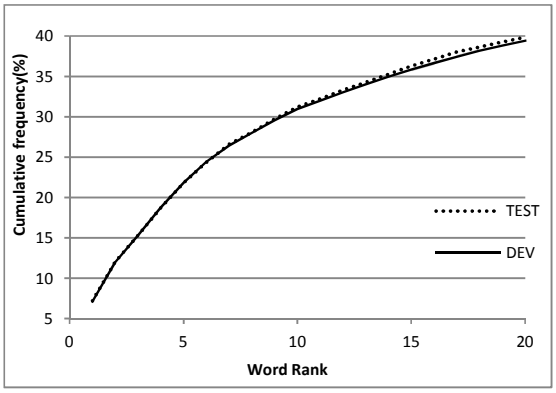

Figure 2. Cumulative relative frequencies of the top ranked words (up to order 20)

We found that when the most frequent words were relaxed from the vocabulary, which means being replaced by a null token, the perplexity (measured with a trigram model) decreased up to 15% for the case of a relaxation order of twenty (Figure 3). This implies that the relaxation causes the text becoming less confusing, which might benefit natural language processing tasks such as machine translation.

Figure 3. Perplexity decreases with the relaxation of the most frequent words

4.2 Word Prediction Accuracy

In order to evaluate the quality of the different prediction strategies, we carried out some experiments for replacing null tokens with relaxed words. For this, frequent words were dropped manually from text (i.e. replaced with null tokens) and were recovered later by using the word predictor. As discussed earlier, a word can be predicted locally, to yield maximum *n*-gram probability, or globally, to yield maximum sen-

tence probability. In a real application, a text may comprise up to 40% of null tokens that must be recovered from the twenty top ranked words.

For *n*-gram level prediction (local), we evaluated word accuracy at different orders of relaxation. More specifically, we tested the forward trigram model, the backward trigram model, the bidirectional model, and the linear interpolation between forward and backward trigram models (with weight 0.5). The accuracy was computed as the percentage of null tokens recovered successfully with respect to the original words. These results are shown in Figure 4. Notice that the accuracy of the relaxation of order one is 100%, so it has not been depicted in the plots.

Notice from the figure how the forward and backward models performed very alike throughout the different relaxation orders. This can be explained in terms of their similar perplexities (both models exhibit a perplexity of 58). Better accuracy was obtained by the interpolated model, which demonstrates the advantage of incorporating the left and right contexts in the prediction.

Different from the interpolated model, which simply adds probabilities from the two models, the bidirectional model estimates the probability from the left and right contexts simultaneously during the training phase; thus it produces a better result. However, due to its heavy computational cost (Equation 8), it is infeasible to apply it at orders higher than five.

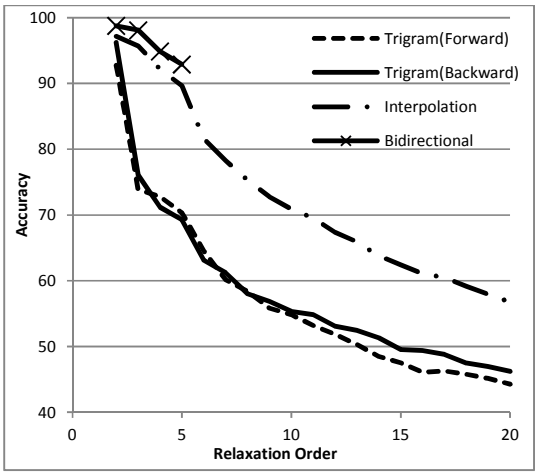

Figure 4. Accuracy of word prediction at *n*-gram level. Models incorporating left and right context yield about a 20% improvement over one-sided models.

Better accuracy has been obtained for sentence-level prediction by using a bigram model and a trigram model. These results are shown in Figure 5. From the cumulative frequency showed in Figure 2, we could see that 40% of the words in text could now be predicted with an accuracy of about 77%.

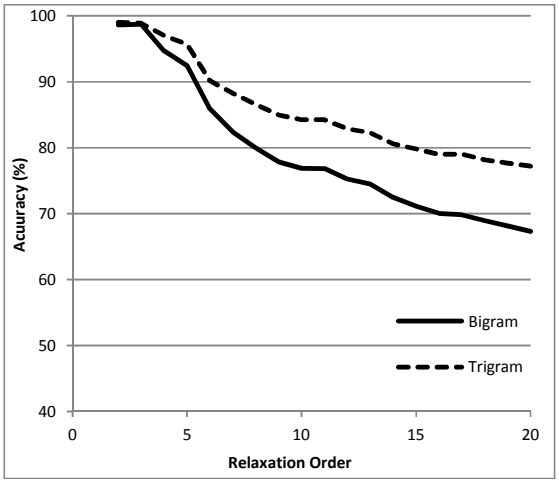

Figure 5. Accuracy of word prediction at the sentence level.

For predicting words by maximizing the sentence probability, two methods have been tried: first, a brute force method that attempts all possible relaxed word permutations for the occurring null tokens within a sentence and finds the one producing the largest probability estimate; and, second, applying Viterbi decoding over a word lattice, which is built from the sentence by replacing the arcs of null tokens with a parallel network of all relaxed words.

All the arcs in the word lattice have to be weighted with *n*-gram probabilities in order to search for the best route. In the case of the trigram model, we expand the lattice with the aid of the SRILM toolkit (Stolcke 2002). Both methods yield almost identical prediction accuracy. However, we discarded the brute force approach for the later experiments because of its heavy computational burden.

Figure 4 and 5 have been plotted with the same scale on the vertical axis for easing their visual comparison. The global prediction strategy, which optimizes the overall sentence perplexity, is much more useful for prediction as compared to the local predictions. Furthermore, as seen from Figure 5, the global prediction has better resistance against higher order relaxations.

We also observed that the local bidirectional model performed closely to the global bigram model, up to relaxation order five, for which the

computation of the bidirectional model is still feasible. In Figure 6 we present a scaled version of the plots to focus on the lower orders for comparison purposes. Although the global bigram prediction makes use of all words in the sentence in order to yield the prediction, locally, a given word is only covered by two consecutive bigrams. Thus, the prediction of a word does not depend on the second word before or the second word after. In other words, we could see the bidirectional model as a global model that is applied to a "short segment" (in this case, a three word segment). The only difference here is that the local bidirectional model estimates the probabilities from the corpus and keeps all seen "short segment" probabilities in the language model (in our case, it is derived from forward bigram), while the global bigram model optimizes the probabilities by searching for the best two consecutive bigrams. The global prediction might only show its advantage when predicting two or more consecutive null tokens.

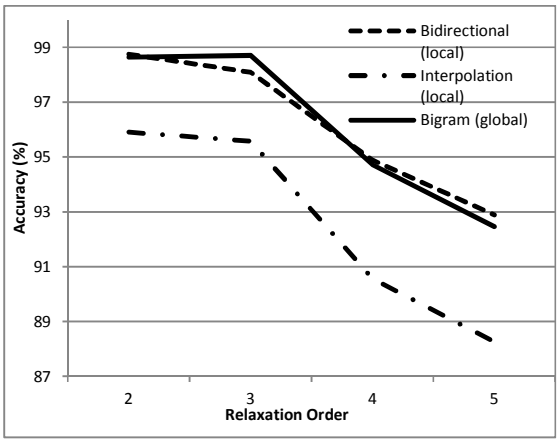

Figure 6. Comparison among local bidirectional, local interpolation, and global bigram models.

Hence, we believe that, if the bidirectional bigram model is computed from counts and stored, it could perform as good as global bigram model at much faster speed (as it involves only querying the language model). Similarly, a local bidirectional trigram model (actually a 5-gram) may be comparable to a global trigram model.

Deriving bidirectional n-gram probabilities from a forward model is computationally expensive. In the worst case scenario, where both companion words are relaxed words, the computation complexity is in the order of $O(|V||R|^3)$, where $|V|$ is the vocabulary size and $|R|$ is the number of relaxed words in V. Building a bidirectional bigram/trigram model from scratch is worth to be considered. As all known language model toolkits do not offer this function (even the backward n-gram model is built by first reversing the sentences manually), the discounting/smoothing of the trigram has to be derived. The methods of Good-Turing, Kneser Ney, Absolute discounting, etc. (Chen and Goodman, 1998) can be imitated.

4.3 Translation Performance

As frequent words have been ignored in the target language vocabulary of the machine translation system, the translation outputs will contain a great number of null tokens. The amount of null tokens should approximate the cumulative frequencies shown in Figure 2.

In this experiment, a word predictor was used to recover all null tokens in the translation outputs, and the recovered translations were evaluated with the BLEU metric. All BLEU scores have been computed up to trigram precision.

The word predictor used was the global trigram model, which was the best performing system in word prediction experiments previously described. In this case, the predictor was used to recover the null tokens in the translation outputs. In order to apply the prediction mechanism as a post-processing step, a word lattice was built from each translation output, for which the null word arcs were expanded with the words of the relaxed set. Finally, the lattice was decoded to produce the best word list as the complete translation output.

To evaluate whether a SMT system benefits from the relaxation strategy, we set up a baseline system in which the relaxed words in the translation output were replaced manually with null tokens. After that, we used the same word predictor as in the relaxed SMT case (global trigram predictor) for recovering the null tokens and regenerating the complete sentences. We then compared the translation output of the relaxed SMT system to the baseline system.

The results for the baseline (BL) and the relaxed (RX) systems are shown in Figure 7. We evaluated the translation performance for relaxation orders of five and twenty.

From the results shown in Figure 7, it becomes evident that the translation task is not gaining any advantage from relaxation strategy and did not outperform the baseline translator, neither at low nor at high orders of relaxation.

Notice how the BLEU score of the baseline systems are better than those of the relaxed systems.

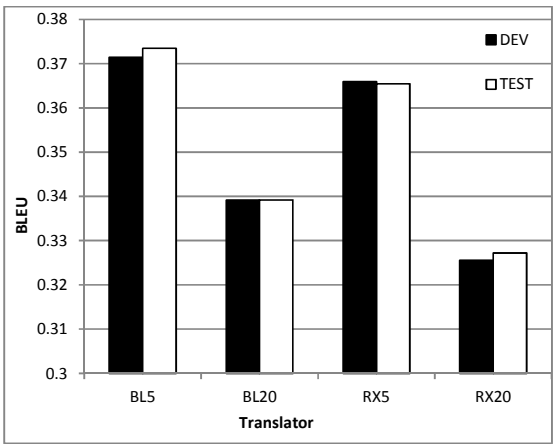

Figure 7. BLEU scores for baseline (BL) and relaxed (RX) translation systems at relaxation orders of five and twenty.

We further analyzed these results by computing BLEU scores for the translation outputs before and after the word prediction step. These results are shown in Figure 8. Notice from Figure 8 that the relaxed translators did not produce any better BLEU score than the corresponding baseline systems, even before word recovery. Although the text after relaxation is less confusing (perplexity decreases about 15% after the twenty most frequent words are relaxed), the resulting perplexity drop was not translated into a BLEU score improvement.

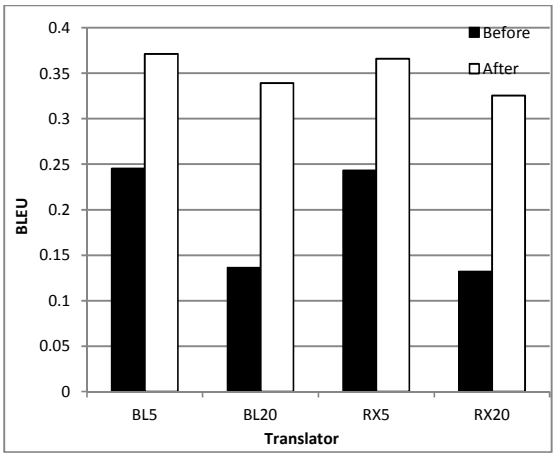

Figure 8. The BLEU scores before and after word prediction

In terms of the word predictions shown in Figure 8, we can see that this post-processing step performed consistently for the relaxed SMT systems, as for the baseline systems (for which the null tokens were inserted manually into sentences). Since the word prediction is based on an *n*-gram model, we may deduce that the relaxed SMT system preserves the syntactic structure of the sentence as the null tokens in the translation output could be recovered as accurate as in the case of the baseline system.

5 Conclusion and Future Work

We have looked into the problem of predicting the most frequently occurring words in a text. The best of the studied word predictors, which is based on an *n*-gram language model and a global search at the sentence level, has achieved 77% of accuracy when predicting 40% words in the text.

We also proposed the idea of relaxed SMT, which consists of replacing top ranked words in the target language vocabulary with a null token. This strategy was originally inspired by the concept of stop word removal in Information Retrieval, and later motivated by the finding that text will become less confusing after relaxation. However, when relaxation is applied to the machine translation system, our results indicate that the relaxed translation task is performing poorer than the conventional non-relaxed system. In other words, the SMT system does not seem to be benefiting from the word relaxation strategy, at least in the case of the specific implementation studied here.

As future work, we will attempt to re-tailor the set of relaxed words by, for instance, imposing some constraints to also include some less frequent function words, which may not be informative to the translation system or, alternatively, excluding some frequent semantically important words from the relaxed set. This remark is based on the observation of the fifty most frequent words in the EPPS dataset, such as *president*, *union*, *commission*, *European*, and *parliament*, which could be harmful when ignored by the translation system but also easy to predict. Hence there is a need to study the effects of different sets of relaxed words on translation performance, as it have already been done for the search problem by researchers in the area of Information Retrieval (Fox, 1990; Ibrahim, 2006).

Acknowledgements

The authors would like to thank their corresponding institutions: the Nanyang Technological Uni-

versity and the Institute for Infocomm Research, for their support regarding the development and publishing of this work.

References

Andreas Stolcke, 2002, SRILM - An Extensible Language Modeling Toolkit, in *Proceedings of ICSLP*, 901-904.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz and Josh Schroeder, 2007, (Meta-)evaluation of machine translation, in *Proceedings of the SMT Workshop*, 136-158

Christopher Fox, 1990, A stop list for general text, *SIGIR Forum*, 24:19-35.

Cornelis Joost van Rijsbergen, 1979, *Information Retrieval*, Butterworth-Heinemann.

Feng Zou, Fu Lee Wang, Xiaotie Deng and Song Han, 2006, Automatic identification of Chinese stop words, *Research on Comp. Science*, 18:151-162.

Frank Thiele, Bernhard Rueber and Dietrich Klakow, 2000, Long range language models for free spelling recognition, in *Proceeding of the 2000 IEEE ICASSP*, 3:1715-1718.

Ibrahim Abu El-Khair, 2006, Effects of stop words elimination for Arabic information retrieval: a comparative study, International Journal of Computing & Information Sciences, 4(3):119-133.

João Luís and Garcia Rosa, 2002, Next word prediction in a connectionist distributed representation system, in *Proceedings of the 2002 IEEE Int. Conference on Systems, Man and Cybernetics*, 6-11.

Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, User interaction with word prediction: the effects of prediction quality, *ACM Transaction on Accessible Computing*, 1(17):1-34.

Ljiljana Dolamic and Jacques Savoy., 2009, When stopword lists make the difference, *Journal of the American Society for Information Science and Technology*, 61(1):1-4.

Patrick Ruch, Robert Baud and Antoine Geissbuhler, 2001, Toward filling the gap between interactive and fully-automatic spelling correction using the linguistic context, in *Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics*, 199-204.

Stanley F. Chen and Joshua Goodman, 1998, An empirical study of smoothing techniques for language modeling, in *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 310-318.

Timothy C. Bell, John G. Cleary and Ian H. Witten., 1990, *Text Compression*, Prentice Hall.

Tonio Wandmacher and Jean-Yves Antoine, 2007, Methods to integrate a language model with semantic information for a word prediction component, in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 506-513.

Yair Even-Zohar and Dan Roth, 2000, A classification approach to word prediction, in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics*, 124-131.