Automatic bilingual phrase extraction from comparable corpora

Ahmet Aker *Yang Feng* *Robert Gaizauskas*
University of Sheffield
{ahmet.aker,y.feng, r.gaizauskas}@sheffield.ac.uk

ABSTRACT
In this work we present an approach for extracting parallel phrases from comparable news articles to improve statistical machine translation. This is particularly useful for under-resourced languages where parallel corpora are not readily available. Our approach consists of a phrase pair generator that automatically generates candidate parallel phrases and a binary SVM classifier that classifies the candidate phrase pairs as parallel or non-parallel. The phrase pair generator is also used to automatically create training and testing data for the SVM classifier from parallel corpora. We evaluate our approach using English-German, English-Greek and English-Latvian language pairs. The performance of our classifier on the test sets is above 80% precision and 97% accuracy for all language pairs. We also perform an SMT evaluation by measuring the impact of phrases extracted from comparable corpora on SMT quality using BLEU. For all language pairs we obtain significantly better results compared to the baselines.

KEYWORDS: SMT for under-resourced languages, phrase extraction from comparable corpora.

1 Introduction

Statistical machine translation (SMT) relies on the availability of rich parallel resources (corpora). However, in many cases, such as for under-resourced languages or in narrow domains, sufficient parallel resources are not readily available. This leads to machine translation systems under-performing relative to those for better resourced languages and domains. To overcome the scarcity of parallel resources the machine translation community has recognized the potential of using comparable corpora as training data. As a result different methods for extracting parallel sentences or smaller text units such as phrases from comparable corpora have been investigated (Munteanu and Marcu, 2006; Sharoff et al., 2006; Kumano et al., 2007; Marcu and Wong, 2002; Barzilay and McKeown, 2001; Kauchak and Barzilay, 2006; Callison-Burch et al., 2006; Nakov, 2008; Zhao et al., 2008; Marton et al., 2009; Skadiņa et al., 2012; Ion, 2012).

A common idea in this related work is the use of some heuristics to pair target and source phrases. By contrast we approach the task of parallel phrase extraction as a classification task and use feature extraction on the training data to train an SVM classifier to distinguish between parallel and non-parallel phrases. Our method is fully automatic and is essentially a "generate and test" approach. In the generate phase, given source and target language sentences S and T, we first generate all possible phrases of a given length for S and for T and then compute all possible phrase pairings consisting of one phrase from S and one phrase from T. In the test phase we use a binary SVM classifier to determine for each generated phrase pair whether it is or is not parallel. The SVM classifier is trained using phrase pairs taken from parallel data word aligned using Giza++ (Och and Ney, 2000, 2003).

We have tested our approach on the English-German, English-Greek and English-Latvian language pairs. Latvian is an under-resourced language, while for Greek and German text resources are more readily available. Considering all three languages allows us to directly compare our method's performance on resource-rich and under-resourced languages. We perform two different tests. First, we evaluate the performance of the classifier on phrases extracted from held-out parallel data using standard measures such as recall, precision and accuracy. Secondly, we test whether the phrases extracted by our method from comparable corpora lead to improved SMT quality, as measured using BLEU (Papineni et al., 2002) .

Hewavitharana and Vogel (2011) also adopt a classification approach for phrase extraction. However, their approach requires manual intervention in data preparation, whereas we perform the preparation of training and testing data fully automatically. In addition, Hewavitharana and Vogel (2011) do not report any SMT performance evaluation of their approach, so it is difficult to estimate how useful their approach is for the actual task it is meant to improve. We test the impact of our extracted phrases on the performance of an SMT system, which allows us to draw conclusions about the likely utility of our approach for SMT in practice.

In Section 2 we present our phrase pair generation method. In Section 3 we describe our classification approach and list the features used within the classifier. Section 4 describes our experimental set-up and results.

2 Phrase Pair Generation

Phrase pairs are generated under two different conditions. During training of the SVM phrase pair classifier, positive and negative instances of aligned phrase pairs are generated from existing parallel resources for the source and target languages. During testing candidate phrase pairs are generated from arbitrary source and target language sentence pairs.

2.1 Training Example Extraction

We use whatever parallel data is available for a language pair to extract training examples for the SVM classifier. To get positive training examples (parallel phrases), we first align the parallel sentence pairs using the Giza++ toolkit (Och and Ney, 2000, 2003) in both directions and then refine the alignments using a "grow-diag-final-and" strategy. Then, we extract all phrases, as defined in the statistical machine translation literature (Koehn et al., 2003; Och and Ney, 2004; Chiang, 2005), and take these phrases as positive examples.

Let S denote a sentence, S_i the i-th word in S and S_i^j the subsequence of words in S from position i to j. Given a word-aligned sentence pair $\langle S, T \rangle$, $\langle S_i^j, T_{i'}^{j'} \rangle$ is a phrase iff:

- S_k is aligned to $T_{k'}$ for some $k \in [i, j]$ and $k' \in [i', j']$
- S_k is not aligned to $T_{k'}$ for all $k \in [i, j]$ and $k' \notin [i', j']$
- S_k is not aligned to $T_{k'}$ for all $k \notin [i, j]$ and $k' \in [i', j']$

To get negative training examples (non-parallel phrases), for each sentence pair, we enumerate all segments on the source side and on the target side, the length of which falls in the range $[minSrcLen..maxSrcLen]$ and $[minTrgLen..maxTrgLen]$, respectively. Then we pair each source segment with each target segment to get all possible training examples. Next, we leave out the positive examples and label the rest as negative examples.

A training example may be discovered many times during extraction process. We do not keep duplicate occurrences but keep all the training examples unique. As the alignment of the parallel corpus inevitably introduces some errors, we do some processing to remove the noise. For instance, a training example may appear both as a positive example and as a negative example, but in our approach, a training example can only have one label, positive or negative. For a training example, assume the number of occurrences as a positive example is N_p and the number of occurrences as a negative example is N_n. We check the following conditions in order:

- If N_p is smaller than a count threshold τ, then we label this example as negative.
- If the ratio N_n/N_p is below a ratio threshold π, then we label it as positive.

2.2 Test Instance Generation

To generate candidate parallel phrase pairs from unseen comparable text pairs we proceed as follows. First we generate all sentence pairs $\langle S, T \rangle$ where S is from the source language text and T is from the target language text. For each such pair we generate all phrase pairs $\langle s, t \rangle$ where s is a word subsequence of S of length i $minSrcLen \leq i \leq maxSrcLen$ and t is a word subsequence of T of length j, $minTrgLen \leq j \leq maxTrgLen$.

3 SVM Classifier

For classifying phrase pairs as parallel or non-parallel we use an SVM classifier. Within the classifier we use the following features as reported in previous work (Munteanu and Marcu, 2005; Hewavitharana and Vogel, 2011):

- ***lengthDifferenceInChar*** is the difference in number of characters in the source and target phrases. We consider duplicates in the phrases when counting the characters.
- ***lengthDifferenceInWords*** is similar to the first feature but use words instead of characters.
- ***sameEnding*** is *1* if source and target phrase have the same ending otherwise *0*.
- ***numberOfWordsInPhrase*** is number of words in the source phrase.
- ***firstWordTranslationScore*** indicates whether the first word in the source phrase is a translation of the first word in the target phrase. If this is the case, the translation probability is returned.
- ***lastWordTranslationScore*** indicates whether the last word in the source phrase is a translation of the last word in the target phrase. If this is the case, the translation probability is returned.
- ***translationCount*** is number of source phrase words which have translations in the target one.
- ***translationRatio*** is ratio of the count of source phrase words which have translations in the target phrase and the number of words in the source language.

- ***isHalfTranslated*** is *1* if at least half of the source phrase words have translations in the target phrase, otherwise *0*.
- ***longestTranslatedUnit*** is count of words within the longest sequence of words which have all translations in the target phrase.
- ***longestNotTranslatedUnit*** similar to the previous one but considers words which do not have translations.
- ***translationPositionDistance*** captures the distance between the source words positions and the position of their maximum likely translations in the target side. E.g. if the first word in the source phrase is the translation of the first word in the target phrase then they have a translation position distance of *0*. For each word in the source phrase we compute its translation position distance, sum all the distances together and return it.

The first three features are independent of which language is taken as source and which as target. The feature *numberOfWordsInPhrase* is computed once for the source and once for the target phrase. The remaining nine features are direction-dependent and are computed in both directions, reversing which language is taken as the source and which as the target. Thus in total we have 21 features. To perform the translation of phrase words we use GIZA++ dictionaries trained on parallel data (see Section 4.2).

3.1 Cognate-based Methods for Translation Purposes

Dictionaries mostly fail to return translation entries for named entities (NEs) or specialized terminology. Because of this we also use cognate-based methods to perform the mapping between source and target words or vice versa. We only apply the cognate-based methods for the *firstWordTranslationScore* and *lastWordTranslationScore* features. For these two features it is easy to compare the first or the last words from both the source and target phrases. The score of the cognate methods becomes the translation score for the features. We adopt several string similarity measures described in Aswani and Gaizauskas (2010): (1) Longest Common Subsequence Ratio, (2) Longest Common Substring, (3) Dice Similarity, (4) Needleman-Wunsch Distance and (5) Levenshtein Distance. Each of these measures returns a score between 0 and 1. We use a weighted linear combination of the scores to compute the final score. We learn the weights using linear regression over training data consisting of pairs of truely and falsely aligned city names available from Wikipedia[1]. For the truely aligned named entities we assign a score of *1* and for the falsely aligned ones a score of *0*. We take the cognate similarity score as the translation score only if it is above 0.7, a threshold which we set experimentally.

The cognate methods assume that the source and target language strings being compared are drawn from the same character set. However, this is not the case for English and Greek. To be able to apply our cognate-based approach to Greek we first map the Greek characters into English characters and apply the cognate metrics on the mapped characters. To learn the mappings we used a list of Greek-English place name variants[2] and the Giza++ tool. The input to Giza++ is a list of aligned NEs (Greek and English) where each NE is split into single characters. The output of the tool is a dictionary with character mappings. We use these mappings to transliterate a Greek word into English characters and use the transliterated version for the cognate comparison. Note, since GIZA++ lists multiple entries as translation variants we always select the one with the highest probability value.

4 Experiments

4.1 Data Sources

Our experiments involve the English-Greek (EN-EL), English-Latvian (EN-LV) and English-German (EN-DE) language pairs. We train a separate classifier for each language pair. Therefore, for each language pair a data set consisting of parallel phrases is needed to train and test the

[1] http://en.wikipedia.org/wiki/Names_of_European_cities_in_different_languages.
[2] http://en.wikipedia.org/wiki/List_of_Greek_place_names

SVM classifier. A second data source needed for our experiments is comparable corpora for the above mentioned language pairs. From these we generate pairs of phrases and judge them for parallelism using the trained classifier. Finally, the phrases judged as parallel by the classifier are used to attempt to improve a baseline SMT system.

4.1.1 Parallel Corpora

We used the JRC-Acquis[3] parallel corpora to prepare the parallel phrases used to train and test the SVM classifier. For each language pair we split the corpus into two parts: a training set and a test set. The test set contains 10K parallel sentences. The training set contains 99K sentences for EN-DE, 423K for EN-EL and 53K sentences for EN-LV.

4.1.2 Comparable Corpora

We used comparable corpora in English-Greek, English-Latvian and English-German language pairs. These corpora were collected from news articles using a light weight approach that only compares titles and date of publication of two articles to judge them for comparability (Aker et al., 2012). The corpora are aligned at the document level and are detailed in Table 1.

language pair	document pairs	EN sentences	target sentences	EN words	target words
EN-DE	66K	623K	533K	14837K	6769K
EN-EL	122K	1600K	313K	27300K	8258K
EN-LV	87K	1122K	285K	18704K	5356K

Table 1: Size of comparable corpora.

4.2 Phrase Extraction for Classifier Training and Testing

On both parallel training and testing data sets (see Section 4.1.1) we separately applied GIZA++ to obtain the word alignment information used in our parallel phrase extraction method (see Section 2.1). Then we ran the training example extraction method on each data set to extract phrase pairs, setting $minSrcLen = minTrgLen = 2$ and $maxSrcLen = maxTrgLen = 7$. To train the classifier we used 20K parallel and 20K non-parallel phrase pairs extracted from the training data. In testing we used 500 parallel and 10K non-parallel phrase pairs extracted from the testing data. Note that the test set contains substantially more non-parallel than parallel data. This is to simulate the real-world scenario where the data from which parallel phrases have to be extracted will necessarily contain more non-parallel entries than parallel ones. It is also important to note that in both the training and testing parallel phrase extraction steps we used GIZA++ dictionaries obtained from the parallel training data which excludes the 10K parallel sentences used in testing. We did this to ensure that feature extraction is testing is performed using a dictionary that has been built by a process which is blind to the test data.

4.3 Phrase Extraction from Comparable Corpora

We used the comparable corpora described in the previous section and for each language and each aligned document pair we extracted phrase pairs as described above in Section 2.2. As when generating training instances we set $minSrcLen = minTrgLen = 2$ and $maxSrcLen = maxTrgLen = 7$. As in the training and testing steps described in previous section, in feature extraction from the phrase pairs generated from the comparable corpora we used the GIZA++

[3]http://langtech.jrc.it/JRC-Acquis.html

dictionary created from parallel sentences in the training data. Table 2 gives details about the phrases extracted from the comparable corpora.

language pair	analysed sentence pairs	analysed phrase pairs	extracted phrase pairs
EN-DE	39659	852327K	248K
EN-EL	33844K	1499169K	125K
EN-LV	30788K	1919128K	106K

Table 2: Phrase pairs extracted from comparable corpora.

We also ran a performance test to evaluate the speed of parallel phrase extraction. We took 1000 comparable document pairs from the EN-DE data and recorded the time it took to process them. We recorded ~ 44 minutes processing time on a single desktop machine with a 2.4GHz processor and 4GB memory. 99% of the processing time was spent on feature extraction and the remaining 1% for phrase pairing and SVM classifier. Note that since the document pairs are independent from each other, multiple processes could be run in parallel on different sets of document pairs which could significantly reduce processing time.

4.4 Results

To test the performance of our approach we performed two different evaluations: classifier evaluation using Information Retrieval (IR) metrics and SMT performance using BLEU.

4.4.1 Classifier Evaluation

In this evaluation we measure the performance of our classifier using precision, recall, F-measure and accuracy (Manning et al., 2008). Note that we use $F_{0.5}$ which puts more emphasis on precision than recall. We sought to optimize SVM classifier performance for our task by finding the SVM-margin distance boundary that maximizes $F_{0.5}$. During training the SVM classifier determines a maximum margin hyperplane between the positive and negative examples. During classification the distance to this boundary is used to classify instances: any instance that has negative distance ($distance < 0$) to the boundary is treated as a negative example, otherwise as positive ($distance >= 0$). We shift the boundary between negative and positive examples to a new value which maximizes the $F_{0.5}$ metric. To do this we determine the maximal negative and maximal positive distance from the classification results, go from the negative value towards the maximal positive value in increments of 0.1 and record the boundary value that leads to the maximum $F_{0.5}$. To learn the new boundary we used held out training data containing 500 parallel and 10K non-parallel phrases. Note that this held out training data is different from the testing data (see Section 4.1.1) but has the same size. Finally, we run the classifier with the new boundary on the testing data. The results are shown in Table 3.

language pair	recall	precision	$F_{0.5}$-measure	accuracy
EN-DE	45	86	73	97
EN-EL	63	81	77	97
EN-LV	59	84	77	97

Table 3: Classifier's performance on phrases extracted from the test data.

From Table 3 we can see that the classifiers for each language pair perform reasonably well on the testing data. They all achieve an accuracy score above 97%, though note that always picking the majority class (non-parallel) gives 95% accuracy given the deliberate skew in the test data. The precision score obtained from each classifier is above 81% showing good performance in identifying correct parallel phrases. In general the recall scores are low, in the neighborhood of

50%. However, given the potentially very large quantities of comparable text pairs available recall is not a primary concern.

To identify the sources of misclassifications we manually checked the EN-DE phrases from the test set which were classified incorrectly. The first source of problems is due to the existence of productive compounds in German and negatively affects recall. For example, the classifier classifies the following parallel phrases as non-parallel. The features we use within the classifier do not capture morphological elements within compound words and thus fail to match, e.g. *tiergesundheitszeugnisse* with *veterinary certificates* or *umweltkriterien* with *ecological criteria*.

(1) *der tiergesundheitszeugnisse für die — veterinary certificates for the*

(2) *zur festlegung überarbeiteter umweltkriterien — establishing revised ecological criteria*

The second problem is due to feature extraction and causes a decrease in precision. The following phrases are non-parallel examples classified by the classifier as parallel. The reason for the misclassification is that while the words in the English phrase can be entirely mapped to those in the German phrase, the phrases are not parallel because they differ either in the number or in the order of constituents.

(3) *parlaments und des rates zur einführung — the council and the*

(4) *die kommission erstattet dem europäischen parlament und — european parliament and of the council*

In (3) all words of the English phrase have translations in the German phrase (both *the*'s are mapped to *des*, *council* is mapped to *rates* or *parlaments* and *and* is mapped to *und*). In (4) we have a similar picture. The words *european parliament* are mapped to *europäischen parlament*, *and* to *und*, *the* to *die* or *dem* and *council* to *kommision*. The problem arises from the fact that in (3) the English word *council* translates into both German *Rat* and *Parlament*. Thus, two German noun phrases (NPs) are covered by one in English, so that the English phrase is not an adequate translation of the German one. In (4), the problem lies in the order of the constituents which results in the two phrases not being parallel. The English phrase contains a coordination of two NPs, while in the German phrase, the coordinating conjunction *und* is at the end of the phrase and serves to link either the entire phrase or the second NP (*dem europäischen parlament*) to a further constituent not extracted as a part of this phrase.

4.4.2 BLEU Evaluation for SMT

In the BLEU evaluation we tested the impact of the phrases extracted from the comparable corpora on improving the performance of the baseline SMT systems. We trained a baseline decoder for each language pair using the entire JRC-Acquis corpus for that language pair, which consists of the training and test data used for our phrase extraction system. We then injected the extracted phrases[4] into the baseline training data and re-trained a new decoder which we call an *extended* decoder. As SMT test data we used 612 parallel sentences manually generated from news articles. The English and the German sentences have both in total 14K words. The Latvian sentences contain around 13K and the Greek ones 15K words. To construct these test sets we used English as the pivot language. We selected from different news articles 612 English sentences and then manually translated them into German, Greek and Latvian. For each language pair a professional translator was hired to perform the translation. Note that these articles are not included in the comparable corpora summarized in Table 1.

From the results shown in Table 4 we can see that all extended decoders significantly outperform the baseline systems[5]. This shows that the phrases extracted from the comparable corpora are

[4]These phrases are extracted with the SVM margin that maximizes the F-measure, see for details Section 4.4.1

[5]Koehn (2004) reports that increase of 1% in BLEU score is a significant improvement.

language pair	baseline BLEU score	extended BLEU score
EN-DE	15.97	**18.05**
EN-EL	28.30	**29.37**
EN-LV	10.24	**12.23**

Table 4: BLEU scores on the SMT testing data.

indeed of usable quality. In the table we also see that the EN-EL BLEU scores are much higher than the others. We think that this is a result of the large size of the EN-EL parallel training data made available by JRC which we used to train the EN-EL decoder. As described in Section 4.1.1 the EN-EL parallel corpus is more than 4 times bigger than the EN-DE corpus and 8 times bigger than the EN-LV parallel corpus. For the language with least training data Latvian, the classifier still significantly outperforms the baseline. This is an encouraging result which shows that although the amount of parallel data is important for SMT performance, our method for phrase extraction from comparable data provides a viable way to significantly improve SMT performance in cases where parallel data is sparse.

Conclusions

In this paper we presented a fully automated approach to extract parallel phrases from comparable corpora using a classifier. The data used to train the classifier is automatically derived from parallel corpora. We measured the performance of our classifier using IR metrics but also performed an SMT evaluation using BLEU. We performed the evaluations EN-DE, EN-EL and EN-LV language pairs. In the IR evaluation we tested our approach on pairs of phrases extracted automatically from parallel corpora. The results of this evaluation show that our approach is precise and accurate in identifying parallel phrases. The SMT evaluation was performed by comparing the translation performance of two decoders on a set of parallel sentences manually collected from news articles. The first decoder is a baseline system trained on the JRC-Acquis parallel corpus. In the second decoder we again use the same parallel corpus but extend it with phrases extracted from a comparable corpus. The results show that the extended decoder performs significantly better than the baselines for all language pairs.

A number of questions remain for further research. First, how much can SMT system performance be improved using this approach? The number of comparable text pairs available is in principle virtually unlimited; however, it is unlikely indefinite improvements to SMT systems can be made using our approach. But how much improvement can be made? Second, the relation between the amount of parallel data initially available, from which dictionaries are derived and parallel phrase pairs are extracted for training the SVM classifier, and the improvement obtainable through use of our approach needs to be better understood. Second, can we bootstrap? – in particular can we use Giza++ to extract a new dictionary from the original parallel data plus the phrase pairs extracted by our classifier during an initial round of phrase extraction and then use this new dictionary to retrain the classifier? Third, more detailed failure analysis needs to be carried out on all of our test languages as well as an analysis of the role of particular features in the classifier. This should provide insights, such as those mentioned in Section 4.4 above, that may allow performance of the classifier to be improved further.

Acknowledgement

The research reported was funded by the ACCURAT and the TaaS projects, European Union Seventh Framework Programme, grant agreements no. 248347 and 296312 respectively. The authors would like to thank Accurat partners and Trevor Cohn for helpful inputs.

References

Aker, A., Kanoulas, E., and Gaizauskas, R. (2012). A light way to collect comparable corpora from the web. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*, pages 21–27.

Aswani, N. and Gaizauskas, R. (2010). English-Hindi transliteration using multiple similarity metrics. In *7th Language Resources and Evaluation Conference (LREC), La Valletta, Malta*.

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57, Morristown, NJ, USA. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006). Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24, Morristown, NJ, USA. Association for Computational Linguistics.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Hewavitharana, S. and Vogel, S. (2011). Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 61–68. Association for Computational Linguistics.

Ion, R. (2012). Pexacc: A parallel sentence mining algorithm from comparable corpora. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*.

Kauchak, D. and Barzilay, R. (2006). Paraphrasing for automatic evaluation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 455–462, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Koehn, P., Och, F., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Kumano, T., Tanaka, H., and Tokunaga, T. (2007). Extracting phrasal alignments from comparable corpora by using joint probability SMT model. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pages 95–103.

Manning, C., Raghavan, P., and Schutze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, volume 10, pages 133–139.

Marton, Y., Callison-Burch, C., and Resnik, P. (2009). Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 381–390. Association for Computational Linguistics.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.

Nakov, P. (2008). Paraphrasing verbs for noun compound interpretation. In *Proc. of the Workshop on Multiword Expressions, LREC-2008*.

Och, F. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Och, F. J. and Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics*, pages 1086–1090, Morristown, NJ, USA. Association for Computational Linguistics.

Och, F. J. O. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sharoff, S., Babych, B., and Hartley, A. (2006). Using comparable corpora to solve problems difficult for human translators. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 739–746, Morristown, NJ, USA. Association for Computational Linguistics.

Skadiņa, I., Aker, A., Mastropavlos, N., Su, F., Tufis, D., Verlic, M., Vasiļjevs, A., Babych, B., Clough, P., Gaizauskas, R., et al. (2012). Collecting and using comparable corpora for statistical machine translation. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*.

Zhao, S., Niu, C., Zhou, M., Liu, T., and Li, S. (2008). Combining multiple resources to improve SMT-based paraphrasing model. In *Proceedings of ACL-08: HLT*, pages 1021–1029, Columbus, Ohio. Association for Computational Linguistics.