Local Context Selection for Aligning Sentences in Parallel Corpora

Ergun Biçici

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

Abstract. This paper presents 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 sentences and does not use any language dependent knowledge. We make use of the context of sentences and introduce the notion of Zipfian word vectors which effectively models the distributional properties of a given sentence. We accept the context to be the frame in which the reasoning about sentence alignment is done. We examine alternatives for local context models and demonstrate that our context based sentence alignment algorithm performs better than prominent sentence alignment techniques. Our system dynamically selects the local context for a pair of set of sentences which maximizes the correlation. 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 1.1951 to 1.5404 times better in reducing the error rate in alignment accuracy and coverage.

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 effects 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 [6]. 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 examine alternatives for local context models and present a system which dynamically selects the local context for a given sentence pair which maximizes the correlation for the pair in the given parallel corpora. The resulting learning methodology is language-independent; it handle non-monotonic and noisy alignments; it does not require any stemming, dictionaries, or anchors; 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 surrounding sentences. Therefore, we investigate the selection of context in relation to the performance increase in the sentence alignment task.

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 [9]. We used a publicly available and easily accessible dataset [5] 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 give some notation about sentence alignment, define Zipfian word vectors, present our feature representation, and discuss context in sentence alignment. We also present the properties of local context in sentence alignment and our sentence alignment algorithm in this section. In Sect. 5, we present the results of our experiments and the last section concludes.

2 Related Work

Brown *et. al.* [2] provide a statistical technique for sentence alignment using the number of word tokens in each sentence in addition to anchor points. The dataset they used (Canadian Hansards corpora [1]) contains comments that serve as anchor points. They define a bead as groupings of English and French sentences that have close lengths and an alignment as a sequence of beads. Gale and Church [6] 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.* [2] is that since there are more characters there is less uncertainty.

Both Brown *et. al.* and Gale and Church [6] assume that the corpus is divided into chunks and they ignore word identities. Chen [4] describes an algorithm that constructs

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

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 [10] sentence alignment model combines sentence-length-based and word-correspondence-based approaches, achieving high accuracy at a modest computational cost. Moore uses a modified version of the IBM Translation Model 1 [3]:

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

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 $P(S, T)$.

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 [16]. The dynamic nature of the context is noticed for the word sense disambiguation task by Yarowsky and Florian [17] 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 [15] 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 this work, we accept the context to be the frame in which the reasoning about sentence alignment is done. We examine alternatives for local context configurations and demonstrate that our context based sentence alignment algorithm performs better than prominent sentence alignment techniques.

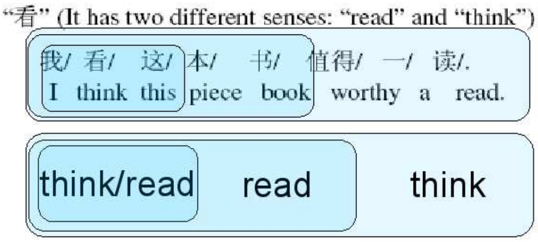
"看" (It has two different senses: "read" and "think")
我/ 看/ 这/ 本/ 书/ 值得/ 一/ 读/.
I think this piece book worthy a read.
think/read
read
think

Fig. 1. Word sense dependence on context

Previous 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. We also introduce 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 9 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 $|\text{corpus}|$ 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, $S_1 \leqslant_{\mathcal{S}} S_2 \ \Leftrightarrow\ \text{start}_{S_1} \leqslant_{\mathcal{S}} \text{start}_{S_2}$. 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" [18]. 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. Note that Zipfian word vectors contain information about anything that is recognized as a token after tokenization.

The TCat concept [8] 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

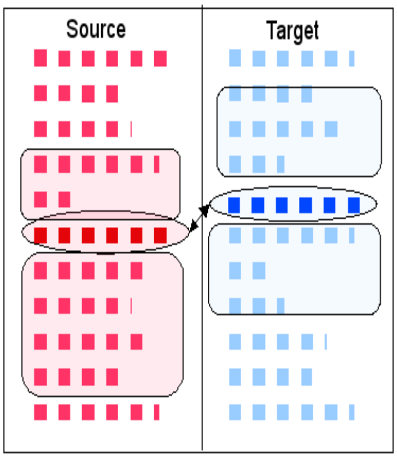

Fig. 2. Example Sentence Alignment Scenario

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. The sentences in a given corpus vary in content and size, therefore setting the local context to a specific value might not be effective. A sample scenario of sentence alignment is depicted in Fig. 2 where the sets of source and target sentences that are being compared are drawn in ellipses. The local context for the given source and target set of sentences in the figure can be represented as $(2, 4)$ and $(3, 3)$ respectively. Although the local context shows variance, for two sets of sentences to be judged as translations of each other, the total length for the source and target sets of sentences' local contexts should be the same. This is because the comparison is based on the value of the 2D weight decaying correlation coefficient score, which is retrieved by using the local contexts of each pair of set of sentences compared.

Let S be the set of sentences from the source corpus and let its local context be represented as (b_s, a_s) representing the number of sentences that come before and after S. Similarly, the local context for the corresponding set of sentences T in the target corpus can be represented as $(b_t, b_s + a_s - b_t)$. For a given context window size limit,

w, there can be w^3 such local context selections for the pair S and T. We call this the *full local context search*.

Given a set of local context configurations, $\mathcal{C}$, how are we going to make better decisions? There are three alternatives that we consider:

- Accept the *maximum* score attained from among $\mathcal{C}$ alternatives for the quality of the alignment and store the corresponding local contexts for observing what kind of local context configurations results in the highest scores.
- Accept the *average* score attained from among $\mathcal{C}$ alternatives for the quality of the alignment.
- Accept the *average of the top k* scores attained from among $\mathcal{C}$ alternatives for the quality of the alignment. We chose k to be 5 for our experiments. Evaluating according to the average of the best k results is a technique which is successfully used in discriminating the semantics among different word pairs [14, 1].

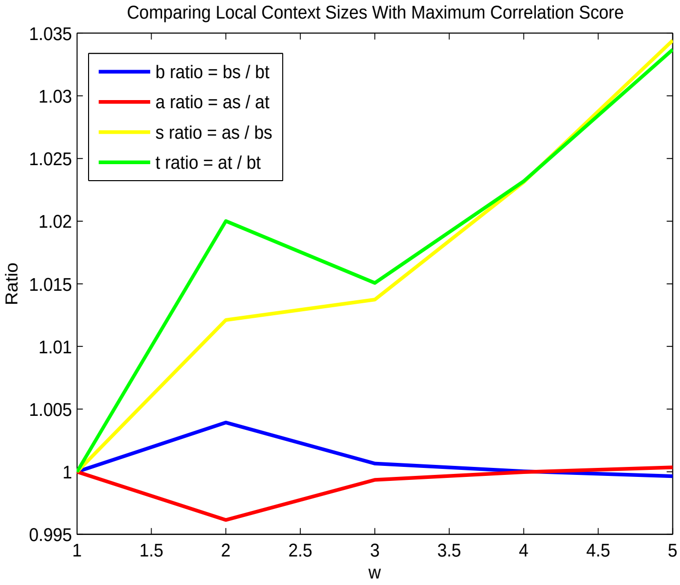

Fig. 3. Comparing local context sizes that return the maximum correlation score

Fig. 3 plots the resulting local context size ratios which result in the highest score after full local context search when we accept the maximum score to be the score for the given pair. These results were collected when the Lithuanian-English dataset was used for the full local context search. The actual average context sizes are listed in Table 1. One interesting observation from the results is that a_s and a_t are increasingly larger than b_s and b_t as we increase w. This is in line with the intuition that the sentences that come before have larger weight in determining the context. Another observation is that the corresponding local contexts for S and T get closer as we increase w. We can see that on the average their sizes are the same with ± 0.001 difference in their ratio. The first row of Table 1 shows that the ratio of exactly the same sized local contexts flattens to 63% as w is increased. So in nearly two thirds of the local context configurations, the local context sizes for S and T are exactly the same as well.

	w=1	w=2	w=3	w=4	w=5
Same Context Size Ratio	1	0.6950	0.6306	0.6302	0.6300
b_s	1	1.4256	1.8029	2.0422	2.2414
b_t	1	1.4200	1.8017	2.0422	2.2422
a_s	1	1.4429	1.8277	2.0895	2.3186
a_t	1	1.4484	1.8289	2.0895	2.3178
Static Context Accuracy	0.9082	0.9184	0.9184	0.92177	0.9116
Static Context Coverage	0.8892	0.8990	0.8990	0.9023	0.8909

Table 1. The ratio of exactly the same sized local contexts and average context sizes that attain maximum score with full local context search for the Lithuanian-English pair

Based on these observations, considering only those local contexts that are symmetric (i.e. $b_s = b_t$ and $a_s = a_t$) appears to be a feasible approach. For a given context window size limit, w, there can be w^2 such local context selections for the pair S and T. We call this the *symmetric local context search*. The resulting local context sizes for each language pair with the maximum score selection when w is chosen to be 4 is given in Table 2.

Dataset	**b**	**a**	**Increase**
Bulgarian	1.9564	1.9936	1.9%
Czech	1.9610	1.9961	1.8%
Estonian	1.9563	2.0129	2.9%
Hungarian	1.9723	1.9964	1.2%
Lithuanian	1.9720	2.0246	2.7%
Latvian	1.9667	2.0043	1.9%
Romanian	1.9275	1.9688	2.1%
Serbo-Croatian	1.9424	1.9755	1.7%
Slovene	1.9486	1.9765	1.4%

Table 2. Local symmetric context sizes per language - English pairs

The other option in context size selection is to set it to a static value for all comparisons. The last two rows in Table 1 show the accuracy and coverage performances when w is chosen globally for the whole dataset. Based on these results, we select w to be 4 in our experiments in which the context is static.

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 [11] with gap penalty set to -0.5. Further discussion on dynamic programming methodology to solve sentence alignment problems can be found in [6] or in [4]. 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 [7]). 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.

4 Experiments

We used the George Orwell's 1984 corpus's first chapter from Multext-East [5], which contains manually sentence split and aligned translations for English, Bulgarian, Czech, Estonian, Hungarian, Romanian, Slovene, Latvian, Lithuanian, and Serbo-Croatian. In all of our experiments, the target language pair is chosen to be English. We compared the results of our system with that of hunalign [13] and Moore's system [10]. Without an input dictionary, hunalign makes use of the Gale and Church [6] algorithm which is based on sentence lengths, and builds a dictionary dynamically based on this alignment.

Our first couple of experiments are based on choosing appropriate parameters. We chose to use the Lithuanian-English pair since its alignment types are more complex compared to the other datasets (6 different alignment types: 2-2, 2-1, 1-1, 1-3, 1-2, 1-6 with 1, 7, 274, 1, 10, 1 counts respectively). To reduce the complexity of calculations to a manageable value, the value of b is chosen to be 10.

Our results show that when we use static context and set the window size, w to 4, the algorithm makes 1 mistake out of every 23.16 sentence alignments and out of every 18.72 sentences. If we used the symmetric local context search with local context chosen as the maximum scoring one for the alignment, these numbers change to 22.22 and 17.88 respectively. When we use the average scoring for the symmetric local contexts, then these numbers become 24.40 and 19.64 respectively and become 23.97 and 19.10 respectively when we use the average of top 5 scores. These results were taken after observing 118, 123, 112, and 114 mistakes for the static, maximum, average, and average top 5 local context configuration schemes on the number of alignments and 299, 313, 285, and 293 mistakes for for the static, maximum, average, and average top 5 local context configuration schemes on the number of sentences. hunalign makes 1 mistake out of every 18.59 sentence alignments and out of every 13.26 sentences. Moore's algorithm makes 1 mistake out of every 16.27 sentence alignments and out of every 12.75 sentences. These results were taken after observing 147 mistakes for hunalign and 168 mistakes for Moore's algorithm on the number of alignments and 422 mistakes for hunalign and 439 mistakes for Moore's algorithm on the number of sentences. The total number of alignments is 2733 and sentences is 5596 in all of our data set.

4.1 Results on Sentence Alignment Accuracy

	Sentence Alignment Accuracy					
Language	**hunalign**	**Moore**	**static**	**maximum**	**average**	**average top** 5
Bulgarian	**96.74 / 3.26**	96.09 / 3.91	96.09 / 3.91	95.77 / 4.23	**96.74 / 3.26**	**96.74 / 3.26**
Czech	96.14 / 3.86	95.82 / 4.18	**96.78 / 3.22**	**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	99.04 / 0.96	98.39 / 1.61	99.04 / 0.96
Hungarian	87.86 / 12.14	88.96 / 11.04	92.98 / 7.02	91.30 / 8.70	**93.98 / 6.02**	91.64 / 8.36
Latvian	95.71 / 4.29	92.74 / 7.26	96.70 / 3.30	96.70 / 3.30	96.70 / 3.30	**97.03 / 2.97**
Lithuanian	88.44 / 11.56	82.31 / 17.69	92.18 / 7.82	**92.52 / 7.48**	91.84 / 8.16	**92.52 / 7.48**
Romanian	89.86 / 10.14	**95.27 / 4.73**	91.22 / 8.78	90.54 / 9.46	91.22 / 8.78	92.23 / 7.77
Serbo-Croatian	**98.70 / 1.30**	97.08 / 2.92	97.73 / 2.27	98.05 / 1.95	97.73 / 2.27	97.73 / 2.27
Slovene	97.70 / 2.30	97.04 / 2.96	98.68 / 1.32	98.36 / 1.64	**99.34 / 0.64**	98.36 / 1.64

Table 3. Sentence alignment accuracy per English - language alignments

In terms of sentence alignment accuracy, our context based sentence alignment algorithm with static, maximum, average, and average top 5 local context configuration schemes reduce the error rate of hunalign by 1.2458, 1.1951, 1.3125, and 1.2895 times and of Moore by 1.4237, 1.3659, 1.5000, and 1.4737 times respectively. The results represent the comparison in terms of the total number of errors made over all English-language alignments. The details can be seen in Table 3.

4.2 Results on Sentence Alignment Coverage

	Sentence Alignment Coverage					
Language	**hunalign**	**Moore**	**static**	**maximum**	**average**	**average top** 5
Bulgarian	95.34 / 4.66	94.86 / 5.14	95.18 / 4.82	94.86 / 5.14	**95.99 / 4.01**	**95.99 / 4.01**
Czech	94.92 / 5.08	95.24 / 4.76	**96.35 / 3.65**	**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	98.88 / 1.12	98.08 / 1.92	98.88 / 1.12
Hungarian	84.30 / 15.70	85.90 / 14.10	91.51 / 8.49	89.10 / 10.90	**92.63 / 7.37**	89.42 / 10.58
Latvian	92.65 / 7.35	90.26 / 9.74	95.37 / 4.63	95.37 / 4.63	95.37 / 4.63	**95.69 / 4.31**
Lithuanian	84.85 / 15.15	79.15 / 20.85	90.23 / 9.77	**90.72 / 9.28**	89.90 / 10.10	**90.72 / 9.28**
Romanian	86.79 / 13.21	**93.64 / 6.36**	89.72 / 10.28	89.07 / 10.93	89.72 / 10.28	90.86 / 9.14
Serbo-Croatian	**97.75 / 2.25**	96.46 / 3.54	97.27 / 2.73	97.59 / 2.41	97.27 / 2.73	97.27 / 2.73
Slovene	95.81 / 4.19	95.64 / 4.36	98.06 / 1.94	97.58 / 2.42	**98.71 / 1.29**	97.58 / 2.42

Table 4. Sentence alignment coverage per English - language alignments

In terms of sentence alignment coverage, our context based sentence alignment algorithm with static, maximum, average, and average top 5 local context configuration schemes reduce the error rate of hunalign by 1.4114, 1.3482, 1.4807, and 1.4403 times and of Moore by 1.4682, 1.4026, 1.5404, and 1.4983 times respectively. The results represent the comparison in terms of the total number of errors made over all English-language alignments. The details can be seen in Table 4.

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 sentences and does not use any language dependent knowledge. The resulting sentence alignment methodology is language-independent; it can handle non-monotonicity and noise in the alignments, it does not require any stemming, or anchors, 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 introduce the notion of Zipfian word vectors which effectively presents an order-free representation of the distributional properties of a given sentence. We define two dimensional weight decaying correlation for calculating the similarities between sentences.

We accept the context to be the frame in which the reasoning about sentence alignment is done. We examine alternatives for local context models and developed a system which dynamically selects the local context for a pair of set of sentences which maximizes the correlation. 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 1.1951 to 1.5404 times better in reducing the error rate in alignment accuracy and coverage. The addition of word translation probabilities and models of word order to our system might give us a better solution to the sentence alignment problem.

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 as well as Nokia for their generous support for our presence at Context 2007.

References

1. Ergun Bicici and Deniz Yuret. Clustering word pairs to answer analogy questions. In *Proceedings of the Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN '06)*, pages 277–284, Akyaka, Mugla, June 2006.
2. Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176, Morristown, NJ, USA, 1991. Association for Computational Linguistics.
3. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June 1993.
4. Stanley F. Chen. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1993. Association for Computational Linguistics.
5. Tomaž Erjavec. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Fourth International Conference on Language Resources and Evaluation, LREC'04*, pages 1535 – 1538, Paris, 2004. ELRA. http://nl.ijs.si/et/Bib/LREC04/.
6. William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.
7. Dan Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.
8. Thorsten Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers, 2002.
9. Joseph B. Kruskal. An overview of sequence comparison. In David Sankoff and Joseph B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 1–44. Addison-Wesley, 1983.
10. Robert C. Moore. 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*, pages 135–144, London, UK, 2002. Springer-Verlag.
11. Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarity in the amino acid sequences of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.
12. Eric Sven Ristad and Robert G. Thomas. New techniques for context modeling. In *ACL*, pages 220–227, 1995.
13. Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. The JRC-acquis: A multilingual aligned parallel corpus with 20+ languages. pages 2142–2147, 2006. Comment: hunalign is available at http://mokk.bme.hu/resources/hunalign.
14. Peter Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1136–1141, Aug 2005.

15. Xiaojie Wang. Robust utilization of context in word sense disambiguation. In Anind Dey, Boicho Kokinov, David Leake, and Roy Turner, editors, *Modeling and Using Context: 5th International and Interdisciplinary Conference*, pages 529–541. Springer-Verlag, Berlin, 2005.
16. David Yarowsky. Decision lists for lexical ambiguity resolution. In Barbara Hayes-Roth and Richard Korf, editors, *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA, 1994. American Association for Artificial Intelligence, AAAI Press.
17. David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310, 2002.
18. George Kingsley Zipf. The meaning-frequency relationship of words. *The Journal of General Psychology*, 33:251–256, 1945.