Improving MT System Using Extracted Parallel Fragments of Text from Comparable Corpora

Rajdeep Gupta, Santanu Pal, Sivaji Bandyopadhyay
Department of Computer Science & Engineering
Jadavpur University
Kolkata – 700032, India
{rajdeepgupta20, santanu.pal.ju}@gmail.com,
sivaji_cse_ju@yahoo.com

Abstract

In this article, we present an automated approach of extracting English-Bengali parallel fragments of text from comparable corpora created using Wikipedia documents. Our approach exploits the multilingualism of Wikipedia. The most important fact is that this approach does not need any domain specific corpus. We have been able to improve the BLEU score of an existing domain specific English-Bengali machine translation system by 11.14%.

1 Introduction

Recently comparable corpora have got great attention in the field of NLP. Extracting parallel fragments of texts, paraphrases or sentences from comparable corpora are particularly useful for any statistical machine translation system (SMT) (Smith et al. 2010) as the size of the parallel corpus plays major role in any SMT performance. Extracted parallel phrases from comparable corpora are added with the training corpus as additional data that is expected to facilitate better performance of machine translation systems specifically for those language pairs which have limited parallel resources available. In this work, we try to extract English-Bengali parallel fragments of text from comparable corpora. We have developed an aligned corpus of English-Bengali document pairs using Wikipedia. Wikipedia is a huge collection of documents in many different languages. We first collect an English document from Wikipedia and then follow the inter-language link to find the same document in Bengali (obviously, if such a link exists). In this way, we create a small corpus. We assume that such English-Bengali document pairs from Wikipedia are already comparable since they talk about the same entity. Although each English-Bengali document pair talks about the same entity, most of the times they are not exact translation of each other. And as a result, parallel fragments of text are rarely found in these document pairs. The bigger the size of the fragment the less probable it is to find its parallel version in the target side. Nevertheless, there is always chance of getting parallel phrase, tokens or even sentences in comparable documents. The challenge is to find those parallel texts which can be useful in increasing machine translation performance.

In our present work, we have concentrated on finding small fragments of parallel text instead of rigidly looking for parallelism at entire sentential level. Munteanu and Marcu (2006) believed that comparable corpora tend to have parallel data at sub-sentential level. This approach is particularly useful for this type of corpus under consideration, because there is a very little chance of getting exact translation of bigger fragments of text in the target side. Instead, searching for parallel chunks would be more logical. If a sentence in the source side has a parallel sentence in the target side, then all of its chunks need to have their parallel translations in the target side as well.

It is to be noted that, although we have document level alignment in our corpus, it is somehow ad-hoc i.e. the documents in the corpus do not belong to any particular domain. Even with such a corpus we have been able to improve the performance of an existing machine translation system built on tourism domain. This also signifies our contribution towards domain adaptation of machine translation systems.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the preparation of the comparable corpus. The system architecture is described in section 4. Section 5 describes the experiments we

Proceedings of the 6th Workshop on Building and Using Comparable Corpora, pages 69–76,
Sofia, Bulgaria, August 8, 2013. ©2013 Association for Computational Linguistics

conducted and presents the results. Finally the conclusion is drawn in section 6.

2 Related Work

There has been a growing interest in approaches focused on extracting word translations from comparable corpora (Fung and McKeown, 1997; Fung and Yee, 1998; Rapp, 1999; Chiao and Zweigenbaum, 2002; Dejean et al., 2002; Kaji, 2005; Gamallo, 2007; Saralegui et al., 2008). Most of the strategies follow a standard method based on context similarity. The idea behind this method is as follows: A target word t is the translation of a source word s if the words with which t co-occurs are translations of words with which s co-occurs. The basis of the method is to find the target words that have the most similar distributions with a given source word. The starting point of this method is a list of bilingual expressions that are used to build the context vectors of all words in both languages. This list is usually provided by an external bilingual dictionary. In Gamallo (2007), however, the starting list is provided by bilingual correlations which are previously extracted from a parallel corpus. In Dejean (2002), the method relies on a multilingual thesaurus instead of an external bilingual dictionary. In all cases, the starting list contains the “seed expressions” required to build context vectors of the words in both languages. The works based on this standard approach mainly differ in the coefficients used to measure the context vector similarity.

Otero et al. (2010) showed how Wikipedia could be used as a source of comparable corpora in different language pairs. They downloaded the entire Wikipedia for any two language pair and transformed it into a new collection: CorpusPedia. However, in our work we have showed that only a small ad-hoc corpus containing Wikipedia articles could be proved to be beneficial for existing MT systems.

3 Tools and Resources Used

A sentence-aligned English-Bengali parallel corpus containing 22,242 parallel sentences from a travel and tourism domain was used in the preparation of the baseline system. The corpus was obtained from the consortium-mode project “Development of English to Indian Languages Machine Translation (EILMT) System”. The Stanford Parser and the CRF chunker were used for identifying individual chunks in the source side of the parallel corpus. The sentences on the target side (Bengali) were POS-tagged/chunked by using the tools obtained from the consortium mode project “Development of Indian Languages to Indian Languages Machine Translation (ILILMT) System”.

For building the comparable corpora we have focused our attention on Wikipedia documents. To collect comparable English-Bengali document pairs we designed a crawler. The crawler first visits an English page, saves the raw text (in HTML format), and then finds the cross-lingual link (if exists) to find the corresponding Bengali document. Thus, we get one English-Bengali document pair. Moreover, the crawler visits the links found in each document and repeats the process. In this way, we develop a small aligned corpus of English-Bengali comparable document pairs. We retain only the textual information and all the other details are discarded. It is evident that the corpus is not confined to any particular domain. The challenge is to exploit this kind of corpus to help machine translation systems improve. The advantage of using such corpus is that it can be prepared easily unlike the one that is domain specific.

The effectiveness of the parallel fragments of text developed from the comparable corpora in the present work is demonstrated by using the standard log-linear PB-SMT model as our baseline system: GIZA++ implementation of IBM word alignment model 4, phrase extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003) on a held-out development set, target language model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with SRILM (Stolcke, 2002), and Moses decoder (Koehn et al., 2007).

4 System Architecture

4.1 PB-SMT(Baseline System)

Translation is modeled in SMT as a decision process, in which the translation $e_1^I = e_1..e_i..e_I$ of a source sentence $f_1^J = f_1..f_j..f_J$ is chosen to maximize (1)

$$\arg\max_{I,e_1^I} P(e_1^I \mid f_1^J) = \arg\max_{I,e_1^I} P(f_1^J \mid e_1^I).P(e_1^I) \qquad (1)$$

where $P(f_1^J \mid e_1^I)$ and $P(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $P(e_1^I \mid f_1^J)$ is directly modeled as a log-linear combination of features (Och and Ney, 2002), that usually comprise of *M* translational features, and the language model, as in (2):

$$\log P(e_1^I \mid f_1^J) = \sum_{m=1}^{M} \lambda_m h_m(f_1^J, e_1^I, s_1^K)$$

$$+ \lambda_{LM} \log P(e_1^I) \qquad (2)$$

where $s_1^k = s_1 ... s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\hat{e}_1, ..., \hat{e}_k)$ and $(\hat{f}_1, ..., \hat{f}_k)$ such that (we set $i_0 = 0$) (3):

$$\forall 1 \le k \le K, \quad s_k = (i_k, b_k, j_k),$$

$$\hat{e}_k = e_{i_{k-1}+1} ... e_{i_k},$$

$$\hat{f}_k = f_{b_k} ... f_{j_k}. \qquad (3)$$

and each feature $\hat{h}_m$ in (2) can be rewritten as in (4):

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) \qquad (4)$$

where $\hat{h}_m$ is a feature that applies to a single phrase-pair. It thus follows (5):

$$\sum_{m=1}^{M} \lambda_m \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) = \sum_{k=1}^{K} \hat{h}(\hat{f}_k, \hat{e}_k, s_k) \qquad (5)$$

where $\hat{h} = \sum_{m=1}^{M} \lambda_m \hat{h}_m$.

4.2 Chunking of English Sentences

We have used CRF-based chunking algorithm to chunk the English sentences in each document. The chunking breaks the sentences into linguistic phrases. These phrases may be of different sizes. For example, some phrases may be two words long and some phrases may be four words long. According to the linguistic theory, the intermediate constituents of the chunks do not usually take part in long distance reordering when it is translated, and only intra chunk reordering occurs. Some chunks combine together to make a longer phrase. And then some phrases again combine to make a sentence. The entire process maintains the linguistic definition of a sentence. Breaking the sentences into N-grams would have always generated phrases of length N but these phrases may not be linguistic phrases. For this reason, we avoided breaking the sentences into N-grams.

The chunking tool breaks each English sentence into chunks. The following is an example of how the chunking is done.

Sentence: India , officially the Republic of India , is a country in South Asia.

After Chunking: (India ,) (officially) (the Republic) (of) (India ,) (is) (a country) (in South Asia) (.)

We have further merged the chunks to form bigger chunks. The idea is that, we may sometimes find the translation of the merged chunk in the target side as well, in which case, we would get a bigger fragment of parallel text. The merging is done in two ways:

Strict Merging: We set a value ‘V’. Starting from the beginning, chunks are merged such that the number of tokens in each merged chunk does not exceed V.

```
Procedure Strict_Merge()
begin
    Oline ← null
    Cur_wc ← 0
    repeat
        Iline←Next Chunk
        Length←Number of Tokens in Iline
        if(Cur_wc + Length > V)
            Output Oline as the next merged chunk
            Cur_wc←Length
        else
            Append Iline at the end of Oline
            Add Length to Cur_wc
        end if
    while (there are more chunks)
end
```

Figure 1. Strict-Merging Algorithm.

Figure 1 describes the pseudo-code for strict merging.

For example, in our example sentence the merged chunks will be as following, where V=4: (India , officially) (the Republic of) (India , is) (a country) (in South Asia .)

```
Procedure Window_Merging()
begin
    Set_Chunk←Set of all English Chunks
    L←Number of chunks in Set_Chunk
    for i = 0 to L-1
        Words←Set of tokens in i-th Chunk in Set_Chunk
        Cur_wc←number of tokens in Words
        Ol←i-th chunk in Set_Chunk
        for j = (i+1) to (L-1)
            C←j-th chunk in Set_Chunk
            w←set of tokens in C
            l←number of tokens in w
            if(Cur_wc + l ≤ V)
                Append C at the end of Ol
                Add l to Cur_wc
            end if
        end for
        Output Ol as the next merged chunk
    end for
end
```

Figure 2. Window-Based Merging Algorithm.

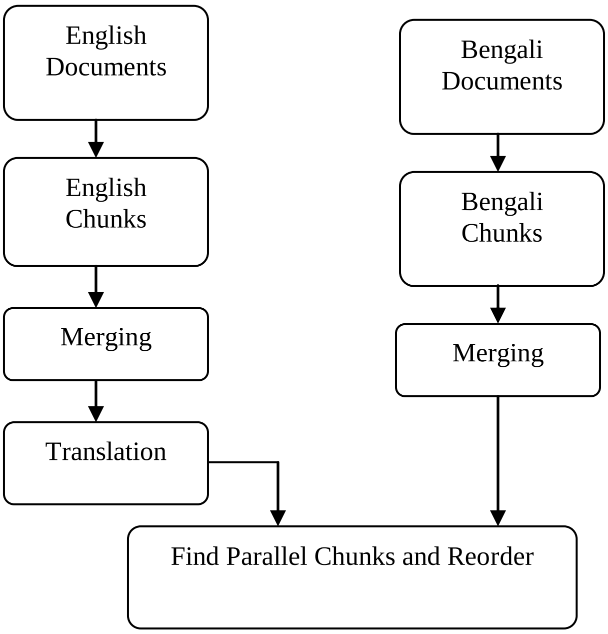

Figure 3. System Architecture for Finding Parallel Fragments

Window-Based Merging: In this type of chunking also, we set a value ‘V’, and for each chunk we try to merge as many chunks as possible so that the number of tokens in the merged chunk never exceeds V.

So, we slide an imaginary window over the chunks. For example, for our example sentence the merged chunks will be as following, where V = 4 :

(India , officially) (officially the Republic of) (the Republic of) (of India , is) (India , is) (is a country) (a country) (in South Asia .)

The pseudo-code of window-based merging is described in Figure 2.

4.3 Chunking of Bengali Sentences

Since to the best of our knowledge, there is no good quality chunking tool for Bengali we did not use chunking explicitly. Instead, strict merging is done with consecutive V number of tokens whereas window-based merging is done sliding a virtual window over each token and merging tokens so that the number of tokens does not exceed V.

4.4 Finding Parallel Chunks

After finding the merged English chunks they are translated into Bengali using a machine translation system that we have already developed. This is also the same machine translation system whose performance we want to improve. Chunks of each of the document pairs are then compared to find parallel chunks.

Each translated source chunk (translated from English to Bengali) is compared with all the target chunks in the corresponding Bengali-chunk document. When a translated source chunk is considered, we try to align each of its token to some token in the target chunk. Overlap between token two Bengali chunks B_1 and $B_{2,}$ where B_1 is the translated chunk and B_2 is the chunk in the Bengali document, is defined as follows:

Overlap(B_1,B_2) = Number of tokens in B_1 for which an alignment can be found in B_2.

It is to be noted that Overlap(B_1,B_2) $\neq$ Overlap(B_2 ,B_1). Overlap between chunks is found in both ways (from translated source chunk to target and from target to translated source chunk). If 70% alignment is found in both the overlap measures then we declare them as parallel. Two issues are important here: the comparison of two Bengali tokens and in case an alignment is found, which token to retrieve (source or target) and how to reorder them. We address these two issues in the next two sections.

4.5 Comparing Bengali Tokens

For our purpose, we first divide the two tokens into their *matra* (vowel modifiers) part and consonant part keeping the relative orders of characters in each part same. For example, Figure 4 shows the division of the word কলকাতা.

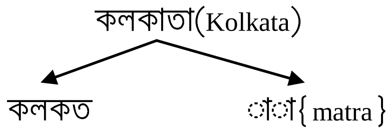

Figure 4. Division of a Bengali Word.

Respective parts of the two words are then compared. Orthographic similarities like minimum edit distance ratio, longest common subsequence ratio, and length of the strings are used for the comparison of both parts.

Minimum Edit Distance Ratio: It is defined as follows:

$$MEDR(B1,B2) = 1 - \frac{|ED(B1,B2)|}{max\ (|B1|,|B2|)}$$

where |B| is the length of the string B and ED is the minimum edit distance or *levenshtein distance* calculated as the minimum number of edit operations – insert, replace, delete – needed to transform B1 into B2.

Longest Common Subsequence Ratio: It is defined as follows:

$$LCSR(B1,B2) = \frac{|LCS(B1,B2)|}{max\ (|B1|,|B2|)}$$

where LCS is the longest common subsequence of two strings.

Threshold for matching is set empirically. We differentiate between shorter strings and larger strings. The idea is that, if the strings are short we cannot afford much difference between them to consider them as a match. In those cases, we check for exact match. Also, the threshold for consonant part is set stricter because our assumption is that consonants contribute more toward the word’s pronunciation.

4.6 Reordering of Source Chunks

When a translated source chunk is compared with a target chunk it is often found that the ordering of the tokens in the source chunk and the target chunk is different. The tokens in the target chunk have a different permutation of positions with respect to the positions of tokens in the source chunk. In those cases, we reordered the positions of the tokens in the source chunk so as to reflect the positions of tokens in the target chunk because it is more likely that the tokens will usually follow the ordering as in the target chunk. For example, the machine translation output of the English chunk “*from the Atlantic Ocean*” is “*থেকে(theke) আটলান্টিক (atlantic) মহাসাগর (mahasagar)*”. We found a target chunk “আটলান্টিক (*atlantic*) মহাসাগর (*mahasagar*) থেকে (*theke*) এবং (*ebong*)” with which we could align the tokens of the source chunk but in different relative order. Figure 5 shows the alignment of tokens.

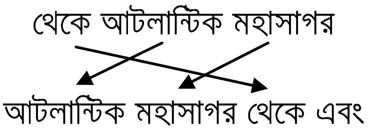

Figure 5. Alignment of Bengali Tokens.

We reordered the tokens of the source chunk and the resulting chunk was “আটলান্টিক মহাসাগর থেকে”.Also, the token “এবং” in the target chunk could not find any alignment and was discarded. The system architecture of the present system is described in figure 3.

5 Experiments And Results

5.1 Baseline System

We randomly extracted 500 sentences each for the development set and test set from the initial parallel corpus, and treated the rest as the training corpus. After filtering on the maximum allowable sentence length of 100 and sentence length ratio of 1:2 (either way), the training corpus contained 22,492 sentences.

	V=4	**V=7**
Number of English Chunks(Strict-Merging)	579037	376421
Number of English Chunks(Window-Merging)	890080	949562
Number of Bengali Chunks(Strict-Merging)	69978	44113
Number of Bengali Chunks(Window-Merging)	230025	249330

Table 1. Statistics of the Comparable Corpus

	V=4	**V=7**
Number of Parallel Chunks(Strict-Merging)	1032	1225
Number of Parallel Chunks(Window-Merging)	1934	2361

Table 2. Number of Parallel Chunks found

		BLEU	**NIST**
Baseline System(PB-SMT)		10.68	4.12
Baseline + Parallel Chunks(Strict-Merging)	**V=4**	10.91	4.16
	V=7	11.01	4.16
Baseline + Parallel Chunks(Window-Merging)	**V=4**	11.55	4.21
	V=7	**11.87**	**4.29**

Table 3.Evaluation of the System

In addition to the target side of the parallel corpus, a monolingual Bengali corpus containing 406,422 words from the tourism domain was used for the target language model. We experimented with different n-gram settings for the language model and the maximum phrase length, and found that a 5-gram language model and a maximum phrase length of 7 produced the optimum baseline result. We therefore carried out the rest of the experiments using these settings.

5.2 Improving Baseline System

The comparable corpus consisted of 582 English-Bengali document pairs.

We experimented with the values V=4 and V=7 while doing the merging of chunks both in English and Bengali. All the single token chunks were discarded. Table 1 shows some statistics about the merged chunks for V=4 and V=7.It is evident that number of chunks in English documents is far more than the number of chunks in Bengali documents. This immediately suggests that Bengali documents are less informative than English documents. When the English merged chunks were passed to the translation module some of the chunks could not be translated into Bengali. Also, some chunks could be translated only partially, i.e. some tokens could be translated while some could not be. Those chunks were discarded. Finally, the number of (Strict-based) English merged-chunks and number of (Window-based) English merged-chunks were 285756 and 594631 respectively.

Two experiments were carried out separately. Strict-based merged English chunks were compared with Strict-Based merged Bengali chunks. Similarly, window-based merged English chunks were compared with window-based merged Bengali chunks. While searching for parallel chunks each translated source chunk was compared with all the target chunks in the corresponding document. Table 2 displays the number of parallel chunks found. Compared to the number of chunks in the original documents the number of parallel chunks found was much less. Nevertheless, a quick review of the parallel list revealed that most of the chunks were of good quality.

5.3 Evaluation

We carried out evaluation of the MT quality using two automatic MT evaluation metrics: BLEU (Papineni et al., 2002) and NIST (Doddington, 2002). Table 3 presents the experimental results. For the PB-SMT experiments, inclusion of the extracted strict merged parallel fragments from comparable corpora as additional training data presented some improvements over the PB-SMT baseline. Window based extracted fragments are added separately with parallel corpus and that also provides some improvements over the PB baseline; however inclusion of window based extracted phrases in baseline system with phrase length 7 improves over both strict and baseline in term of BLEU score and NIST score.

Table 3 shows the performance of the PB-SMT system that shows an improvement over baseline with both strict and window based merging even if, we change their phrase length from 4 to 7. Table 3 shows that the best improvement is achieved when we add parallel chunks as window merging with phrase length 7. It gives 1.19 BLEU point, i.e., 11.14% relative improvement over baseline system. The NIST score could be improved up to 4.12%. Bengali is a morphologically rich language and has relatively free phrase order. The strict based extraction does not reflect much improvement compared to the window based extraction because strict-merging (Procedure Strict_Merge) cannot cover up all the segments on either side, so very few parallel extractions have been found compared to window based extraction.

6 Conclusion

In this work, we tried to find English-Bengali parallel fragments of text from a comparable corpus built from Wikipedia documents. We have successfully improved the performance of an existing machine translation system. We have also shown that out-of-domain corpus happened to be useful for training of a domain specific MT system. The future work consists of working on larger amount of data. Another focus could be on building ad-hoc comparable corpus from WEB and using it to improve the performance of an existing out-of-domain MT system. This aspect of work is particularly important because the main challenge would be of domain adaptation.

Acknowledgements

This work has been partially supported by a grant from the English to Indian language Machine Translation (EILMT) project funded by the Department of Information and Technology (DIT), Government of India.

Reference

Chiao, Y. C., & Zweigenbaum, P. (2002, August). Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th international conference on Computational linguistics-Volume 2* (pp. 1-5). Association for Computational Linguistics.

Déjean, H., Gaussier, É., & Sadat, F. (2002). Bilingual terminology extraction: an approach based on a multilingual thesaurus applicable to comparable corpora. In *Proceedings of the 19th International Conference on Computational Linguistics COLING* (pp. 218-224).

Doddington, G. (2002, March). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the second international conference on Human Language Technology Research (pp. 138-145). Morgan Kaufmann Publishers Inc..

Fung, P., & McKeown, K. (1997, August). Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora* (pp. 192-202).

Fung, P., & Yee, L. Y. (1998, August). An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1* (pp. 414-420). Association for Computational Linguistics.

Hiroyuki, K. A. J. I. (2005). Extracting translation equivalents from bilingual comparable corpora. *IEICE Transactions on information and systems, 88*(2), 313-323.

Kneser, R., & Ney, H. (1995, May). Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on* (Vol. 1, pp. 181-184). IEEE.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... & Herbst, E. (2007, June). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (pp. 177-180). Association for Computational Linguistics.

Koehn, P., Och, F. J., & Marcu, D. (2003, May). 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* (pp. 48-54). Association for Computational Linguistics.

Munteanu, D. S., & Marcu, D. (2006, July). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 81-88). Association for Computational Linguistics..

Och, F. J. (2003, July). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 160-167). Association for Computational Linguistics.

Och, F. J., & Ney, H. (2000). Giza++: Training of statistical translation models.

Otero, P. G. (2007). Learning bilingual lexicons from comparable english and spanish corpora. *Proceedings of MT Summit xI,* 191-198.

Otero, P. G., & López, I. G. (2010). Wikipedia as multilingual source of comparable corpora. In *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora, LREC* (pp. 21-25).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computa-*

tional linguistics (pp. 311-318). Association for Computational Linguistics.

Rapp, R. (1999, June). Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 519-526). Association for Computational Linguistics.

Saralegui, X., San Vicente, I., & Gurrutxaga, A. (2008). Automatic generation of bilingual lexicons from comparable corpora in a popular science domain. In *LREC 2008 workshop on building and using comparable corpora*.

Smith, J. R., Quirk, C., & Toutanova, K. (2010, June).Extracting parallel sentences from comparable corpora using document level alignment. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 403-411). Association for Computational Linguistics.

Stolcke, A. (2002, September). SRILM-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing* (Vol. 2, pp. 901-904).