MT-based Sentence Alignment for OCR-generated Parallel Texts

Rico Sennrich and **Martin Volk**
Insitute of Computational Linguistics
University of Zurich
Binzmühlestr. 14
CH-8050 Zürich
lastname@cl.uzh.ch

Abstract

The performance of current sentence alignment tools varies according to the to-be-aligned texts. We have found existing tools unsuitable for hard-to-align parallel texts and describe an alternative alignment algorithm. The basic idea is to use machine translations of a text and BLEU as a similarity score to find reliable alignments which are used as anchor points. The gaps between these anchor points are then filled using BLEU-based and length-based heuristics. We show that this approach outperforms state-of-the-art algorithms in our alignment task, and that this improvement in alignment quality translates into better SMT performance. Furthermore, we show that even length-based alignment algorithms profit from having a machine translation as a point of comparison.

1 Introduction

Sentence alignment is an important first step for Statistical Machine Translation. While existing algorithms report excellent performance (95% or more (Singh and Husain, 2005)), performance significantly decreases as the amount of noise or linguistic distance between the languages increases. An alignment between highly structured texts in similar languages is fairly easy; length-based approaches such as that by Gale and Church (1993) achieve high precision and recall scores and are used today to align the Europarl corpus and JRC-Acquis, among others.

If we work with pairs of fundamentally different languages and/or less structured texts, the alignment task becomes more difficult. Working with an OCR-generated parallel corpus of Swiss Alpine texts, we encountered a high proportion of noise such as misrecognized paragraph boundaries and 1-to-many alignments. Since we found the performance of existing sentence alignment tools unsatisfactory, we developed an alignment algorithm based on automatic translations of the to-be-aligned texts and BLEU as a similarity score. The approach requires an MT system with a reasonable performance for the language pair, but no other language-specific resources. It is thus fairly language-independent, and as we will show, more robust to textual noise than other approaches.

2 Related Work

Overviews of sentence alignment algorithms are provided in (Manning and Schütze, 1999; Singh and Husain, 2005). Most widespread methods are based on a comparison of sentence length, lexical correspondences, or a combination of the two.

Length-based algorithms have first been proposed by Brown, Lai and Mercer (1991) (word count), and Gale and Church (1993) (character count). The Gale and Church algorithm is still widely used today, for instance to align Europarl (Koehn, 2005).

Kay and Röscheisen (1993) introduce an alignment algorithm based on word correspondences. Chen (1993) constructs a word-to-word translation model during alignment, using it to estimate the probability of an alignment.

Moore (2002) and Varga et al. (2005) describe a two-pass algorithm, using a length-based approach for a first alignment. The first alignment subsequently serves as training data for a translation

model, which is then used in a complex similarity score. The two approaches differ in that Varga et al. (2005) use a dictionary-based translation model, with a dictionary that can be manually expanded, while Moore (2002) works with a IBM-1 translation model, following Chen (1993).

Different degrees of textual and metatextual structure open different possibilities for sentence alignment. Tiedemann (2007) shows that movie subtitles are a highly attractive text type for sentence alignment because the texts can be aligned on the basis of time stamps. Sentence alignment is also performed on comparable corpora for which no parallel structure can be assumed (Fung and Cheung, 2004; Adafre and de Rijke, 2006; Yasuda and Sumita, 2008). Adafre and de Rijke (2006) describe an MT-based approach to find corresponding sentences in Wikipedia based on sentence similarity. Our approach is based on the same basic idea of first automatically translating one of the to-be-aligned language portions, and then measuring the similarity between this translation and the other language portion.

3 The Parallel Corpus

Our sentence aligner has been developed to align the parallel part of the Text+Berg corpus, a corpus consisting of the yearbooks of the Swiss Alpine Club from 1864–1982[1] (Volk et al., 2010). Since 1957, the yearbooks are published in both French and German, with most articles being translated between the two languages. Currently, the parallel part of the corpus spans 138 000 sentences with 2.3/2.6 million tokens (German and French, respectively). Some examples from the 1911 yearbook illustrate the diversity. There are the typical reports on mountain expeditions: "Klettereien in der Gruppe der Engelhörner" (English: Climbing in the Engelhörner group) or "Aus den Hochregionen des Kaukasus" (English: From the high regions of the Caucasus). But the 1911 book also contains scientific articles on the development of caves ("Über die Entstehung der Beaten- und Balmfluhhöhlen") and on the periodic variations of the Swiss glaciers ("Les variations périodiques des glaciers des Alpes suisses").

We want to exploit the parallelism of the two language portions, and not resort to algorithms for comparable corpora with little to no structural parallels between the texts to be aligned. In this, we differ from the approach by Adafre and de Rijke (2006), who extract the best-scoring sentence pairs regardless of their position in the texts. We suspect that establishing an alignment between sentences that are not translated well with current SMT systems is especially valuable. We can quickly illustrate this with examples 1–3:

1. (DE) Also zurück zum Basislager.
   (So [we went] back to the base camp.)
2. (FR) Alors, retour au camp de base.
3. (MT) donc, basislager.

The fact that Basislager is an unknown word to our MT system means that we are especially interested in adding the translation pair (Basislager|camp de base) to our knowledge base. At the same time, data sparseness leads to a bad translation, which makes the automatic translation and the French sentence dissimilar on a word level.

While the Text+Berg corpus is more structured than comparable corpora, structural properties that facilitate sentence alignment in other parallel corpora are missing or unreliable in the Text+Berg corpus because of errors in the digitisation process. In less noisy parallel texts, it is possible to define paragraph boundaries or section boundaries as anchor points. Some texts even contain metatextual information such as speaker information or time stamps (e.g. in Parliament Proceedings and subtitles, respectively). However, out of the 7 articles used for our evaluation, only one has the same number of paragraphs in both language versions. In one article, the discrepancy amounts to 50% (39 vs. 64) because the French author separated utterances by different speakers with paragraph boundaries, while the German translator did not. Smaller differences in the number of paragraphs are the result of the automatic digitisation process.

Furthermore, section boundaries are not retained in the digital corpus, which means that the only reliable anchor points that we can establish beforehand are article boundaries. We manually created tables of contents for all the books we digitised for

[1] The publications after 1982 are currently being digitised.

the Text+Berg corpus, identifying both the author(s) and language of the articles. These tables of contents are then used to automatically establish article alignment on the basis of word overlap in the titles, author names, and the position of the articles in the respective yearbooks. In rare cases, article boundaries are not found, meaning that we may pass two concatenated articles to the sentence alignment algorithm.

Translators may decide to translate a single sentence as two, or to join two sentences in their translation. Manning and Schütze (1999) report that typically, approximatively 90% of sentence alignments (beads) are 1-to-1, the remaining 10% being 1-to-many beads because translators sometimes join or break up sentences. Additional 1-to-many beads are introduced in our corpus by sentence boundaries being misrecognized because of OCR or tokenisation errors.

Sentence alignment is further complicated by image captions, footnotes or advertisements that are not marked as such, and consequently considered part of the running text of the article. These text fragments typically occur at different positions in the two language versions, or only in one of them. They can be very disruptive to sentence alignment algorithms if they are not correctly recognized as deletions (1-to-0 or 0-to-1 beads), since a misalignment may cause consecutive sentences to be misaligned as well.

We have manually aligned a set of 1000 sentences, spanning 7 articles, we report the low number of 74% 1-to-1 beads. We observe 9% 2-to-1 beads[2], 6.9% 1-to-2 beads, and 6.3% 1-to-0 or 0-to-1 beads. The remaining 4% are n-to-m beads of higher order. Given that complex matches are more difficult[3], these numbers indicate that our corpus is harder to align than Europarl, for instance.

4 Algorithm

We describe an algorithm that computes a sentence alignment for a parallel text. The input for the algorithm are two sets of documents, articles or paragraphs separated by hard delimiters, and any number of translations. The choice and number of hard delimiters depends on the to-be-aligned texts, but they need to be reliable, since the algorithm does not search for alignments crossing these delimiters. The algorithm has been developed for articles spanning several dozen pages (500 or more sentences), and is sufficiently fast for these text lengths, but because of its quadratic complexity, it is not suited to process entire text collections without the use of hard delimiters.

The main alignment algorithm is computed for every text segment between two hard delimiters (including the beginning and end of file), and is composed of two steps. First, a set of anchor points is identified using BLEU as a similarity score between the translated source text and the target text. In a second step, the sentences between these anchor points are either aligned using BLEU-based heuristics or the length-based algorithm by Gale and Church.

4.1 Using BLEU as Similarity Score

BLEU has been developed as an automatic means to measure the translation quality of MT systems by comparing the system translation with one or more reference translations (Papineni et al., 2002). This is done by measuring the token n-gram precision of the system translation (hypothesis) for all n-gram levels up to 4, and combining the n-gram precisions using the geometric mean. The score of hypotheses shorter than the reference is reduced by a brevity penalty.

BLEU has been criticised as a measure of translation quality, and it is not considered reliable on a sentence level (Callison-Burch and Osborne, 2006). On the other hand, judging the quality of a translation is a much harder task than deciding whether two sentences are possible translations of each other. We found that BLEU is very sensitive to misalignments, often yielding a score of 0 if two unrelated sentences are compared, which means that it is capable of discriminating between aligned and unaligned sentence pairs.

Still, we made a number of modifications to the scoring implementation to fit it to our needs. Usually, BLEU is measured on up to 4-grams, motivated by the fact that n-gram scores of higher order are a good measure of a translation's fluency, while

[2]For all numbers, German is considered the source language, French the target language. In fact, 2 of the 7 articles are originally French, 5 German.

[3]Gale and Church (1993) report an error rate of 2.6% for 1-to-1 beads, 14% for 2-to-1 ones, and 100% for deletions (1-to-0).

unigrams measure their adequacy (Papineni et al., 2002). We found 4-grams problematic because too many translations would receive a score of 0, partly because of the low performance of MT systems on our test set. For our task, 2-grams yielded better results. We also decided to have both the hypothesis and reference sentence tokenized and lowercased, since case and tokenization, while they do affect the quality of a translation, are of little importance for our purposes.

Also, BLEU scores are asymmetrical for all sentence pairs that differ in length. If we compare two sentences s and t, we get different BLEU scores depending on whether we select s or t as our hypothesis because of the brevity penalty. We decided to do both and use the harmonic mean as our final score.

4.2 Pruning

We can visualize alignments as elements of a $|S|$-by-$|T|$ matrix, every row representing a source sentence, every column a target sentence. In the trivial case, the alignment between two texts is (s_1, t_1), (s_2, t_2), ..., (s_n, t_n). In other words, the alignment falls along the main diagonal of the matrix.

Naively, any sentence in the source text can be aligned to any sentence in the target text, but it is possible to reduce the search space by making assumptions about the position of alignments. Typically, the search is pruned to a band around the main diagonal of the search matrix (e.g. (Kay and Röscheisen, 1993; Moore, 2002)). Since our algorithm has been developed with hard-to-align corpora in mind, we do not prune the search space for our initial BLEU comparison. However, we prune aggressively after calculating BLEU scores, only keeping the 3 best-scoring alignment candidates for each sentence. This keeps the number of vertices in the best-path search linear to the number of sentences, and the search itself quadratic at worst.

4.3 Establishing Anchor Points Using BLEU

In a first pass, the algorithm tries to find a set of 1-to-1 beads considered reliable based on BLEU scores and sentence order. Given the sets of sentences T and T', T being the target text and T' being the automatic translation of the source text[4], the BLEU score is calculated for all members of the cartesian product of T and T'. This results in a list of possible alignment pairs, as shown in table 1 (only the best-scoring candidates are shown). This is the costliest step in the alignment algorithm, with a complexity of O($|T| * |T'|$).

(t'_8, t_8)	(t'_{12}, t_{14})	(t'_{15}, t_{275})	(t'_{19}, t_{231})
(t'_9, t_9)	(t'_{13}, t_{19})	(t'_{16}, t_{16})	(t'_{21}, t_{18})
(t'_{11}, t_{13})	(t'_{14}, t_{15})	(t'_{18}, t_{14})	(t'_{23}, t_{20})

Table 1: Alignment pairs between two texts as identified by BLEU. Sentences identified by their index.

Looking at table 1, one can easily see that some pairs are inconsistent with their neighbours, one of them being (t'_{15},t_{275}). If t'_{14} is aligned with t_{15} and t'_{16} with t_{16}, we do not expect t'_{15} to be aligned with t_{275}. A manual investigation of some development data confirms this expectation: as long as we are dealing with true parallel texts, and not with comparable corpora as in (Adafre and de Rijke, 2006), crossing alignments are a rare exception. What is far more common are sentences that cannot be correctly aligned one-to-one because of differences in length (see table 2 for the sentences in question). We can hence safely disallow crossing alignments. In other words, we only allow beads which can be ordered monotonically according to both the position of t and t' in their respective texts.

We use dynamic programming to search for the path of beads that maximizes the BLEU score while at the same time retaining monotonic sentence order[5]. The list of possible beads obtained in the first step forms a directed acyclic graph. We define all beads to be vertices, and edges to exist between any two vertices v and w if both sentences in w occur later than those in v in their respective texts. The weight of each edge is defined as the negative BLEU score of the successor vertex. Any shortest-path finding algorithm for weighted directed acyclic graphs can be used, the most efficient one relying on a topological sort and having the complexity O($|V|$+$|E|$), $|V|$ being the number of vertices, $|E|$ the number of edges. Since there are at worst

[4] We call German the source text and French the target text in this paper, irrespective of which articles are originally French, and which German.

[5] Alternatively, we tried searching for the longest valid path, which led to similar results because of our aggressive pruning.

s_{15} t'_{15}	Ein solcher Gipfel war für mich das Nadelhorn . un tel sommet était pour moi le nadelhorn .
t_{16} (correct, BLEU = 0.002)	Tel fut pour moi le Nadelhorn , mais après avoir personnifié la montagne comme ne peut manquer de le faire celui qui s' est souvent mesuré à elle , je m' em de dire que la cause principale des avatars qui vont être rapportés doit être recherchée dans le comportement humain , ce que démontreront à l' évidence les propos suivants .
t_{275} (wrong, BLEU = 0.236)	Pour moi l' affaire était dans le sac :

Table 2: Example of a BLEU misalignment. BLEU bigram scores between t'_{15} and t_i in both directions given.

$\frac{|V|-1}{2} * |V|$ edges conforming to the definition given above, the complexity of the algorithm can be simplified to O($|V|^2$). The result of the algorithm is an ordered list of 1-to-1 alignments, while a number of the sentences in both T and T' remain unaligned.

4.4 Processing the Gaps

We can treat the beads obtained in the previous step as anchor points (or hard delimiters) for any other alignment algorithm. For all pairs of 1-to-1 beads (t'_i,t_j),(t'_k,t_l) in the graph resulting from the previous step, we extract all unaligned sentences t'_x and t_y for which $i < x < k$ and $j < y < l$. Similarly, we extract all unaligned sentences before the first and after the last vertex in the graph.

The resulting sets of unaligned sentences t'_x and t_y, subsequently referred to as gaps, are aligned using a number of heuristics.

4.4.1 1-to-n Alignment Heuristic

As a first step, we try to find if one of the 1-to-1 beads found is in fact part of a 1-to-n or n-to-1 alignment, using the following procedure.

- From the list of t'_i, all t'_x for which $i < x < k$ and t'_k, create a list of all possible 1-, 2- or 3-sentence sequences.
- Do the same for t_j, all t_y for which $j < y < l$ and t_l.
- Calculate the BLEU score for the cartesian product of the two lists.
- If any bead ((t'_i,t'_{i+1},...,t'_{i+n}),t_j) scores higher than (t'_i,t_j), replace (t'_i,t_j) with ((t'_i,t'_{i+1},...,t'_{i+n}),t_j) in the graph. If not, do analogous checks for t_j, t'_k and t_l.
- If a 1-to-n bead is found, repeat the previous step.
- Else, proceed to the next heuristic.

In order to minimize the number of false positives based on a reduction of the brevity penalty which is part of BLEU, we define "scores higher than" as both an increase in BLEU score and in the absolute number of matches between the two sentence sequences.

4.4.2 1-to-1 Alignment Heuristic

The bead (t'_{i+1},t_{j+1}) is added to the graph if it is the best scoring pair out of all (t'_{i+1},t_y) in the gap. Especially in larger texts, it is possible that (t'_{i+1},t_{j+1}) was pruned after scoring and thus was not considered in the path-finding algorithm. If this heuristic is successful, the previous heuristic is repeated. If not, the algorithm proceeds to the next heuristic.

4.4.3 Gale and Church

If the gap reaches size 0 on either language side, or if its size is asymmetrical by a factor larger than two (except for two or three-word gaps), the remaining sentences in the gap are left unaligned.

Otherwise, the length-based alignment procedure based on Gale and Church (1993) establishes an alignment between all remaining sentences in the gap. It is worth noting that the algorithm is not given the source and target sentences as input, but the translations and target sentences. This gives slightly better results, and should be more robust for unrelated language pairs, for which a length-based comparison is less suited.

4.5 Combining Several Alignments

The algorithm can be run in two directions: we can either establish an alignment between T and T' or between S and S' (T' and S' being translations of S and T, respectively).[6] In order to obtain a high-precision alignment between two texts, we calculate an alignment in both directions and intersect the results, discarding all beads that differ between the two runs. Of course, we can also calculate an alignment in the same direction multiple times, using different automatic translations.

5 Sentence Alignment Performance

5.1 Method

We selected two existing sentence aligners as our baselines: an implementation of the algorithm by Gale and Church (1993), and the Microsoft Bilingual Sentence Aligner, which implements the algorithm by Moore (2002). The former was selected because it is still one of the most widely used sentence alignment algorithms, the latter because it produced the best results in the evaluation conducted by Singh and Husain (2005). The Microsoft Bilingual Sentence Aligner by default only returns 1-to-1 alignments. We modified its source code to return all alignments found, and will report both results.

Our algorithm described above was developed on a Text+Berg article of 400 sentences, half of which was hand-aligned to measure performance. We will subsequently call our Python implementation of the algorithm bleualign.

For the evaluation, we hand-aligned a different set of 1000 sentences, spanning 7 Text+Berg articles.[7] Given the high proportion of complex alignments in our evaluation set (see section 3), the alignment task is harder than in other evaluations which only consider 1-to-1 alignments (e.g. (Singh and Husain, 2005)). We expect most algorithms to align the two sentence pairs in table 3 as two 1-to-1 beads, where, in fact, they form a 2-to-2 bead.[8] Since intuitively, this is not completely wrong, we will report precision and recall measures using a *strict* and a *lax* truth condition. For a strict true positive, both the alignment hypothesis and the reference alignment need to be identical. In the lax condition, an alignment is considered true if there is an overlap on both language sides between the hypothesis and the reference. If a 2-to-2 bead is analysed as two 1-to-1 beads, this results in a lax precision of $\frac{2}{2}$ and lax recall of $\frac{1}{1}$. In the strict condition, precision is $\frac{0}{2}$ and recall $\frac{0}{1}$.

5.2 Results

Table 4 shows the sentence alignment scores achieved by different systems.[9] Looking at the two experiments running Gale and Church's algorithm, we see that a sentence-length comparison between the target text T and an automatic translation of the source text T' produces better scores than one between the source text S and T, even for the two not-too-distant languages French and German. As expected, the baseline is comparatively low, with F_1 scores of 0.68 (strict) and 0.80 (lax).

We found that Moore's algorithm clearly beats the Gale and Church algorithm in precision. In contrast to Gale and Church's algorithm, the performance does not increase when aligning between T' and T rather than S and T. Since the Moore algorithm produces a translation model after the initial sentence alignment pass, we expect it to perform better for larger text collections. This is indeed the case. The best system using Moore was run on the full Text+Berg parallel corpus and obtained an F-score of 0.78 (strict) / 0.87 (lax). The threshold for retaining sentence alignments was left at the default value (0.5); modifying it results in a slight precision-recall-tradeoff.

Bleualign performance depends heavily on the translation provided. Without any translation, that is if the algorithm computes sentence similarity directly between the target text and the source text, it performs worse than Gale and Church on our evaluation set. This is because only few sentence alignments are found using BLEU, some of them being wrong (e.g. 3 out of 7 wrong in a 75-sentence text).

[6] For texts rich in names and numbers that are unchanged between the two languages, we can even directly use BLEU on S and T, without the need for any translations.

[7] The exact number of sentences is 1011 and 991 for the French and German version, respectively.

[8] The first German sentence corresponds to one-and-a-half French sentences. Unless we try to align on a sub-sentence level, a 2-to-2 bead is the lowest possible correct sentence-level alignment.

[9] For a description of the MT systems used, see table 5.

Die Pause ist um , und ich muss wieder etwas tun : Den nächsten Versuch wagen .	La pause est terminée . Agir de nouveau , risquer l' essai suivant .

Table 3: Example of a 2-to-2 alignment

Alignment Algorithm	MT system used	strict			lax		
		P	R	F_1	P	R	F_1
Gale and Church	none	0.67	0.68	0.68	0.79	0.80	0.80
Gale and Church	google de-fr	0.71	0.72	0.72	0.83	0.83	0.83
Moore (eval set, 1-1)	none	0.84	0.60	0.70	0.93	0.67	0.78
Moore (eval set, 1-1)	google de-fr	0.83	0.58	0.68	0.90	0.63	0.74
Moore (full T+B, 1-1)	none	0.88	0.66	0.75	0.96	0.72	0.82
Moore (full T+B, all)	none	0.86	0.71	0.78	0.96	0.80	0.87
bleualign	none	0.57	0.44	0.50	0.71	0.54	0.61
bleualign	europarl-K de-fr	0.72	0.60	0.66	0.91	0.77	0.83
bleualign	europarl-M de-fr	0.83	0.78	0.81	0.98	0.92	0.95
bleualign	google de-fr	0.83	0.78	0.81	0.98	0.92	0.95
bleualign	europarl-M fr-de	0.80	0.76	0.78	0.96	0.90	0.93
bleualign (intersected)	europarl-M	0.92	0.69	0.78	0.99	0.73	0.84
bleualign (intersected)	europarl-M & google	0.95	0.60	0.73	0.99	0.62	0.77

Table 4: Sentence alignment precision and recall scores on a 1000-sentences evaluation set DE-FR.

Clearly, these alignment pairs were too unreliable to be used as anchor points.

With automatically translated text, the performance of bleualign increases, reaching precision levels similar to those of Moore with a superior recall. The best system using a single translation achieves an F_1 score of 0.81 (strict) / 0.95 (lax), compared to 0.75 / 0.82 achieved by Moore's system.

Intersecting the alignment obtained by running the tool with translations in both directions results in a strong precision-gain at a cost in recall. The intersected systems are the highest-precision ones in our evaluation. Running the algorithm with four different translations and intersecting the results led to a strict precision of 0.95, although with a low recall of 0.60. Lax precision reached 0.99, which means that almost no entirely unrelated sentence pairs are aligned by mistake.

In terms of speed, we found runtime differences between the alignment tools to be small. The computationally costliest step in our approach is by far the machine translation of the text. However, the computational complexity of machine translation is linear to the number of sentences[10], while the sentence alignment algorithm has the complexity of O($|S|$*$|T|$), $|S|$ and $|T|$ being the number of sentences in the source and target language. This means that aligning large text collections without using any hard delimiters may significantly slow down the algorithm.

In the evaluation set, the two language sides have about the same number of sentences. Since we observed that length-based algorithms perform far worse in our development set, the French version having 20% more sentences than the German one, we performed a second analysis on an especially hard-to-align parallel article. The article in question spans 77 sentences in the German version, and 281 in the French one. A closer look reveals that the article boundary at the end of the French article is not recognized, which means that the German article is erroneously aligned to two French ones. For this second evaluation, we did not work with a gold standard, but manually evaluated the sentence pairs returned by the alignment tools. The Gale and Church algorithm returns 77 beads, none of which are cor-

[10]Depending on the MT system, factors influencing translation speed include model size and sentence length.

MT system	Corpus	Alignment	TM (units)	LM (sent.)	MERT
DE-FR					
europarl-K	Europarl	Gale and Church	1000	1 270 000	Europarl
europarl-M	Europarl	Gale and Church	1 050 000	1 270 000	Europarl
T+B galechurch	Text+Berg	Gale and Church	102 000	138 000	Text+Berg
T+B moore	Text+Berg	Moore (all alignments)	45 000	138 000	Text+Berg
T+B bleu-single	Text+Berg	bleualign [europarl-M de-fr]	97 000	138 000	Text+Berg
T+B bleu-intersect	Text+Berg	bleualign [europarl-M]	78 000	138 000	Text+Berg
FR-DE					
europarl-M	Europarl	Gale and Church	1 050 000	1 260 000	Europarl
google fr-de / de-fr	(Google Translate: `http://translate.google.com`)				

Table 5: MT systems used in this evaluation.

rect. The algorithm by Moore returns 0 beads in all tested configurations.

Bleualign, using Europarl-M de-fr (trained on one million sentences) to translate the German article, returns 58 sentence pairs, with a precision of 60% (strict) or 79% (lax). Since the sentence pair (t'_{73},t_{73}) is recognized as an alignment point using BLEU comparison, only one German sentence (s'_{75}) is misaligned to the second French article. A set of 8 consecutive wrong beads, aligned by our gap filler algorithm (i.e. Gale and Church), is the reason for the comparatively low performance. Such a grouping of misaligned sentences is typical, since alignment errors often cause consecutive sentences to be misaligned as well. The only reason why the error was limited to 8 sentences is a BLEU-based anchor point which restores synchronicity.

6 SMT Performance

A handful of questions remain: we have evaluated how alignment differs for different translations, but we have not yet presented an evaluation of the quality of these MT systems on a Text+Berg test set. Since we expect the quality of the sentence alignment to correlate with the quality of the MT systems used, this should also give an indication as to how good translations need to be for this approach to work. A second question is how SMT performance scales with sentence alignment performance, and whether high-precision or high-recall alignment is preferable.

6.1 Method

Except for the Google translations, which were obtained online through Google Translate, all systems were trained using the instructions for building a baseline system by the 2010 ACL Workshop on SMT.[11] The SMT systems are built using Moses (Koehn et al., 2007), SRILM (Stolcke, 2002), and GIZA++ (Och and Ney, 2003). The systems we trained are different from the baseline system described above in that we used various training sets, and did not train a recaser, using a BLEU implementation that ignores case differences.

Table 5 describes the training data used for the different MT systems. Translation model (TM) training data is a subset of language model (LM) training data: only sentences with a valid alignment and fewer than 40 words in length are used to train the translation model. Europarl-K uses a sample of 1000 (1K) units of training data; all other systems use all the data available for TM training. The development sets used for tuning (Minimum Error Rate Training; MERT) are held out from training, as is the Text+Berg test set which consists of the same 1000 hand-aligned sentences that are also used for the sentence alignment evaluation. The size of language model is given in sentences, that of the TM training data and the development sets in units. Typically, one unit corresponds to one sentence, but in case of 1-to-many beads, a unit consists of several sentences in one language portion.

[11] `http://www.statmt.org/wmt10/baseline.html`

MT system	BLEU
untranslated	1.83
europarl-M fr-de	7.02
google fr-de	8.97
europarl-K de-fr	3.87
europarl-M de-fr	9.93
google de-fr	12.74
T+B galechurch de-fr	12.60
T+B moore de-fr	12.55
T+B bleu-single de-fr	13.41
T+B bleu-intersect de-fr	13.40

Table 6: BLEU scores of different MT systems.

6.2 Results

Table 6 shows the performance of various MT systems on our Text+Berg test set. We see that translation quality on the test set is very low for all systems trained on out-of-domain data.[12] It is a pleasant surprise that bleualign produces good alignment results based on these poorly performing SMT systems. These results are encouraging because they imply that our alignment tool is not only useful for texts that are already translated well by existing MT systems, but also for texts and language pairs that are not. Even with a small training set of 1000 out-of-domain parallel sentences (and an admittedly much larger monolingual corpus for language model training), we achieve acceptable results.

To investigate the effect of alignment quality on SMT performance, we trained several SMT systems on the automatically aligned Text+Berg corpus, using different alignment algorithms. We found that the two SMT systems using bleualign significantly outperform the systems using Moore's and Gale and Church's alignment algorithms.

Since the alignment quality measured in table 4 needs not be representative for the entire corpus, it is not possible to directly juxtapose alignment quality with the SMT results in table 6. For instance, Moore's algorithm returns far fewer sentence units than the system bleu-intersect (running bleualign in both translation directions and intersecting the results), even though we would expect the former to have a higher recall based on table 4. However, for a large minority of articles (330 out of 700), Moore's algorithm fails to return any beads, like in our second alignment analysis.

The evaluation does not allow us to draw any conclusions about the relative relevance of sentence alignment recall and precision for SMT. The ongoing trend for larger training sets in SMT speaks for the importance of recall, which directly determines the amount of correctly-aligned training data we can extract from a corpus.

7 Conclusion and Outlook

We have successfully demonstrated an MT-based sentence alignment algorithm that outperforms conventional algorithms on our task. Our approach is costlier in that it requires at least one automatic translation of the to-be-aligned texts. We have shown that the quality of the alignment algorithm increases with the quality of the MT system used to obtain the translation, and that the improved alignment from our algorithm has a significant effect on the performance of an SMT system trained on automatically aligned data. We found the increase in performance to be highest for very hard-to-align text pairs, specifically for texts with a high number of 1-to-0 alignments. Even though our algorithm was developed for an OCR-generated parallel text, its application needs not be restricted to such. Also, we demonstrated that the length-based algorithm by Gale and Church profits from basing the sentence alignment on an automatic translation instead of the source and target texts.

We currently do not conduct research on typologically distant language pairs, but we expect this effect to increase as the linguistic distance between the two languages increases, and we welcome any evaluation on how our translation-based approach fares against conventional sentence aligners under such conditions.

Further improvements to the tool include setting BLEU score thresholds, either on the article or sentence level, below which an alignment is discarded. Also, we can prune the search space of our initial similarity estimation to speed up the algorithm, although pruning the search space is especially dangerous in hard-to-align settings, for which this algorithm was developed. A further possible modifica-

[12] As a point of comparison, Europarl-M achieves a BLEU score of 26 on a Europarl test set.

tion to the algorithm is the use of a similarity score other than BLEU.

To reduce the dependence of our approach on external MT systems, we plan to investigate an iterative approach in which a text is first aligned using Gale and Church's algorithm. By training an SMT system with the resulting data, one can then compute a new and hopefully better alignment with bleualign.

Acknowledgments

We would like to thank Christian Hardmeier and the anonymous reviewers for detailed comments. This research was funded by the Swiss National Science Foundation under grant 105215_126999.

References

S. F. Adafre and M. de Rijke. 2006. Finding similar sentences across multiple languages in Wikipedia. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69.

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

Chris Callison-Burch and Miles Osborne. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of EACL 2006*, pages 249–256.

Stanley F. Chen. 1993. 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.

Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 57–63, Barcelona, Spain, July.

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

Martin Kay and Martin Röscheisen. 1993. Text-translation alignment. *Comput. Linguist.*, 19(1):121–142.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Robert C. Moore. 2002. 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. Springer-Verlag.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computat. Linguist.*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Morristown, NJ, USA.

Anil Kumar Singh and Samar Husain. 2005. Comparison, selection and use of sentence alignment algorithms for new language pairs. In *Proceedings of ACL 2005 Workshop on Parallel Text*, Ann Arbor, Michigan.

A. Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, CO, USA.

Jörg Tiedemann. 2007. Improved sentence alignment for movie subtitles. In *In Proceedings of RANLP, Borovets*.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005*, pages 590–596.

Martin Volk, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in building a multilingual alpine heritage corpus. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Keiji Yasuda and Eiichiro Sumita. 2008. Method for building sentence-aligned corpus from wikipedia. In *2008 AAAI Workshop on Wikipedia and Artificial Intelligence (WikiAI08)*.