Aligning the un-alignable — a pilot study using a noisy corpus of nonstandardized, semi-parallel texts

Florian Petran

Ruhr-University Bochum
Linguistics Department
Bochum, Germany
petran@linguistics.rub.de

Abstract. We present the outline of a robust, precision oriented alignment method that deals with a corpus of comparable texts without standardized spelling or sentence boundary marking. The method identifies comparable sequences over a source and target text using a bilingual dictionary, uses various methods to assign a confidence score, and only keeps the highest scoring sequences. For comparison, a conventional alignment is done with a heuristic sentence splitting beforehand. Both methods are evaluated over transcriptions of two historical documents in different Early New High German dialects, and the method developed is found to outperform the competing one by a great margin.

Keywords: word alignment, noisy text processing, semi-parallel corpora

1 Introduction[1]

Word and sentence alignment is largely regarded as a solved problem. Yet the common approaches to this either presuppose sentence splitting and standardized spelling [1], or they only work on completely parallel texts where no parts are deleted or inserted over the source and target texts (e.g. [2]). With texts that have neither of these properties, the consensus seems to be that in such situations, alignment is impossible to do automatically in a generalized way, and has to be done manually. We present an approach to the alignment of semi-parallel texts without reliable sentence breaks and with non-standardized spelling, and compare it to a standard approach using a sentence splitting heuristics. It was tested with historical texts, but there are conceivably comparable situations in Internet communication, especially in forum or newsgroup posts on common topics.

[1] We would like to thank the anonymous reviewers for helpful comments. The research reported here was financed by Deutsche Forschungsgemeinschaft (DFG), Grant DI 1557/4-1.

This paper is outlined as follows. As stated above, work with this exact type of scenario is rather scarce, but a few related approaches are described in section 2. In section 3, we introduce the text corpus used for the experiments, and in section 4 we explain the extraction of the translation dictionary we used for our alignment method. The alignment algorithm itself is described in section 5. For comparative purposes, we also tried a traditional alignment method with a heuristic sentence breaking, described in section 6. Evaluation method and results are detailed in section 7. Finally, section 8 discusses possible and desirable directions for future research.

2 Related Work

State of the art sentence breaking algorithms are usually mostly concerned with disambiguation of existing punctuation marks. There is some work on using conditional random fields to determine sentence boundaries in languages usually written without punctuation such as Chinese (e.g. [3]). However, we only have small samples available for each dialect, so it is unlikely we would have enough training data for a machine learning approach. Spelling variations even within a single text, and the abundance of inflectional morphology would be additional obstacles to the collection of training data. The clause breaking we employed is novel as far as we are aware.

As stated in section 1, alignment is largely regarded as a solved problem, so approaches that differ from current state of the art methods are largely found in older literature. Since we currently use cognates for the dictionary, our method is technically related to that of [4], who use the number of cognates found in a sentence to extend length based sentence alignment [5]. But one of the advantages of our method is that it does not presuppose any text segmentation, even though it would likely profit from a paragraph segmentation and alignment. It would furthermore be trivial to make it work with a bilingual dictionary that is not based on cognates, and going beyond the stage where we align similar words would in fact be high up on the list of future work to do (see below).

Other approaches construct a vector of the number of tokens between each occurrence of a token, and then infer an ad-hoc translation dictionary from a comparison of those vectors ([2], [6]). This is similar to our approach in that the dictionary is then used to give possible points of alignment, and that it is also designed to work with noisy texts without punctuation. But even though it is designed to work with only roughly parallel texts, the index differences will not work at all if there are larger parts of text inserted or omitted. Furthermore, the spelling variation present in our texts make it difficult to determine the identity between words in a reliable way.

Probably most similar to our approach is the char_align algorithm [7]. It was also conceived to deal with noisy text (OCR documents), and it uses cognates as well. The difference is that it tries to find alignments on the character level by constructing a scatterplot of character correspondences. The plot is then smoothed by signal processing techniques such as low-pass filtering, and a search heuristics is employed to find the path with the largest average weight. It appears that probably due to recent improvements on sentence boundary detection algorithms, this venue was not much explored afterwards. The paper does not offer a quantitative evaluation of the performance, and in fact it would be difficult to compare to other methods. Finally, a huge problem with this kind of approach is that it does not seem like it would generalize too well over languages that are not related, and do not have too many corresponding cognates. This is something that our approach should be able to do if one were to substitute our cognate based dictionary for a real translation dictionary.

3 Corpus

The texts this study is based on are two versions of the medieval religious text *Interrogatio Sancti Anselmi de Passione Domini* (Questions by Saint Anselm on the Lord's passion). There are at least 40 versions of the text from the Early Modern period in different dialects of Early New High German (ENHG). Even though these texts have the same topic and roughly the same content, they also differ greatly in language use, and there are passages missing or inserted in between the texts. Although there is a Latin version that is believed to be older, the fact that there are completely novel passages inserted in some versions indicates that they are not necessarily translations of the same text, or of each other. The absence of standardized spelling and consistent sentence boundary marking complicates matters even further.

With a length of 8,000 tokens on average, they are just about too long to do alignments and annotation all manually, yet still too short to employ machine learning approaches. The texts come in prose and verse forms, and the prose forms come in long, medium, and short lengths. As already mentioned, sentences and whole passages are missing between the texts. Additionally, the fact that a text is shorter than another one does not necessarily mean that it cannot have passages that may be absent in the longer ones, and frequently the ordering between passages is changed. In sum, even though the texts cover the same topic and indeed tell the same story, they are extremely heterogeneous, and present a very difficult scenario for automatic alignment.

For the experiments described below, we used two prose versions of the story that originated in the broader Bavarian region with slightly different dialectal background. The first is a transcription of a manuscript written in the 14th century.[2] With about 10,000 tokens it is one of the longest versions of the story; this will be our source text. The second one is a transcription of manuscript written in the late 15th century. [3] It is a medium length version with about

[2] Clm. 23371, fol. 126v – 138v from the Bavarian State Library in Munich.

[3] Lit. 176 Ed. VII, fol. 13v – 58v from the State Library Bamberg.

Source:

Sand Anſhelm der pat vnſer vrowē vō himelreich lange zeit. mit vaſten vn̄ mit wachen Vnd mit andechtigem gepet.

“Saint Anselm, he begged our lady of the heavens for a long time, fasting, waking, and with devout prayers”

Target:

Ain hoher lerer hieſz anſhelmus der pat vnſer frauen lange weill vnd zeit wainent vaſten vnd peten.

“A high teacher called Anselm, he begged our lady for a long time, crying, fasting, and praying”

Fig. 1. The beginning of source and target texts respectively. Capitalization and punctuation are the same as in the original.

5,900 tokens; this will be the target text for our aligments. [4] Fig. 1 shows the first proper sentence of each text.

4 Dictionary

The alignment method is based on a translation dictionary. There is no such preexisting dictionary, but since the dialects are very similar, we were able to automatically extract a list of cognates[5] using the BI-SIM measure [8]. BI-SIM returns a similarity value between 0 and 1, where 1 stands for an identical form, and 0 stands for two completely distinct strings.

BI-SIM has been successfully used to extract seed dictionaries for Slovenian and Croatian [9], with a similarity cutoff of 0.7. For our experiments, we set that cutoff to 0.8, based on the intuition that ENHG dialects are related more closely than Croatian and Slovenian. This was confirmed by calculating the average BI-SIM value of a small sample of cognates extracted from the texts, and also empirically determined to work better with the experiments than a lower cutoff value. To mitigate the number of false positives in the dictionary, we further excluded words with only three characters from the cognate extraction, unless they were identical. The texts are from a restricted domain, dealing with religious topics. As a consequence, we made the simplifying assumption that that similar words largely have a similar meaning, and did no additional verification of the entries.

We cannot easily stem the words due to the lack of standard orthography. Hence the dictionary contains inflected forms. Since these forms may occur in different syntactic contexts and, moreover, the texts sometimes use slightly different inflectional paradigms, not all inflectional variants of a word may be recognized as translations of the same word. If the noise stays below a certain threshold,

[4] Our versions also contained a low amount of noise where transcribers did not properly follow markup conventions. This is not unsimilar to what might be the noisy output of a web crawler or the result of an OCR.

[5] We use the term *cognates* to refer to words with a similar form and similar meaning. They do not need to have common ancestors, as in the linguistically strict sense.

the alignment method should be able to discard the wrong translations at a later stage because they will not form sequences with other token pairs.

Table 1. Sample entries from the dictionary.

		Source		Target	
1		*auzsetzigen*	"leper"	*aussetziger*	"leper"
2		*enphangen*	"receive"	*enpfangen*	"receive"
3		*ewangelist*	"Evangelist"	*eubangelist*	"Evangelist"
4	*	*gewant*	"garment"	*gewalt*	"violence"
5		*grozz*	"great/tall"	*grosz*	"great/tall"
6	*	*land*	"land"	*lang*	"long"
7	*	*leit*	"suffering"	*leib*	"body"
8	[*]	*seine*	"his" NOM PL	*seinem*	"his" DAT SG
	[*]	*seine*		*seinen*	"his" ACC SG
9		*weib*	"woman"	*weip*	"woman"
	*	*weib*		*wein*	"wine"
	*	*weib*		*weil*	"because"

Table 1 shows some sample entries from the dictionary. Lines 1–3 and 5 illustrate correct mappings of non-identical forms. Lines 4 and 6–7 show false positive entries, and lines 8 and 9 illustrate ambiguous mappings. Line 8 shows a mapping of different inflectional variants of the possessive pronoun *seine* "his." It is not strictly correct in the lexicographic sense, since the case differs, but it captures the sense in a way that can be used for alignments. In line 9, the first mapping is correct, while the other two are false positives. Overall, the dictionary covers about 69% of the types in the longer source text S, and about 99% of the types in the shorter text T.

5 Alignment Method

The basic idea of the alignment method is to find the longest non-conflicting, corresponding sequence of translation candidates in both texts. In this section, we explain the procedure used to find those sequences in detail.

Be s_i a token $s \in S$ at position i, and t_j a token $t \in T$ at position j. As a first step, we collect all alignment candidates for each s. That is, for each s, we retrieve its translation from the dictionary and record their occurrences (indices) in T as alignment candidates. This way, 56.2% of the tokens in S are assigned at least one candidate, with an overall average of 28.2 candidates per source token.

In the next step, we merge candidates to bigram sequences if both their source and target components are *close* to each other. The *proximity condition* $C(r_i, r_j)$ states that the difference between their indices may not exceed 2.

$$C(r_i, r_j) = \begin{cases} 1, & \text{if } j - i < 3 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

So for each pair $s_i : t_j$ we check if the token at s_{i+1} has a translation candidate t_k that is close to t_j. If it does, we merge both translation pairs to a bigram sequence. Then we remove them from the set of data still to be treated, and continue with the next pair that is not yet part of any sequence.

Often, there are tokens that cannot be aligned to a counterpart in the other text. There may be punctuation marks that are absent in the other text; some tokens are missing from the dictionary, or some may not have a counterpart in the other text at all. Hence, we allow the algorithm to skip a single token in S when looking for the next component of a bigram sequence, instead of moving onward token by token.[6] The amount of candidates in S that may be skipped as well as the maximum difference of the indices still allowed to satisfy the proximity condition were empirically determined by investigating a small sample. Note that the treatment of S and T is asymmetrical here. This is possible because the same process is later applied in the other direction as well.

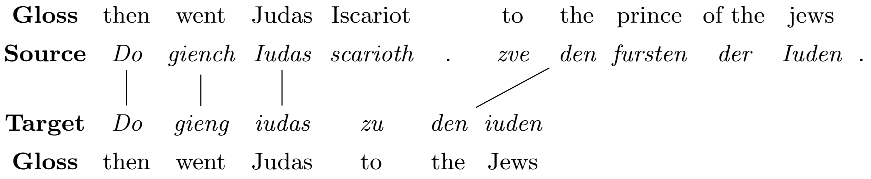
Gloss then went Judas Iscariot to the prince of the jews
Source Do giench Iudas scarioth . zve den fursten der Iuden .
Target Do gieng iudas zu den iuden
Gloss then went Judas to the Jews

Fig. 2. Example of a correctly aligned sequence.

Fig. 2 shows a sequence of correct alignment pairs. For this example, the algorithm previously arrived at *Do* “then” which has 137 translation candidates in T, and finds *giench* “went” as the next token, which has 8 translation candidates. Among the $137 \cdot 8$ possible pairings, there are only two alignments where the translation candidates occur in proximity of each other, so that we merge these pairings to a bigram. For other bigrams in S, there are multiple instances of translation candidates occurring close to each other in T, which means we would have multiple competing bigram sequences at that position. Overall, at each position where some bigram starts there are 3.72 competing sequences on average. The combination of candidates results in 7,557 bigram pairs starting at 2,034 different positions. As stated above, 56.2% of the tokens in S are assigned one or more translation candidates. Among those, 35.72% are at the beginning of at least one bigram sequence after this step.

Now follows the sequence expansion step. Here, we try to expand the sequences at their tail end by adding alignment pairs that are not yet part of any other sequence. We also relax the conditions for proximity slightly: while in the bigram merging step, we allow a one-token skip in S, the skip may now also occur

[6] The skipping of candidates when looking for similar n-grams in different texts is also successfully employed in the ROUGE-S metrics [10].

on the target side, in addition to the maximum allowable index difference from the proximity condition. That means the difference between the indices may be three if one of the tokens does not have a candidate assigned to them, and two otherwise. The reasoning behind this is that we first try to find a close bigram pair to anchor the alignment to, and then gradually expand outwards from it.

In the example given in Fig. 2, we find the next token in S to be *Iudas* "Judas," which aligns well with its counterpart in T. The following token *scarioth* "Iscariot" is entirely absent from T, as is the full stop after that. Then follows *zve* – *zu* "to," which should be aligned. In this case, the translation is missing from the dictionary because we excluded words with three characters or less from cognate extraction process (see section 4). Because we relaxed the proximity criteria, as just described, we are now able to add the following token *den* and its counterpart in T to the alignment sequence. It may appear as if the maximum allowable index difference from the proximity condition was exceeded in this case. However, *scarioth* has no translation candidate assigned to it, and hence, can be skipped and does not count towards the maximum.

Each pair that has been added to a sequence is removed from the collection of candidates. We iterate through the sequences until we have made a full round without adding any more pairs. Then the process is repeated for the reverse direction, from T to S to account for the asymmetrical treatment in the first part of the process.

In our texts, the sequence expansion step results in a collection of sequences of up to 4 token pairs, with an average length of 2.1. The fact that the average only slightly exceeds the minimum sequence length indicates that the majority of sequences did not get expanded in this way, since they are chance co-occurrences by chance of two token pairs. They will get discarded at a later stage when preferrable matches for their participating tokens have been found.

The next step concerns sequence merging. Since the algorithm only add pairs to the tail ends of the sequences, and the proximity criteria is laxer than in forming the original sequences, it is possible that the expansion has brought the end of a sequence so close to the beginning of another one that they can be merged to one. We iterate through all the sequences at each positions and apply the proximity criteria to the end of this sequence and the next position where a sequence starts.

For the example in Fig. 2, we do not find any such sequence, but consider the correctly aligned sequence in Fig. 3. In the bigram merging step, we found bigrams starting at *gewalt* and at *stat*. The subsequent sequence expansion step has created a situation in which the tail end of the first sequence, *in*, is close to the beginning of the second one. Consequently, both are merged to one long sequence in the sequence merging step. This increases the maximum sequence length to 10, while the average remains at 2.1. Again, this indicates that the majority of sequences are neither expanded nor merged.

The final step is score assignment, where a confidence score is determined for each sequence. This score is based on the length of the sequence, the average difference between the indices in S and T, and the average BI-SIM value of the aligned words. The employment of the BI-SIM measure to grade the alignments again relies on the assumption that the languages are related, as did the dictionary extraction method discussed above. But since this measure did not contribute as much to the quality of the results as the other two components, it might be possible to replace or omit it should the alignment method be employed with unrelated languages.

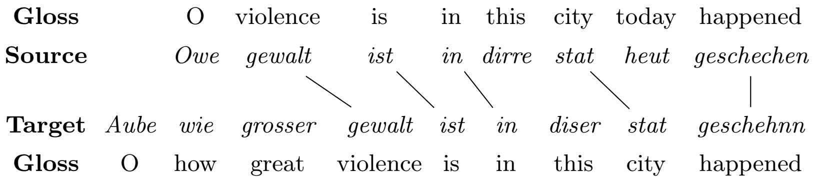
Gloss O violence is in this city today happened
Source *Owe gewalt ist in dirre stat heut geschechen*
Target *Aube wie grosser gewalt ist in diser stat geschehnn*
Gloss O how great violence is in this city happened

Fig. 3. Example of two sequences that have been merged.

Be r a sequence of aligned indexes in T and S respectively, with a length of n and with $r_{i,1}$ denoting the source index part of alignment pair i, and $r_{i,2}$ the target index part of that same alignment pair. The BI-SIM measure can then be defined as follows in equation 2.

$$b = \frac{\sum_{i=1}^{n} \text{bi\_sim}(s_{r_{i,1}}, t_{r_{i,2}})}{n} \tag{2}$$

The index difference measure is defined in the following equation 3. As above, s_i and t_i are tokens at index position i from s and t respectively, while $|S|$ and $|T|$ are the total amount of tokens in the respective text. Note that a lower index difference is desirable here, so we subtract from 1 to invert the values.

$$d = 1 - \left(\frac{\sum_{i=1}^{n} \left| \frac{r_{i,1}}{|S|} - \frac{r_{i,2}}{|T|} \right|}{n} \right) \tag{3}$$

Now be B the set of b values, D the set of all d values, and N the set of all lengths of sequences for all $r \in R$. In order to give all measures of confidence the same weight, every one is normalized by the maximum value over all sequences. Then we take the mean value.

$$c = \frac{1}{3} \left(\frac{b}{\max B} + \frac{d}{\max D} + \frac{n}{\max N} \right) \tag{4}$$

Other confidence measures to the score we considered include the amount of alternate sequences starting at the same position, and the average length difference between the words, and the average difference in relative frequency. These were tested but ultimately not employed, since the first two had a detrimental effect on the quality of the results, and the latter did not make any difference at all.

After the scores are assigned, lower-scored sequences are discarded. That is, if any token of a sequence (from S or T) is simultaneously a member of a higher scoring sequence, the lower scoring one is discarded. Since we aim for a a high-precision result rather than for a strong recall, we also discard all ambiguous sequences. Those are sequences that conflict with others, but a decision could not be made for either one because they have the same score. As a final result, this method gives us 1,280 1:1 alignment pairs. That is a coverage of 22.92% of the shorter text T.

6 Alternate Alignment

For the reasons outlined in section 1, a conventional alignment is difficult to accomplish. The main reason was that punctuation marks cannot be used as segment boundaries. To apply conventional alignment tools, we therefore have to come up with some segmentation for the text. Since statistical alignment methods do not usually use linguistic knowledge, we assumed that it is not necessarily required that those units be sentences.

We used a list of possible spelling alternates of frequent conjunctions such as *und* "and" or *oder* "or" to split the text into segments that might resemble clauses. This will enable us to align those segments, and afterwards align the words within them. We did not extract the conjunction alternates from the text, but had them supplied by a historical linguist based on their intuition of what possible alternates could occur. An additional presupposition for the segmentation step was that a sentence was not allowed to have only one word, since conjunctions frequently occurred directly next to each other.

However, the results of the segmentation step already seems problematic. For one, clauses or sentences do not always start with a conjunction, so the segments frequently crossed clause boundaries. Furthermore, ENHG texts make excessive use of binomial pairs of synonymous or partly synonymous words ("Zwillingsformel") joined with a conjunction to express a single concept, a stilistic device borrowed from classical Latin. For example, Jesus is described as *gefangen und gebunden* "caught and bound," and his disciples upon hearing this as *schreiend und weinend* "crying and weeping." Using our segmentation heuristics, those would then be taken as two clause segments of their own, even when, grammatically speaking, they should be part of one clause. According to a tentative look at the output of this step, the latter problem seemed to make up the majority of the mis-segmentations, indicating that this step might well work better with a modern language text.

The results of the segmentation step were sentence aligned using the Gargantua toolkit [11]. It is basically an extension to other work in sentence alignment that combines length based alignment with multiple iterations over translation model based alignment [12], but it handles M:N alignments with N,M > 2. Due to the way the results of our segmentation step turned out (see section 7), this was deemed highly desirable. Word alignment within the aligned segments was then done using GIZA++ [1]. Alignments involving the NULL token and N:1/1:N alignments had to be specially considered for the output. For the former, our method produces no output at all if it does not find an alignment, as does Gargantua, but GIZA explicitly aligns it to NULL. Those were accordingly not considered in the evaluation, although further manual alignments seem to indicate that they may constitute up to about a third of the alignments. Our method does not yet support multi-token alignments, so to compare the results, they had to be converted into sequences of 1:1 alignments. For example, s_i : (t_j t_k) became s_i : t_j; s_i : t_k. After all that, the method produced about 4,979 unique pairs, which amounts to a coverage of 84.62% of the shorter text.

7 Evaluation and Conclusion

Since the method is very much a work in progress on an ongoing project, there is no complete gold standard so far with these two texts, which complicates evaluation. Instead, we evaluated recall on a subset of 2,500 pairs that have been completed by one annotator. This is a bit of a problem for the evaluation of precision, since it does not cover all of the tokens in the text. Running the alignments on just a subset of the texts is not an option either, since the ordering of sequences is heavily changed between the texts, so it is not easily possible to evaluate on the first n tokens of both. All unannotated tokens are therefore counted as errors in the results presented below, even if they may in truth be correct.

Two properties of our alignment method further complicate the matter. First, we currently only output 1:1 pairs, whereas a lot of N:M alignments seem to occur in the part that is already annotated. For the evaluation, we converted those into sequences of N·M 1:1 alignments, as described in section 6. But even if our system's output for these is partly correct, this would decrease the count of correct results. Since our algorithm only gives one alignment for each token, for every 1:1 pair that is correctly produced, there would be N-1 pairs that can definitely not be in the output. Alternatively, we could exclude all N:M alignments from the evaluation until our system is able to handle such cases. Second, NULL alignments occur quite frequently, as should be expected with semi-parallel texts. We could count every token where we did not produce an alignment as an alignment with the NULL token if we assumed that a NULL alignment was the default case. This would increase recall at the expense of precision. If we did not include NULL alignments at all in our method, this would increase precision, but at the expense of recall, since we would not cover all of the tokens in the text.

Table 2 shows precision, recall, and balanced F-score for all four possible cases. The F-measure remains more or less constant as we trade off between precision and recall, as should be expected. As just explained, we suspect that the actual precision may be higher than our partial gold standard accounts for, so we had our annotator manually examine the non-NULL pairs our method produced. It was found that 51.7% of these were actually correct ones which

Table 2. Evaluation of our method.

NULL	N:M	precision	recall	F-measure
-	+	42.2%	23.4%	30.1
+	+	22.1%	50.7%	30.8
-	-	42.2%	25.7%	31.9
+	-	22.1%	55.5%	31.6

puts the actual precision we can achieve higher than reported in the table, even though this value is not exactly comparable.

Evaluating the output of the traditional method is simpler, since it does account for both NULL and N:M alignments. Its output was found to contain only two of the correct pairs in our incomplete gold standard, which amounts to a precision of 0.04% and recall of 0.07%. It should be noted, however, that all pairs that were not covered by the gold standard are again counted as wrong answers, so actual performance may be higher. On the other hand, we already know that those responses involving a token our method aligned correctly are wrong if they differ. In addition, we manually evaluated 540 pairs involving a token from T or S that had been incorrectly aligned by our system. After all, it is possible that the traditional method was right in those cases where our method guessed wrong. Of all those, not a single one was correct. This means that we have 23% of the output of the traditional method we know are incorrect.

All in all, according to the cursory evaluation outlined here, this method seems to perform far worse than the algorithm proposed in this paper. Since about 77% of the results of the traditional method were still not checked at all, it is not technically impossible that it successfully produced correct alignments where our method did not find anything. Judging from the quality of the evaluated alignments, however, that is highly unlikely, to say the least. Further evaluation will help clarify all these points.

A tentative qualitative error analysis of both methods seems to indicate that the errors of our method comprise mostly sequences of function words, and that it might benefit from a list of stop words to disregard in the alignments. The traditional method is difficult to analyze since there are few sensible alignments in the final output. Based on the problems with the segmentation step outlined in section 6, the sentence aligner did not seem to produce a lot of sensible output, either, which, of course, posed problems for the word aligner. It seems that it did not handle the large omissions between the texts very well. The overall method seems to suffer greatly from the haphazard way of sentence splitting, and would likely benefit greatly from improvements upon that.

In conclusion, we have shown that it is possible to word-align only roughly comparable texts. Since the environment the method was developed for does not have standardized spelling, or punctuation, we do not make use of such clues, and accordingly do not rely on pre-aligned larger beads to accomplish the final word alignment. The only resource our method makes use of is a translation dictionary. In our case, this is extracted using cognates - hence in its current state, the method is only applicable to closely related languages. It might be extended to unrelated languages by employing a translation dictionary from another source, or by employing a different way of extracting the dictionary. Even though the method is still in its infancy, it already outperforms conventional tools in this setting by a great margin. It succeeds in delivering alignments in a very difficult environment, and this is a success we hope to further improve upon.

8 Future Work

Future work would of course first and foremost include finishing the creation of a gold standard covering the whole document to evaluate more thoroughly against. This concerns especially the evaluation of the traditional method. Regarding the evaluation, another venue that should be explored is the performance of the char_align algorithm [7] for our particular problem.

Regarding the algorithm itself, the most pressing issues are to have it output more alignments, aligning tokens that are not found in the dictionary, and handling alignment patterns other than 1:1. In its current state, the method does omit a lot of the output it produces because it cannot decide between the sequences with our scoring method. Improvements on the assignment of confidence scores could help to improve the coverage of our system. The alignment of tokens not covered by the dictionary is something that could be handled based on cognates, such as alignment of the closest match, or possibly word length, although it would be prudent to limit those to a window around established sequences.

Employing a stemmer could theoretically help, but since those mostly work with a list of possible affixes, it would be difficult to do. As for the alignment patterns, N:1/1:N and even N:M alignments did form a considerable part of the manually annotated data, since expressions are often paraphrased between the texts. So this is a crucial issue, but so far, not one where an obvious solution presents itself. Handling of alignments involving the NULL token is connected to that, since a NULL token alignment means that something should not be added to a multi-alignment. As stated above, these appear to be fairly frequent as well.

Yet a different option to explore is if and how our method could benefit from a combination with the traditional, or other methods. Since the sequences provide a kind of text segmentation that could be similar to paragraphs, the traditional method might produce better results if it were to be combined somehow, and the output of the traditional method could be used to enhance the results of our method. An open question in this regard is whether the texts are at all long enough to train a translation model on them.

A final option we want to explore comes from the specific scenario we work in. Since we have about 40 different versions of the text that are all supposed to be aligned to each other, we could try to use the amount of text to our advantage by exploiting alignment transitivity [13]. This means that if a_i aligns to c_j, and c_j to b_k, then we can assume that a_i aligns to b_k. It could conceivably contribute greatly to the overall coverage of the results. Since we would need multiple passes over various texts for this, improving the performance of the algorithm would also be an issue.

References

1. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics **29** (2003) 19–51
2. Fung, P., Church, K.W.: K-vec: a new approach for aligning parallel texts. In: Proceedings of the 15th conference on Computational linguistics-Volume 2, Association for Computational Linguistics (1994) 1096–1102
3. Huang, H., Chen, H.: Pause and Stop Labeling for Chinese Sentence Boundary Detection. In: Proceedings of Recent Advances in Natural Language Processing. (2011) 146–153
4. Simard, M., Foster, G.F., Isabelle, P.: Using cognates to align sentences in bilingual corpora. In: Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing-Volume 2, IBM Press (1993) 1071–1082
5. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. Computational Linguistics **19** (1993) 75–102
6. Fung, P., McKeown, K.: Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. In: Proceedings of the Association for Machine Translation in the Americas (AMTA-94). (1994) 81–88
7. Church, K.W.: char_align: a program for aligning parallel texts at the character level. In: Proceedings of the 31st annual meeting on Association for Computational Linguistics, Association for Computational Linguistics (1993) 1–8
8. Kondrak, G., Dorr, B.: Identification of confusable drug names: A new approach and evaluation methodology. In: Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics (2004) 952–958
9. Ljubešić, N., Fišer, D.: Bootstrapping bilingual lexicons from comparable corpora for closely related languages. In Habernal, I., Matoušek, V., eds.: Text, Speech and Dialogue. Volume 6836 of Lecture Notes in Computer Science., Berlin / Heidelberg, Springer (2011) 91–98
10. Lin, C.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004). (2004) 25–26
11. Braune, F., Fraser, A.: Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora. Coling 2010: Poster Volumes (2010) 81–89
12. Moore, R.: Fast and accurate sentence alignment of bilingual corpora. In Richardson, S., ed.: Machine Translation: From Research to Real Users. Volume 2499 of Lecture Notes in Computer Science. Springer, Berlin / Heidelberg (2002) 135–144
13. Simard, M.: Text-Translation Alignment: Three Languages Are Better Than Two. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, Association for Computational Linguistics (1999) 2–11