Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora

Fabienne Braune **Alexander Fraser**
Institute for Natural Language Processing
Universität Stuttgart
{braunefe,fraser}@ims.uni-stuttgart.de

Abstract

We address the problem of unsupervised and language-pair independent alignment of symmetrical and asymmetrical parallel corpora. Asymmetrical parallel corpora contain a large proportion of 1-to-0/0-to-1 and 1-to-many/many-to-1 sentence correspondences. We have developed a novel approach which is fast and allows us to achieve high accuracy in terms of F_1 for the alignment of both asymmetrical and symmetrical parallel corpora. The source code of our aligner and the test sets are freely available.

1 Introduction

Sentence alignment is the problem of, given a parallel text, finding a bipartite graph matching minimal groups of sentences in one language to their translated counterparts. Because sentences do not always align 1-to-1, the sentence alignment task is non-trivial.

The achievement of high accuracy with minimal consumption of computational resources is a common requirement for sentence alignment approaches. However, in order to be applicable to parallel corpora in any language without requiring a separate training set, a method for sentence-alignment should also work in an unsupervised fashion and be language pair independent. By "unsupervised", we denote methods that infer the alignment model directly from the data set to be aligned. Language pair independence refers to approaches that require no specific knowledge about the languages of the parallel texts to align.

We have developed an approach to unsupervised and language-pair independent sentence alignment which allows us to achieve high accuracy in terms of F_1 for the alignment of both symmetrical and asymmetrical parallel corpora. Due to the incorporation of a novel two-pass search procedure with pruning, our approach is acceptably fast. Compared with Moore's bilingual sentence aligner (Moore, 2002), we obtain an average F_1 of 98.38 on symmetrical parallel documents, while Moore's aligner achieves 94.06. On asymmetrical documents, our approach achieves 97.67 F_1 while Moore's aligner obtains 88.70. On average, our sentence aligner is only about 4 times slower than Moore's aligner.

This paper is organized as follows: previous work is described in section 2. In section 3, we present our approach. Finally, in section 4, we conduct an extensive evaluation, including a brief insight into the impact of our aligner on the overall performance of an MT system.

2 Related Work

Among approaches that are unsupervised and language independent, (Brown et al., 1991) and (Gale and Church, 1993) use sentence-length statistics in order to model the relationship between groups of sentences that are translations of each other. As shown in (Chen, 1993) the accuracy of sentence-length based methods decreases drastically when aligning texts containing small deletions or free translations. In contrast, our approach augments a sentence-length based model with lexical statistics and hence constantly provides high quality alignments.

(Moore, 2002) proposes a multi-pass search

Coling 2010: Poster Volume, pages 81–89,
Beijing, August 2010

procedure where sentence-length based statistics are used in order to extract the training data for the IBM Model-1 translation tables. The acquired lexical statistics are then combined with the sentence-length based model in order to extract 1-to-1 correspondences with high accuracy[1]. Moore's approach constantly achieves high precision, is robust to sequences of inserted and deleted text, and is fast. However, the obtained recall is at most equal to the proportion of 1-to-1 correspondences contained in the parallel text to align. This point is especially problematic when aligning asymmetrical parallel corpora. In contrast, our approach allows to extract 1-to-many/many-to-1 correspondences. Hence, we achieve high accuracy in terms of precision and recall on both symmetrical and asymmetrical documents. Moreover, because we use, in the last pass of our multi-pass method, a novel two-stage search procedure, our aligner also requires acceptably low computational resources.

(Deng et al., 2006) have developed a multi-pass method similar to (Moore, 2002) but where the last pass is composed of two alignment procedures: a standard dynamic programming (DP) search that allows one to find many-to-many alignments containing a large amount of sentences in each language and a divisive clustering algorithm that optimally refines those alignments through iterative binary splitting. This alignment method allows one to find, in addition to 1-to-1 correspondences, high quality 1-to-many/many-to-1 alignments. However, 1-to-0 and 0-to-1 correspondences are not modeled in this approach[2]. This leads to poor performance on parallel texts containing that type of correspondence. Furthermore performing an exhaustive DP search in order to find large size many-to-many alignments involves high computational costs. In comparison to (Deng et al., 2006), our approach works in the opposite way. Our two-step search procedure first finds a model-optimal alignment composed of the smallest possible correspondences, namely 1-to-0/0-to-1 and 1-to-1, and then merges those correspondences into larger alignments. This allows the finding of 1-to-0/0-to-1 alignments as well as high quality 1-to-many/many-to-1 alignments, leading to high accuracy on parallel texts but also on corpora containing large blocs of inserted or deleted text. Furthermore, our approach keeps the computational costs of the alignment procedure low: our aligner is, on average, about 550 times faster than our implementation[3] of (Deng et al., 2006).

Many other approaches to sentence-alignment are either supervised or language dependent. The approaches by (Chen, 1993), (Ceausu et al., 2006) or (Fattah et al., 2007) need manually aligned pairs of sentences in order to train the used alignment models. The approaches by (Wu, 1994), (Haruno and Yamazaki, 1996), (Ma, 2006) and (Gautam and Sinha, 2007) require an externally supplied bilingual lexicon. Similarly, the approaches by (Simard and Plamondon, 1998) or (Melamed, 2000) are language pair dependent insofar as they are based on cognates.

3 Two-Step Clustering Approach

We present here our two-step clustering approach to sentence alignment[4] which is the main contribution of this paper. We begin by giving the main ideas of our approach using an introductory example (section 3.1). Then we show to which extent computational costs are reduced in comparison to a standard DP search (section 3.2) before presenting the theoretical background of our approach (section 3.3). We further discuss a novel pruning strategy used within our approach (section 3.4). This pruning technique is another important contribution of this paper. Next, we present the alignment model (section 3.5) which is a slightly modified version of the alignment model used in (Moore, 2002). Finally, we describe the overall

[1] The used search heuristic is a forward-backward computation with a pruned dynamic programming procedure as the forward pass.

[2] In (Deng et al., 2006), p. 5, the $p(a_k) = p(x, y)$ which determines the prior probability of having an alignment containing x source and y target sentences, is equal to 0 if $x < 1$ or $y < 1$. As $p(a_k)$ is a multiplicative factor of the model, the probability of having an insertion or a deletion is always equal to 0.

[3] In order to provide a precise comparison between our aligner and (Deng et al., 2006), we have implemented their model into our optimized framework.

[4] Note that our approach does not aim to find many-to-many alignments. None of the unsupervised sentence alignment approaches discussed in section 2 are able to correctly find that type of correspondence.

procedure required to align a parallel text with our method (section 3.6).

3.1 Sketch of Approach

Consider a parallel text composed of six source language sentences F_i and four target language sentences E_j. Further assume that the correct alignment between the given texts is composed of four correspondences: three 1-to-1 alignments between F1, E1; F2, E2 and F6, E4 as well as a 3-to-1 alignment between F3, F4, F5 and E3. Figure 1 illustrates this alignment.

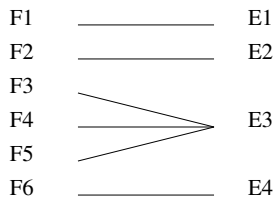

Figure 1: *Correct Alignment between F_i and E_j*

In the perspective of a statistical approach to sentence alignment, the alignment in figure 1 is found by computing the model-optimal alignment A^* for the bitext considered:

$$A^* = \underset{A}{\operatorname{argmax}} \prod_{a_k \in A} SCORE(a_k) \qquad (1)$$

where $SCORE(a_k)$ denotes the score attributed by the alignment model[5] to a minimal alignment a_k composing A^*. The optimization given in equation 1 relies on two commonly made assumptions: (c_1) a model-optimal alignment A^* can be decomposed into k minimal and independent alignments a_k; (c_2) each alignment a_k depends only on local portions of text in both languages.

The search for A^* is generally performed using a dynamic programming (DP) procedure over the space formed by the l source and m target sentences. The computation of A^* using a DP search relies on the assumption (c_3) that sentence alignment is a monotonic and continuous process. The DP procedure recursively computes the optimal score $D(l,m)^*$ for a sequence of alignments covering the whole parallel corpus. The optimal score $D(l,m)^*$ is given by the following recursion:

$$D(l,m)^* = \min_{0 \leq x,y \leq R \ , \ x=1 \vee y=1} D(l-x, m-y)^* - \log SCORE(a_k) \qquad (2)$$

where x denotes the number of sentences on the source language side of a_k and y the number of sentences on the target language side of a_k.

The constant R constitutes an upper bound to the number of sentences that are allowed on each side of a minimal alignment a_k. This constant has an important impact on the computational costs of the DP procedure insofar as it determines the number of minimal alignments that have to be compared and scored at each step of the recursion given in equation 2. As will be shown in section 3.2, the number of comparisons increases depending on R.

The solution we propose to the combinatorial growth of the number of performed operations consists of dividing the search for A^* into two steps. First, a model-optimal alignment A_1^*, in which the value of R is fixed to 1, is found. Second, the alignments a'_k composing A_1^* are merged into clusters m_r containing up to R sentences on either the source or target language side. The alignment composed of these clusters is A_R^*.

The search for the first alignment A_1^* is performed using a standard DP procedure as given in equation 2 but with $R = 1$. This first alignment is, hence, only composed of 0-to-1, 1-to-0 and 1-to-1 correspondences. Using our example, we show, in figure 2, the alignment A_1^* found in the first step of our approach. The neighbors of F4, that is F3 and F5, are aligned as 1-to-0 correspondences.

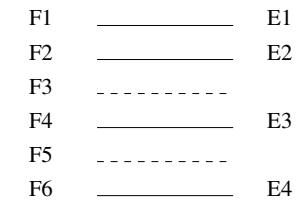

Figure 2: *A_1^* in our Approach (first step)*

The search for A_R^* is performed using a DP search over the alignments a'_k composing A_1^*. The score $D(A_R)^*$ obtained when all alignments $a'_k \in A_1^*$ have been optimally clustered can be written

[5]The alignment model will be presented in section 3.5.

recursively as:

$$D(A_R)^* = \min_{0 \leq r \leq R} D(A_R - r)^* - \log SCORE(m_r) \quad (3)$$

where $D(A_R - r)^*$ denotes the best score obtained for the prefix covering all minimal alignments in A_1^* except the last r minimal alignments considered for composing the last cluster m_r.

The application of the second step of our approach is illustrated in figure 3. The first alignment, between F1 and E1, cannot be merged to be part of a 1-to-many or many-to-1 cluster because the following alignment in A_1^* is also 1-to-1. So it must be retained as given in A_1^*. The five last alignments are, however, candidates for composing clusters. For instance, the alignment F2-E2 and F3-ϵ, where ϵ denotes the empty string, could be merged in order to compose the 2-to-1 cluster F2,F3-E2. However, in our example, the alignment model chooses to merge the alignments F3-ϵ, F4-E3 and F5-ϵ in order to compose the 3-to-1 cluster F3,F4,F5-E3.

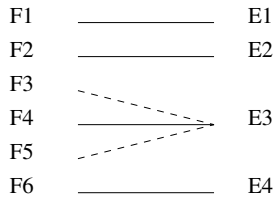

Figure 3: A_R^* *in our Approach (second step)*

3.2 Computational Gains

The aim of this section is to give an idea about why our method is faster than the standard DP approach. Let C denote the number of comparisons performed at each step of the recursion of the standard DP procedure, as given in equation 2. This amount is equivalent to the number of possible combinations of x source sentences with y target sentences. Hence, for an approach finding all types of correspondences except many-to-many, we have:

$$C = 2R + 1 \quad (4)$$

In terms of lookups in the word-correspondence tables of a model including lexical statistics, the number of operations C_l performed at each step of the recursion is given by:

$$C_l = R' * w^2 \quad (5)$$

where R' denotes the number of scored sentences[6]. w denotes the average length of each sentence in terms of words. The total number of lookups performed in order to align a parallel text containing l source and m target sentences using a standard DP procedure is hence given by:

$$L = R' * w^2 * l * m \quad (6)$$

In the perspective of our two-step search procedure, the computational costs of the search for the initial alignment A_1^* is given by:

$$L'_1 = w^2 * l * m \quad (7)$$

For the second step of our approach, because A_R^* is a cluster of A_1^*, the dynamic programming procedure used to find this alignment is no longer over the $l * m$ space formed by the source and target sentences but instead over the space formed by the minimal alignments a'_k in A_1^*. The average number of those alignments is approximately $\frac{l+m}{2}$.[7] The number of lookups performed at each step of our DP procedure is given by:

$$L'_2 = R' * w^2 * \frac{l+m}{2} \quad (8)$$

where R' and w are defined as in equation 6. The total number of lookups for our clustering approach is hence given by:

$$L'_{1+2} = (w^2 * l * m) + (R' * w^2 * \frac{l+m}{2}) \quad (9)$$

In order to compare the costs of our approach and a standard DP search over the $l * m$ space formed by the source and target sentences, we re-write equation 6 as:

$$L = (w^2 * l * m) + ((R' - 1) * w^2 * l * m) \quad (10)$$

The comparison of equation 9 with equation 10 shows that the computational gains obtained using our two-step approach reside in the reduction of the search space from $l * m$ to $\frac{l+m}{2}$.[8]

[6] In a framework where no caching of scores is performed, we have $R' = R^2 + R + 1$ compared sentences while score-caching allows one to reduce R' to R.

[7] Note that this amount tends to $l + m$ when A_1^* contains a large number of 0-to-1/1-to-0 correspondences.

[8] It should be noted that through efficient pruning, the search space of the standard (DP) procedure can be further reduced, see section 3.4.

3.3 Theoretical Background

We now present the theoretical foundation of our approach. First, we rewrite equation 1 in a more detailed fashion as:

$$A_R^* = \underset{A}{\operatorname{argmax}} \prod_{a_k(x_k,y_k)\in A_R} P(a_k(x_k,y_k), s_i^q, t_j^r) \tag{11}$$

with $0 \leq x_k, y_k \leq R$, where R denotes the maximal amounts x and y of source and target language sentences composing a minimal alignment $a_k(x_k, y_k)$. The distribution $P(a_k(x_k, y_k), s_i^q, t_j^r)$ specifies the alignment model presented in section 3.5.

As seen in section 3.1, the formulation of the alignment problem as given in equation 11 and the use of a DP search in order to solve this equation rely on the assumptions (c_1) to (c_3). Following these assumptions, a model-optimal alignment A_1^* can be defined as an ordered set of minimal alignments $a'_k(x_k, y_k)$, with $0 \leq x_k, y_k \leq 1$, where the aligned portions of text are sequential. In other words, if the $k-th$ alignment $a'_k(x_k, y_k)$ contains the sequences s_i^q and t_j^r of source and target language sentences, then the next alignment $a'_{k+1}(x_{k+1}, y_{k+1})$ is composed of the sequences s_{q+1}^u and t_{r+1}^v. Hence, each alignment composing A_R, with $R > 1$, can be obtained through sequential merging of a series of alignments $a'_k(x_k, y_k) \in A_1^*$.[9] Accordingly, the sequences of sentences s_1^u and t_1^v are obtained by merging s_1^q and t_1^r with s_{q+1}^u and t_{r+1}^v. It can then be assumed that (c_4) the ordered set of minimal alignments composing A_R^* under equation 11 is equivalent to the set of clusters obtained by sequentially merging the minimal alignments composing A_1^*. Following assumption (c_4), the optimization over $a_k(x_k, y_k) \in A_R$ is equivalent to an optimization over the merged alignments $m_r(x_r, y_r) \in A_R$. Hence, equation 11 is equivalent to:

$$A_R^* = \underset{A_R}{\operatorname{argmax}} \prod_{m_r(x_r,y_r)\in A_R} P(m_r(x_r,y_r), s_i^u, t_j^v) \tag{12}$$

where each $m_r(x_r, y_r)$ is obtained by merging r minimal alignments $a'_k(x_k, y_k) \in A_1^*$.

The computation of A_R^* is done in two steps. First, a model-optimal alignment A_1^* is found using a standard DP procedure as defined in equation 2 but with $R = 1$ and where $SCORE(a_k)$ is given by the alignment model $-\log P(a_k, s_{l-x+1}^l, t_{m-y+1}^m)$. In the second step, the search procedure used to find the optimal clusters is defined as in equation 3 but where $SCORE(m_r)$ is given by the alignment model $-\log P(m_r, s_i^u, t_j^v)$.

3.4 Search Space Pruning

In order to further reduce the costs of finding A_1^*, we initially pruned the search space in the same fashion as (Moore, 2002). We explored a narrow band around the main diagonal of the bitext to align. Each time the approximated alignment came close to the boundaries of the band, the search was reiterated with a larger band size. However, the computational costs for alignments that were not along the diagonal quickly increased with this pruning strategy. A high loss of efficiency was hence observed when aligning asymmetrical documents with this technique. Incidentally, Moore reports, in his experiments, that for the alignment of a parallel text containing 300 deleted sentences, the computational costs of his pruned DP procedure is 40 times higher than for a corpus containing no deletions.

In order to overcome this problem, we developed a pruning strategy that allows us to avoid the loss of efficiency occurring when aligning asymmetrical documents. Instead of exploring a narrow band around the main diagonal of the text to align, we use sentence-length statistics in order to compute an approximate path through the considered bitext. Our search procedure then explores the groups of sentences that are around this path. If the approximated alignment comes close to the boundaries of the band, the search is re-iterated.

The path initially provided using a sentence-length model[10] and then iteratively refined is closer to the correct alignment than the main diagonal of the bitext to align. Hence, the approximated alignment does not come close to the band

[9] Alignments of type 1-to-0/0-to-1 and 1-to-1 are assumed to be clusters where a minimal alignment $a'_k(x_k, y_k) \in A_1^*$ has been merged with the empty alignment $e_0(0,0)(\epsilon, \epsilon)$.

[10] The used model is the sentence-length based component of (Moore, 2002), which is able to find 1-to-0/0-to-1 correspondences.

as often as when searching around the main diagonal. This results in relatively high computational gains, especially for asymmetrical parallel texts (see section 4).

3.5 Moore's Alignment Model

The model we use is basically the same as in (Moore, 2002) but minor modifications have been made in order to integrate this model in our two-step clustering approach. The three component distributions of the model are given by[11]:

$$P(a_k, s_i^q, t_j^r) = P(a_k)P(s_i^q|a_k)P(t_j^r|a_k, s_i^q) \quad (13)$$

The first component, $P(a_k)$, specifies the generation of a minimal alignment a_k. The second component, $P(s_i^q|a_k)$, specifies the generation of a sequence s_i^q of source language sentences in a minimal alignment a_k. The last component, i.e. $P(t_j^r|a_k, s_i^q)$, specifies the generation of a sequence of target language sentences depending on a sequence of generated source sentences.

Our first modification to Moore's model concerns the component distribution $P(a_k)$. In the second pass of our two-step approach, which is the computation of the model-optimal clustered alignment A_R^*, we estimate $P(a_k)$ by computing the relative frequency of sequences of alignments a_k' in the initial alignment A_1^* that are candidates for composing a cluster m_r of specific size.[12] A second minimal modification to Moore's model concerns the lexical constituent of $P(t_j^r|a_k, s_i^q)$, which we denote here by $P(f_b|e_n, a_k)$. In contrast with Moore, we use the best alignment (Viterbi alignment) of each target word f_b with all source words e_n, according to IBM Model-1:

$$P(f_b|e_n, a_k) = \frac{\arg\max_{n=1}^{l_e} P_t(f_b|e_n)}{l_e + 1} \quad (14)$$

where l_e denotes the number of words in the source sentence(s) of a_k. Our experimental results have shown that this variant performed slightly better than Moore's summing over all alignments.

[11] In order to simplify the presentation of the model, we use the short notation a_k for denoting $a_k(x_k, y_k)$

[12] For the computation of A_1^*, the distribution $P(a_k)$ is defined as in Moore's work.

3.6 Alignment Procedure

In order to align a parallel text (s_1^l, t_1^m) we use a multi-pass procedure similar to (Moore, 2002) but where the last pass is replaced by our two-step clustering approach. In the first pass, an approximate alignment is computed using sentence-length based statistics and the one-to-one correspondences with likelihood higher than a given threshold are selected for the training of the IBM Model-1 translation tables[13]. Furthermore, each found alignment is cached in order to be used as the initial diagonal determining the search space for the next pass. In the second pass, the corpus is re-aligned according to our two-step approach: (i) a model-optimal[14] alignment containing at most one sentence on each side of the minimal alignments $a_k(x_k, y_k)$ is found; (ii) those alignments are model-optimally merged in order to obtain an alignment containing up to R sentences on each side of the clusters $m_r(x_r, y_r)$. In our experiments, a maximum number of 4 sentences is allowed on each side of a cluster.

4 Experiments

We evaluate our approach (CA) using three baselines against which we compare alignment quality and computational costs.[15] The first (Mo) is the method by (Moore, 2002). As a second baseline (Std), we have implemented an aligner that finds the same type of correspondences as our approach but performs a standard DP search instead of our two-pass clustering procedure and implements Moore's pruning strategy. Our third baseline (Std P.) is similar to (Std) but integrates our pruning technique.[16] We also evaluate the impact

[13] Words with frequency < 3 in the corpus have been dropped.

[14] This is optimal according to the alignment model which will be presented in section 3.5.

[15] We do not evaluate sentence-length based methods in our experiments because these methods obtain an F_1 which is generally about 10% lower than for our approach on symmetrical documents. For asymmetrical documents the performance is even worse. For example, when using Gale&Church F_1 sinks to 13.8 on documents which are not aligned at paragraph level and contain small deletions.

[16] We do not include (Deng et al., 2006) in our experiments because our implementation of this aligner is 550 times slower than our proposed method and the inability to find 1-to-0/0-to-1 correspondences makes it inappropriate for asymmetrical documents.

S	1-1	1-N/N-1	0-1/1-0	Oth.	Tot.
1	88.2%	10.9 %	0.005%	0.85%	3,877
2	91.9%	7.5%	0.007%	0.53%	2,646
3	91.6%	2.7%	4.3%	1.4%	23,715
4	44.8%	6.2%	49%	0.01%	2,606

Table 1: *Test Set for Evaluation with $2 \leq N \leq 4$*

of our aligner on the overall performance of an MT system.

Evaluation. We evaluate the alignment accuracy of our approach using four test sets annotated at sentence-level. The two first are composed of hand aligned documents from the Europarl corpus for the language-pairs German-to-English and French-to-English. The third is composed of an asymmetric document from the German-to-English part of the Europarl corpus. Our fourth test set is a version of the BAF corpus (Simard, 1998), where we corrected the tokenization. BAF is an interesting heterogeneous French-to-English test set composed of 11 texts belonging to four different genres. The types of correspondences composing our test sets are given in table 1. The metrics used are precision, recall and F_1[17]. Only alignments that correspond exactly to reference alignments count as correct. The computational costs required for each approach are measured in seconds. The time required to train IBM Model-1 is not included in our calculations[18].

Summary of Results. Regarding alignment accuracy, the results in table 2 show that (CA) obtains, on average, an F_1 that is 4.30 better than for (Mo) on symmetrical documents. The results in table 3 show that, on asymmetrical texts, (CA) achieves an F_1 which is 8.97 better than (Mo). The accuracy obtained using (CA), (Std) and (Std P.) is approximately the same. We have further compared the accuracy of (CA) with (Std) for finding 1-to-many/many-to-1 alignments. The obtained results show that (CA) achieves an F_1 that is 5.0 better than (Std).

Regarding computational costs, the time required by (CA) is on average 4 times larger than for (Mo) when aligning symmetrical documents. On asymmetrical documents, (Mo) is, however, only 1.5 times faster than (CA). Compared to (Std), (CA) is approximately 6 times faster on symmetrical and 80 times faster on asymmetrical documents. The time of (Std P.) is 3 times higher than for (CA) on symmetrical documents and 22 times higher on asymmetrical documents. This shows that, first, our pruning technique is more efficient than Moore's and, second, that the main increase in speed is due to the two step clustering approach.

Discussion. On the two first test sets, (Mo) achieves high precision while the obtained recall is limited by the number of correspondences that are not 1-to-1 (see table 1). Regarding (Std), (Std P.) and (CA), all aligners achieve high precision as well as high recall, leading to an F_1 which is over 98% for both documents. The computational costs of (CA) for the alignment of symmetrical documents are, on average, 4 times higher than (Mo), 6 times lower than (Std) and 3.5 times lower than (Std P.). On our third test set (Mo) achieves, with an F_1 of 88.70, relatively poor recall while the other aligners reach precision and recall values that are over 98%. Regarding the computational costs, (CA) is only 1.5 times slower than (Mo) on asymmetrical documents while it is 80 times faster than (Std) and about 22 times faster than (Std P.). On our fourth test set all evaluated aligners perform approximately the same than on Europarl. While (Mo) obtains, with 94.46, an F_1 which is the same as for Europarl, (CA) performs, with an F_1 of 97.67, about 1% worse than on Europarl. A slightly larger decrease of 1.6% is observed for (Std) which obtains 96.81 F_1. Note, however, that (CA), (Std) and (Std P.) still perform about 3% better than (Mo). Regarding computational costs, (CA) is 4 times slower than (Mo) and 40 times faster than (Std). The high difference in speed between our approach and (Std) is due to the fact that the BAF corpus contains texts of variable symmetry while (Std) shows a great speed decrease when aligning asymmetrical documents. Finally, we have compared the accuracy of (Std) and (CA) for the finding of 1-to-many/many-to-1 alignments containing at least 3 sentences on the "many"

[17] We measure precision, recall and F_1 on the 1-to-N/N-to-1 alignments, $N >= 1$, which means that we view insertions and deletions as "negative" decisions, like Moore.

[18] The reason for this decision is that our optimized framework trains the Model-1 translation tables far faster than Moore's bilingual sentence aligner.

Appr.	Lang.	Prec.	Rec.	F1	Speed
Mo	D-E	98.75	87.88	92.99	935s
Mo	F-E	98.97	91.56	95.12	1,661s
Std	D-E	98.42	98.57	98.49	24,152s
Std	F-E	98.45	98.83	98.64	35,041s
Std P.	D-E	98.37	98.49	98.43	13,387s
Std P.	F-E	98.41	98.78	98.60	21,848s
CA	D-E	98.25	98.70	98.47	3,461s
CA	F-E	98.00	98.60	98.30	6,978s

Table 2: *Performance on Europarl*

Appr.	Prec.	Rec.	F1	Speed
Mo	97.90	81.08	88.70	552s
Std	97.66	97.74	97.70	71,475s
Std P.	97.74	97.81	97.77	17,502s
CA	97.38	97.97	97.67	800s

Table 3: *Performance on asym. documents*

Appr.	Prec.	Rec.	F1	Speed
Mo	96.58	92.43	94.46	563s
Std	96.82	96.80	96.81	84,988s
CA	97.05	97.63	97.34	2,137s

Table 4: *Performance on BAF*

side. This experiment has shown that (Std) finds a larger amount of those alignments while making numerous wrong conjectures. On the other hand, (CA) finds less 1-to-many/many-to-1 correspondences but makes only few incorrect hypotheses. Hence, F_1 is about 5% better for (CA).

MT evaluation We also measured the impact of 1-to-N/N-to-1 alignments (which are not extracted by Moore) on MT. We used standard settings of the Moses toolkit, and the Europarl devtest2006 set as our test set. We ran MERT separately for each system. System (s1) was trained just on the 1-to-1 alignments extracted from the Europarl v3 corpus by our system while system (s2) was trained with all correspondences found. (s1) obtains a BLEU score of 0.2670 while (s2) obtains a BLEU score of 0.2703. Application of the pairwise bootstrap test (Koehn, 2004) shows that (s2) is significantly better than (s1).

5 Conclusion

We have addressed the problem of unsupervised and language-pair independent alignment of symmetrical and asymmetrical parallel corpora. We have developed a novel approach which is fast and allows us to achieve high accuracy in terms of F_1 for the alignment of bilingual corpora. Our method achieved high accuracy on symmetrical and asymmetrical parallel corpora, and we have shown that the 1-to-N/N-to-1 alignments extracted by our approach are useful. The source code of the aligner and the test sets are available at http://sourceforge.net/projects/gargantua .

6 Acknowledgements

The first author was partially supported by the Hasler Stiftung[19]. Support for both authors was provided by Deutsche Forschungsgemeinschaft grants Models of Morphosyntax for Statistical Machine Translation and SFB 732.

References

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *In Proceedings of 29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176.

Ceausu, Alexandru, Dan Stefanescu, and Dan Tufis. 2006. Acquis communautaire sentence alignment using support vector machines. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*.

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16.

Deng, Yoggang, Shankar Kumar, and William Byrne. 2006. Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, 12:1–26.

Fattah, Mohamed Abdel, David B. Bracewell, Fuji Ren, and Shingo Kuroiwa. 2007. Sentence alignment using p-nnt and gmm. *Computer Speech and Language*, (21):594–608.

Gale, William A. and Kenneth Ward Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Gautam, Mrityunjay and R. M. K. Sinha. 2007. A program for aligning sentences in bilingual corpora. *Proceedings of the International Conference*

[19] http://www.haslerstiftung.ch/.

on Computing: Theory and Applications, ICCTA '07, (1):480–484.

Haruno, M. and T. Yamazaki. 1996. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of ACL '96*, pages 131–138.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In Lin, Dekang and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Ma, Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*.

Melamed, I. Dan. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26:221–249.

Moore, Robert. 2002. Fast and accurate sentence alignment of bilingual corpora. In *In Proceedings of 5th Conference of the Association for Machine Translation in the Americas*, pages 135–244.

Simard, Michel and Pierre Plamondon. 1998. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13(1):59–80.

Simard, Michel. 1998. The baf: A corpus of english-french bitext. In *Proceedings of LREC 98*, Granada, Spain.

Wu, Dekai. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *In Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics, 80–87, Las*, pages 80–87.