Context-Based Sentence Alignment in Parallel Corpora

Ergun Biçici

Koç University
Rumelifeneri Yolu 34450
Sariyer, Istanbul, Turkey
ebicici@ku.edu.tr

Abstract. This paper presents a language-independent context-based sentence alignment technique given parallel corpora. We can view the problem of aligning sentences as finding translations of sentences chosen from different sources. Unlike current approaches which rely on pre-defined features and models, our algorithm employs features derived from the distributional properties of words and does not use any language dependent knowledge. We make use of the context of sentences and the notion of Zipfian word vectors which effectively models the distributional properties of words in a given sentence. We accept the context to be the frame in which the reasoning about sentence alignment is done. We evaluate the performance of our system based on two different measures: sentence alignment accuracy and sentence alignment coverage. We compare the performance of our system with commonly used sentence alignment systems and show that our system performs 1.2149 to 1.6022 times better in reducing the error rate in alignment accuracy and coverage for moderately sized corpora.

Keywords
sentence alignment, context, Zipfian word vectors, multilingual

1 Introduction

Sentence alignment is the task of mapping the sentences of two given parallel corpora which are known to be translations of each other to find the translations of corresponding sentences. Sentence alignment has two main burdens: solving the problems incurred by a previous erroneous sentence splitting step and aligning parallel sentences which can later be used for machine translation tasks. The mappings need not necessarily be 1-to-1, monotonic, or continuous. Sentence alignment is an important preprocessing step that affects the quality of parallel text.

A simple approach to the problem of sentence alignment would look at the lengths of each sentence taken from parallel corpora and see if they are likely to be translations of each other. In fact, it was shown that paragraph lengths for the English-German parallel corpus from the economic reports of Union Bank of Switzerland (UBS) are highly correlated with a correlation value of 0.991 [1]. A more complex approach would look at the neighboring sentence lengths as well. Our approach is based on this knowledge of context for given sentences from each corpus and the knowledge of distributional features of words, which we name Zipfian word vectors, for alignment purposes. A Zipfian

word vector is an order-free representation of a given sentence in a corpus, in which the length and the number of words in each entry of the vector are determined based on the quantization of the frequencies of all words in the corpus.

In this paper, we consider sentence alignment based on the local context information. The resulting methodology is language-independent; it can handle non-monotonic alignments; it does not require any stemming, dictionaries, or anchors; it handles deletions; and it extends the type of alignments available up to 6-way. Sentence alignments of given parallel corpora are determined by looking at the local context of a given sentence which consists of the surrounding sentences. The problem of sentence alignment is a central problem in machine translation and similar in essence to many other problems that involve the identification of mappings. It is a subset of the problem of *sequence comparison*, which deals with difficult comparisons that arise when the correspondence of items in the sequences are not known in advance [2]. We used a publicly available and easily accessible dataset [3] for our experiments, so that our results can be easily replicated by others.

We observe that valuable information can be inferred from the context of given sentences and their distributional properties for alignment purposes. The following sections are organized as follows. In the next section, we review related work and present its limitations. In Sect. 3, we provide a formalization of the sentence alignment problem, define Zipfian word vectors, present our feature representation, and discuss context in sentence alignment and our alignment algorithm. In Sect. 4, we present the results of our experiments and the last section concludes.

2 Related Work

Brown *et. al.* [4] provide a statistical technique for sentence alignment using the number of word tokens in each sentence as well as the comments in their dataset (Canadian Hansards corpora [1]), which serve as anchor points. They define an alignment as a sequence of beads, which are considered as groupings of English and French sentences that lengths that are close. Gale and Church [1] observe that sentence lengths of source and target sentences are correlated. They limit their alignments to 1-1, 1-0, 0-1, 2-1, 1-2, and 2-2 types of mappings, where the numbers represent the number of sentences that map to each other. The reason for their choice in using sentence lengths in terms of characters rather than in terms of word tokens as was chosen by Brown *et. al.* [4] is that since there are more characters there is less uncertainty.

Both Brown *et. al.* and Gale and Church [1] assume that the corpus is divided into chunks and they ignore word identities. Chen [5] describes an algorithm that constructs a simple statistical word-to-word translation model on the fly during sentence alignment. The alignment of a corpus $(\mathcal{S}, \mathcal{T})$ is the alignment $\mathbf{m}$ that maximizes $P(\mathcal{T}, \mathbf{m} \mid \mathcal{S})$, where P denotes the probability. Chen found that 100 sentence pairs are sufficient to train the model to a state where it can align correctly. Moore's [6] sentence alignment model combines sentence-length-based and word-correspondence-based approaches,

[1] Available from Linguistic Data Consortium at `http://www.ldc.upenn.edu/`

achieving high accuracy at a modest computational cost. Moore uses a modified version of the IBM Translation Model 1 [7]:

$$P(T \mid S) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} \mathrm{tr}(t_j|s_i),$$

where $\mathrm{tr}(t_j|s_i)$ corresponds to the translation probability of the word $t_j \in T = \{t_1, \ldots, t_m\}$ given $s_i \in S = \{s_1, \ldots, s_l\}$ and ϵ is some small fixed number. Instead of $P(T|S)$, Moore makes use of $P(S, T)$ due to the noisy channel model [8], since S is hypothetically corrupted by some "noise" and turned into t.

Context and its selection is very important in many areas of natural language processing. Most of the work on context focuses on finding an optimal context size which gives good performance globally on the test cases. Yet this optimal value is sensitive to the type of ambiguity [9]. The dynamic nature of the context is noticed for the word sense disambiguation task by Yarowsky and Florian [10] and they further claimed that the context sizes for nouns, verbs, and adjectives should be in the 150, 60-80, and 5 word vicinity of a given word respectively. Wang [11] gives a nice example of word senses' context dependence in Fig. 1. As we increase the size of the context, the sense of the Chineese word varies between think and read. Ristad [12] makes use of a greedy heuristic to extend a given context for the purpose of finding models of language with fewer parameters and lower entropy. In our previous work [13], we examined alternatives for local context models for sentence alignment. As we did previously, in this work, we accept the context to be the frame in which the reasoning about sentence alignment is done.

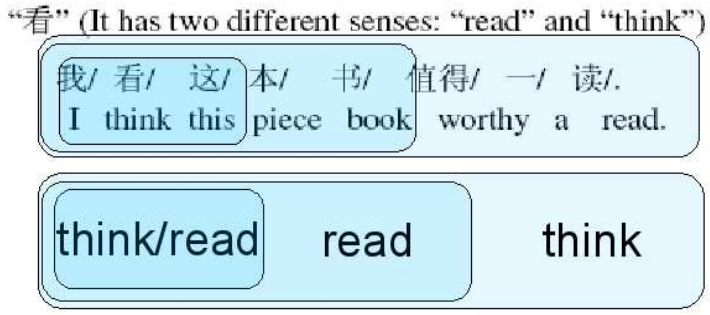

Fig. 1. Word sense dependence on context

Earlier work on sentence alignment assume that the order of sentences in each corpus is preserved; as the beads on a string preserve the order, their models assume that the mapping function $\mathbf{m}$ is monotonic. Sentence alignment literature makes extensive use of simplifying assumptions (e.g. the existence of anchors, dictionaries, or stemming), biased success criterion (e.g. selecting only 1-1 type alignments or removing badly aligned sentences from consideration), and the use of datasets that cannot be

qualitatively judged and compared to other results. In this paper, we overcome these limitations by removing simplifying assumptions about the dataset and generalizing the problem space by generalizing our representation of the data. Our goal is not to seek the best performance in only 1-1 type alignments since machine translation tasks cannot be reduced to 1-1 type alignments. Although 1-1 type alignments constitute 97.21% of the 52594 alignments overall, they cover only 96.01% of the 106504 sentences involved in the Multext-East [3] dataset. We also use a new measure of success, sentence alignment coverage, which also considers the number of sentences involved in the alignment. We use the Multext-East [2] corpus, which provides us access to large amounts of manually sentence-split and sentence-aligned parallel corpora and a good dataset for the evaluation of performance. As this dataset contains alignments for 8 different language pairs, it suits well for demonstrating our system's language independence.

3 Sentence Alignment

3.1 Problem Formulation

A *parallel corpus* is a tuple $(\mathcal{S}, \mathcal{T})$, where $\mathcal{S}$ denotes the source language corpus and $\mathcal{T}$ denotes the target language corpus such that $\mathcal{T}$ is the translation of $\mathcal{S}$. Since the translation could have been done out of order or lossy, the task of *sentence alignment* is to find a mapping function, $\mathbf{m} : \mathcal{S} \rightarrow \mathcal{T}$, such that a set of sentences $T \subseteq \mathcal{T}$ where $T = \mathbf{m}(S)$ is the translation of a set of sentences $S \subseteq \mathcal{S}$. Then, under the mapping $\mathbf{m}$, we can use T whenever we use S.

We assume that $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}\}$, where $|\mathrm{corpus}|$ refers to the number of sentences in corpus and s_i and t_i correspond to the ith sentences in $\mathcal{S}$ and in $\mathcal{T}$ respectively. The sentences in $\mathcal{S}$ and $\mathcal{T}$ form an ordered set where an *ordered set* is an n-tuple, denoted by $\{a_1, a_2, \ldots, a_n\}_{\leqslant}$, such that there exists a total order, $\leqslant$, defined on the elements of the set. We also assume that a set of sentences $S \subseteq \mathcal{S}$ where $S = \{s_i, s_{i+1}, \ldots, s_j\}$ is chosen such that $\forall k, i \leq k < j,\ s_k \leqslant_{\mathcal{S}} s_{k+1}$. The same argument applies for a set of sentences selected from $\mathcal{T}$. Therefore, it is also meaningful to order two sets of sentences S_1 and S_2 selected from a given corpus $\mathcal{S}$ with the following semantics: Let start_{S_1} and start_{S_2} be the starting sentences of S_1 and S_2 correspondingly, then, $S_1 \leqslant_{\mathcal{S}} S_2 \Leftrightarrow \mathrm{start}_{S_1} \leqslant_{\mathcal{S}} \mathrm{start}_{S_2}$. A mapping $\mathbf{m} : \mathcal{S}_{\leqslant_{\mathcal{S}}} \rightarrow \mathcal{T}_{\leqslant_{\mathcal{T}}}$, is *monotone* or *order-preserving*, if for $S_1, S_2 \subseteq \mathcal{S}$, $S_1 \leqslant_{\mathcal{S}} S_2$ implies $\mathbf{m}(S_1) \leqslant_{\mathcal{T}} \mathbf{m}(S_2)$, where $\mathbf{m}(S_1), \mathbf{m}(S_2) \subseteq \mathcal{T}$.

The usual evaluation metric used is the percentage of correct alignments found in a given set of alignments, which we name sentence alignment accuracy. This measure does not differentiate between an alignment that involves only one sentence as in 1-0 or 0-1 type alignments and an alignment that involves multiple sentences as in 1-5. Therefore, we define sentence alignment coverage as follows:

Definition 1 (Sentence Alignment Coverage). *Sentence alignment coverage is the percentage of sentences that are correctly aligned in a given parallel corpus.*

Thus, for sentence alignment coverage, an alignment of type 1-5 is three times more valuable than an alignment of type 1-1.

[2] Also available at `http://nl.ijs.si/ME/V3/`

3.2 Zipfian Word Vectors

It is believed that distribution of words in large corpora follow what is called Zipf's Law, where "a few words occur frequently while many occur rarely" [14]. We assume that distributions similar to Zipfian are ubiquitous in all parallel corpora. Based on this assumption, we create Zipfian word vectors by making use of the distributions of words in a given corpus.

Definition 2 (Zipfian Word Vector). *Given a set of sentences, S, chosen from a given corpus, $\mathcal{S}$, where $\mathrm{maxFreq}$ represents the frequency of the word with the maximum frequency in $\mathcal{S}$, and a binning threshold, b, the Zipfian word vector representation of S is defined as a vector V of size $\frac{\log(\mathrm{maxFreq})}{\log(b)}$, where $V[i]$ holds the number of words in S that have a frequency of $\lfloor \frac{\log(\mathrm{freq}(w))}{\log(b)} \rfloor = i$ for word $w \in S$.*

Thus, each bin contains the number of words with similar frequencies in the given corpus. We assume that $\mathrm{ZWV(S)}$ is a function that returns the Zipfian word vector of a given set of sentences S. Thus, for a single sentence as in:

$S =$`" big brother is watching you " , the caption beneath it ran .`,

the Zipfian word vector becomes:

$$\mathrm{ZWV(S)} = [14, 1, 3, 0, 1, 3, 2, 0, 1, 1, 2],$$

where the sentence length in the number of tokens is added to the beginning of the Zipfian word vector as well. In Sect. 4 we examined the performance when different sentence length definitions for the Zipfian word vectors used in the system. Note that Zipfian word vectors contain information about anything that is recognized as a token after tokenization. We used the Penn Tree Bank [15] tokenizer, which was designed for English but still effective since all of the Eastern European languages we experimented with use the same punctuation characters with English.

The TCat concept [16] used for text classification is similar in its use of Zipfian distribution of words. While TCat is based on three levels of frequency (high, medium, and low frequency levels) we vary the length of the Zipfian word vector to increase the accuracy in the learning performance and adapt to the problem. Also, in TCat, each level of frequency behaves as a binary classifier, differentiating between positive and negative examples whereas each bin in our model behaves as a quantization of features to be used in learning.

3.3 Feature Representation

We assume that $\mathcal{S} = \{S_1, \ldots, S_i, \ldots, S_N\}$ and $\mathcal{T} = \{T_1, \ldots, T_i, \ldots, T_N\}$ where N is the total number of alignments and S_i and T_i correspond to the set of sentences involved in the ith alignment. For each set of sentences, that become a candidate for alignment within the sentence alignment algorithm, we create what we call the *Zipfian*

word matrix. The Zipfian word matrix of a given set of sentences, S, is essentially the matrix we get when we concatenate the Zipfian word vectors surrounding S based on S's local context, which contains at most $2 \times w + 1$ rows for a given window size of w. Then the decision whether T is the translation of S is based on the two dimensional (2D) weight decaying Pearson correlation coefficient of their corresponding Zipfian word matrices.

Weight decaying is applied to the sentences that are far from S, which is the sentence according to which the context is calculated. Exponential decaying is applied with decaying constant set to 0.7. The use of weight decaying for 2D Pearson correlation coefficient does not improve statistically significantly, but it increases the accuracy and decreases the variance; hence giving us a more robust value.

3.4 Context in Sentence Alignment

The sentence alignment algorithm we have developed is context-based in the sense that features belonging to the sentences that come before and after the current sentence are also considered. We represent the local context of a given set of sentences as a pair, the number of sentences to consider before and after the given set of sentences.

We consider two options for context size selection: (i) static context size selection, which uses (w, w) for both the source and the target local contexts; (ii) symmetric local context selection, which uses (b, a) for both the source and the target local contexts, where b and a correspond to the number of sentences that come before and after a given sentence. For a given context window size limit, w, where $2w = b + a$, there can be w^2 symmetric local context selections for the pair S and T. For the Lithuanian-English pair on the first chapter when w=10 the maximum score attaining symmetric local context is found to be $(3.12, 3.35)$.

3.5 Sentence Alignment Algorithm

Our sentence alignment algorithm makes use of dynamic programming formulation with up to 6-way alignments with extensions to handle non-monotonic alignments. The algorithm is essentially a modified version of the Needleman-Wunsch sequence alignment algorithm [17] with gap penalty set to -0.5. Further discussion on dynamic programming methodology to solve sentence alignment problems can be found in [1] or in [5]. We use the assumption that the alignments are found close to the diagonal of the dynamic programming table to further speed up the alignment process. Another property of our system is its ability to model up to 6-way alignments.

Another benefit in using sequence alignment methodology is our ability to model not only constant gap costs in the alignments but also affine as well as convex gap costs (a good description for affine and convex gap costs is in [18]). However, as the dataset does not provide enough contiguous gaps, we have not tested this capability; yet it is likely that affine and convex gap costs model the gap costs in sentence alignment better.

3.6 Non-monotonic Alignments

Previous approaches assume that sentence alignment functions are monotonic. We relax this assumption and assume monotonicity in the flow of the arguments (semantic mono-

tonicity) within a 4 sentence window from both the source and target corpus. Thus, we also allow non-monotonic alignments of type $([1,2],[1,2]\times[1,2],[1,2])$, where $[1,2]$ corresponds to 1 or 2 sentences that are selected and $\times$ means that $\blacksquare_1$ is aligned with $\blacksquare_4$ and $\blacksquare_2$ with $\blacksquare_3$ in $(\blacksquare_1,\blacksquare_2\times\blacksquare_3,\blacksquare_4)$. Although the Multext-East dataset does not contain non-monotonic alignment examples, we have observed that by allowing non-monotonicity, in some cases our system is able to backtrack and recover from errors that were incurred from previous steps and therefore reducing noise.

4 Experiments

We used the George Orwell's 1984 corpus from Multext-East [3], which contains manually sentence split and aligned translations for English, Bulgarian, Czech, Estonian, Hungarian, Romanian, Latvian, Lithuanian, Serbo-Croatian, and Slovene, which are abbreviated as `en`, `bg`, `cs`, `et`, `hu`, `ro`, `lv`, `lt`, `sr`, and `sl` respectively. In all of our experiments, the corresponding language pair is chosen to be English. We compared the results of our system with that of hunalign [19] and Moore's system [6]. Without an input dictionary, hunalign makes use of the Gale and Church [1] algorithm which is based on sentence lengths, and builds a dictionary dynamically based on this alignment.

We calculated the Pearson correlation coefficient score for the sentence lengths in characters and in word tokens for the datasets in the full parallel corpus based on the correct alignments. The correlation coefficients for the sentence lengths in characters are found as: (bg, 0.9692), (cs, 0.9674), (et, 0.9738), (hu, 0.9585), (lt, 0.9719), (ro, 0.9730), (sr, 0.9694), (sl, 0.9805). The correlation coefficients for the sentence lengths in word tokens are found as: (bg, 0.9598), (cs, 0.9542), (et, 0.9489), (hu, 0.9440), (lt, 0.9554), (ro, 0.9578), (sr, 0.9575), (sl, 0.9669). The total number of alignments is 2733 and sentences is 5596 in the first chapter of Multext-East, which gives a moderately sized corpora, when all languages are combined. These numbers rise to 52594 for alignments and 106504 for sentences when the full dataset is used.

Our first couple of experiments are based on choosing appropriate parameters. Fig. 2 show the change in sentence alignment accuracy with varying window sizes in the first chapter of the corresponding corpora in Multext-East (the corresponding graph for the sentence alignment coverage has similar trends). Based on these graphs, we chose the window size w to be 7. To reduce the complexity of calculations to a manageable value, the value of b is chosen to be 10.

We have experimented with using different sentence length definitions for the Zipfian word vectors used in the system. The alternatives that we consider are: (i) no sentence length information, (ii) only the sentence length in the number of characters added, (iii) only the sentence length in the number of word tokens added, (iv) both added. The results show that when w=7, adding only the sentence length in the number of tokens perform better than other alternatives that we considered and the performance decreases in this order: (iii) $\gg$ (iv) $\gg$ (ii) $\gg$ (i).

In the first chapter of the dataset, our context-based sentence alignment algorithm (CBSA), CBSA when the average score for the symmetric local contexts is used (CBSAavg), and non-monotonic CBSA (nmCBSA) sentence alignment techniques reduce the sentence alignment accuracy error rate of hunalign by 1.3611, 1.2895, and 1.2149

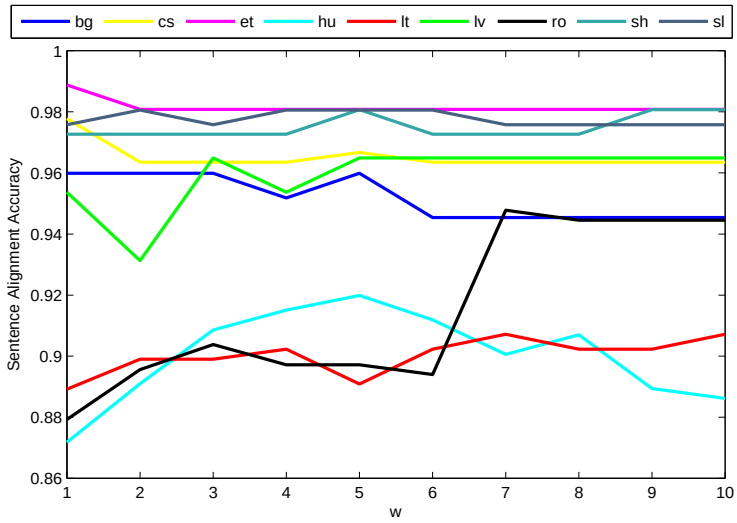

Fig. 2. Sentence Alignment Accuracy versus Window Size w

times and of Moore by 1.5648, 1.4825, and 1.3967 times respectively. The results can be seen in Table 1. In terms of sentence alignment coverage, CBSA, CBSAavg, and nmCBSA reduce the error rate of hunalign by 1.5401, 1.4602, and 1.2827 and of Moore by 1.6022, 1.5190, and 1.3343 times respectively. The results can be seen in Table 3.

The results on the full dataset are presented in Table 2 and in Table 4. Our CBSA and nmCBSA sentence alignment techniques reduce the sentence alignment accuracy error rate of hunalign by 2.0243 and 1.4360 times and increase the error rate of Moore by 1.7404 and 2.4534 times respectively. In terms of sentence alignment coverage, CBSA and nmCBSA reduce the error rate of hunalign by 2.0678 and 1.4986 times and increase the error rate of Moore by 1.6129 and 2.2254 times respectively.

First Chapter	**Sentence Alignment Accuracy**				
Language	**hunalign**	**Moore**	**CBSA, w=7**	**CBSA, w=7, average**	**nmCBSA, w=7**
Bulgarian	**96.74** / **3.26**	96.09 / 3.91	95.44 / 4.56	**96.74** / **3.26**	95.11 / 4.89
Czech	96.14 / 3.86	95.82 / 4.18	**96.78** / **3.22**	**96.78** / **3.22**	**96.78** / **3.22**
Estonian	**99.68** / **0.32**	98.39 / 1.61	98.39 / 1.61	98.39 / 1.61	98.39 / 1.61
Hungarian	87.86 / 12.14	88.96 / 11.04	91.64 / 8.36	**92.98** / **7.02**	91.30 / 8.70
Latvian	95.71 / 4.29	92.74 / 7.26	**97.69** / **2.31**	**97.69** / **2.31**	**97.69** / **2.31**
Lithuanian	88.44 / 11.56	82.31 / 17.69	**92.52** / **7.48**	91.84 / 8.16	92.18 / 7.82
Romanian	89.86 / 10.14	95.27 / 4.73	**95.61** / **4.39**	91.22 / 8.78	92.23 / 7.77
Serbo-Croatian	**98.70** / **1.30**	97.08 / 2.92	97.73 / 2.27	97.73 / 2.27	97.73 / 2.27
Slovene	97.70 / 2.30	97.04 / 2.96	98.36 / 1.64	**98.68** / **1.32**	98.36 / 1.64

Table 1. Sentence alignment accuracy per English - language alignments on the first chapter.

Full Dataset	Sentence Alignment Accuracy				
Language	**hunalign**	**Moore**	**CBSA,** w=4	**CBSA,** w=7	**nmCBSA,** w=7
Bulgarian	72.92 / 27.08	**98.77** / **1.23**	98.23 / 1.77	98.29 / 1.71	97.40 / 2.60
Czech	95.65 / 4.35	**97.92** / **2.08**	95.92 / 4.08	96.18 / 3.82	94.89 / 5.11
Estonian	96.88 / 3.12	**98.36** / **1.64**	97.02 / 2.98	97.13 / 2.87	95.53 / 4.47
Hungarian	94.77 / 5.23	**97.74** / **2.26**	95.91 / 4.09	96.25 / 3.75	94.47 / 5.53
Lithuanian	95.42 / 4.58	**97.12** / **2.88**	95.60 / 4.40	95.65 / 4.35	93.60 / 6.40
Romanian	92.50 / 7.50	**96.52** / **3.48**	92.39 / 7.61	92.55 / 7.45	90.75 / 9.25
Serbo-Croatian	97.21 / 2.79	**98.40** / **1.60**	97.53 / 2.47	97.54 / 2.46	96.06 / 3.94
Slovene	97.91 / 2.09	**98.96** / **1.04**	97.98 / 2.02	98.15 / 1.85	97.47 / 2.53

Table 2. Sentence alignment accuracy per English - language alignments on the full dataset.

First Chapter	Sentence Alignment Coverage				
Language	**hunalign**	**Moore**	**CBSA,** w=7	**CBSA,** w=7**, average**	**nmCBSA,** w=7
Bulgarian	95.34 / 4.66	94.86 / 5.14	94.54 / 5.46	**95.99** / **4.01**	90.54 / 9.46
Czech	94.92 / 5.08	95.24 / 4.76	**96.35** / **3.65**	**96.35** / **3.65**	**96.35** / **3.65**
Estonian	**99.52** / **0.48**	98.08 / 1.92	98.08 / 1.92	98.08 / 1.92	98.08 / 1.92
Hungarian	84.30 / 15.70	85.90 / 14.10	90.06 / 9.94	**91.51** / **8.49**	89.74 / 10.26
Latvian	92.65 / 7.35	90.26 / 9.74	**96.49** / **3.51**	**96.49** / **3.51**	**96.49** / **3.51**
Lithuanian	84.85 / 15.15	79.15 / 20.85	**90.72** / **9.28**	89.90 / 10.10	90.39 / 9.61
Romanian	86.79 / 13.21	93.64 / 6.36	**94.78** / **5.22**	89.72 / 10.28	90.54 / 9.46
Serbo-Croatian	**97.75** / **2.25**	96.46 / 3.54	97.27 / 2.73	97.27 / 2.73	97.27 / 2.73
Slovene	95.81 / 4.19	95.64 / 4.36	97.58 / 2.42	**98.06** / **1.94**	97.58 / 2.42

Table 3. Sentence alignment coverage per English - language alignments on the first chapter.

The comparison of the performance of different sentence alignment techniques is presented in Table 5. The reason for the worse performance of CBSA than Moore's system in the full dataset can be explained by the contribution of the word translation models that the Moore's system builds. Thus, given a moderately sized corpus with which to work with, hunalign as well as Moore's system will likely to have a bad performance since the word translation probabilities and the dictionaries that their systems build respectively will be more error-prone. Therefore, it might be a good idea to incorporate word translation models to the CBSA system when working with large corpora.

5 Conclusion

We have developed a novel language-independent context-based sentence alignment technique given parallel corpora. We can view the problem of aligning sentences as finding translations of sentences chosen from different sources. Unlike current approaches which rely on pre-defined features and models, our algorithm employs features derived from the distributional properties of words in sentences and does not use

Full Dataset	**Sentence Alignment Coverage**				
Language	**hunalign**	**Moore**	**CBSA,** w=4	**CBSA,** w=7	**nmCBSA,** w=7
Bulgarian	72.72 / 27.28	**98.65** / **1.35**	98.20 / 1.80	98.27 / 1.73	97.37 / 2.63
Czech	94.39 / 5.61	**97.52** / **2.48**	95.51 / 4.49	95.80 / 4.20	94.42 / 5.58
Estonian	95.67 / 4.33	**97.98** / **2.02**	96.44 / 3.56	96.56 / 3.44	94.86 / 5.14
Hungarian	93.60 / 6.40	**97.12** / **2.88**	95.41 / 4.59	95.77 / 4.23	93.89 / 6.11
Lithuanian	94.07 / 5.93	**96.50** / **3.50**	94.97 / 5.03	95.04 / 4.96	92.92 / 7.08
Romanian	89.85 / 10.15	**95.54** / **4.46**	91.19 / 8.81	91.35 / 8.65	89.43 / 10.57
Serbo-Croatian	96.60 / 3.40	**98.17** / **1.83**	97.37 / 2.63	97.38 / 2.62	95.83 / 4.17
Slovene	97.15 / 2.85	**98.72** / **1.28**	97.68 / 2.32	97.86 / 2.14	97.18 / 2.82

Table 4. Sentence alignment coverage per English - language alignments on the full dataset.

	Alignment Method	**Alignment Mistakes**	**Alignments Per Mistake**	**Sentence Mistakes**	**Sentences Per Mistake**
	hunalign	147	18.55	422	13.26
First	**Moore**	169	16.17	439	12.75
Chapter	**CBSA,** w=7	**108**	**25.31**	**274**	**20.43**
	CBSA, w=7**, average**	114	23.97	289	19.37
	nmCBSA, w=7	121	22.58	329	17.02
	hunalign	3745	14.04	8788	12.12
Full	**Moore**	**1063**	**49.50**	**2635**	**40.42**
Dataset	**CBSA,** w=7	1850	28.44	4250	25.06
	nmCBSA, w=7	2608	20.16	5864	18.16

Table 5. Comparing the performance of different sentence alignment methods.

any language dependent knowledge. The resulting sentence alignment methodology is language-independent; it can handle non-monotonic alignments; it does not require any stemming, dictionaries, or anchors; it handles deletions; and it extends the type of alignments available up to 6-way.

The main advantage of Moore's and Chen's methods are their employment of the word translation probabilities and their updates when necessary. It is a custom to feed previous alignment results back into the aligner to further improve on the results. This process is generally referred to as bootstrapping and there may be multiple passes needed until convergence. We can easily improve our model by making use of word translation models and bootstrapping.

We provide formalizations for sentence alignment task and the context for sentence alignment. We use the notion of Zipfian word vectors which effectively presents an order-free representation of the distributional properties of words in a given sentence. We define two dimensional weight decaying correlation scores for calculating the similarities between sentences.

We accept the context to be the frame in which the reasoning about sentence alignment is done. We can also further improve our model by using a pre-specified dictionary, by dynamically building a dictionary, by using stemming, by using a larger corpus to estimate frequencies and generating Zipfian word vectors based on them, by using larger window sizes to select the local context size from, or by using bootstrapping which makes use of the previously learned alignments in previous steps.

We evaluate the performance of our system based on two different measures: sentence alignment accuracy and sentence alignment coverage. We compare the performance of our system with commonly used sentence alignment systems and show that our system performs 1.2149 to 1.6022 times better in reducing the error rate in alignment accuracy and coverage for moderately sized corpora. The addition of word translation probabilities and models of word order to the CBSA system is likely to achieve better sentence alignment results when working with large corpora.

Acknowledgments

The research reported here was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK). The author would like to thank Deniz Yuret for helpful discussions and for guidance and support during the term of this research.

References

1. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. Computational Linguistics **19** (1993) 75–102
2. Kruskal, J.B.: An overview of sequence comparison. In Sankoff, D., Kruskal, J.B., eds.: Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley (1983) 1–44
3. Erjavec, T.: MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In: Fourth International Conference on Language Resources and Evaluation, LREC'04, Paris, ELRA (2004) 1535 – 1538 http://nl.ijs.si/et/Bib/LREC04/.

4. Brown, P.F., Lai, J.C., Mercer, R.L.: Aligning sentences in parallel corpora. In: Proceedings of the 29th annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1991) 169–176
5. Chen, S.F.: Aligning sentences in bilingual corpora using lexical information. In: Proceedings of the 31st annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1993) 9–16
6. Moore, R.C.: Fast and accurate sentence alignment of bilingual corpora. In: AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users, London, UK, Springer-Verlag (2002) 135–144
7. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1993) 263–311
8. Knight, K.: A statistical machine translation tutorial workbook (1999) URL: http://www.isi.edu/natural-language/mt/wkbk.rtf.
9. Yarowsky, D.: Decision lists for lexical ambiguity resolution. In Hayes-Roth, B., Korf, R., eds.: Proceedings of the Twelfth National Conference on Artificial Intelligence, Menlo Park, CA, American Association for Artificial Intelligence, AAAI Press (1994)
10. Yarowsky, D., Florian, R.: Evaluating sense disambiguation across diverse parameter spaces. Natural Language Engineering **8** (2002) 293–310
11. Wang, X.: Robust utilization of context in word sense disambiguation. In Dey, A., Kokinov, B., Leake, D., Turner, R., eds.: Modeling and Using Context: 5th International and Interdisciplinary Conference. Springer-Verlag, Berlin (2005) 529–541
12. Ristad, E.S., Thomas, R.G.: New techniques for context modeling. In: ACL. (1995) 220–227
13. Biçici, E.: Local context selection for aligning sentences in parallel corpora. In: Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2007), LNAI. Volume 4635., Roskilde, Denmark (2007) 82–93
14. Zipf, G.K.: The meaning-frequency relationship of words. The Journal of General Psychology **33** (1945) 251–256
15. Treebank, P., Marcus, M.P., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank (2004)
16. Joachims, T.: Learning to Classify Text using Support Vector Machines. Kluwer Academic Publishers (2002)
17. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarity in the amino acid sequences of two proteins. J. Mol. Biol. **48** (1970) 443–453
18. Gusfield, D.: Algorithms on Strings, Trees, and Sequences. Cambridge University Press (1997)
19. Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., Nagy, V.: Parallel corpora for medium density languages. In: Proceedings of the Recent Advances in Natural Language Processing 2005 Conference, Borovets. Bulgaria (2005) 590–596 Comment: hunalign is available at http://mokk.bme.hu/resources/hunalign.