Simultaneous Sentence Boundary Detection and Alignment with Pivot-based Machine Translation Generated Lexicons

Antoine Bourlon[1], **Chenhui Chu**[1], **Toshiaki Nakazawa**[1], **Sadao Kurohashi**[2]
[1]Japan Science and Technology Agency
[2]Graduate School of Informatics, Kyoto University
E-mail: {bourlon, chu, nakazawa}@pa.jst.jp, kuro@i.kyoto-u.ac.jp

Abstract

Sentence alignment is a task that consists in aligning the parallel sentences in a translated article pair. This paper describes a method to perform sentence boundary detection and alignment simultaneously, which significantly improves the alignment accuracy on languages like Chinese with uncertain sentence boundaries. It relies on the definition of hard (certain) and soft (uncertain) punctuation delimiters, the latter being possibly ignored to optimize the alignment result. The alignment method is used in combination with lexicons automatically generated from the input article pairs using pivot-based MT, achieving better coverage of the input words with fewer entries than pre-existing dictionaries. Pivot-based MT makes it possible to build dictionaries for language pairs that have scarce parallel data. The alignment method is implemented in a tool that will be freely available in the near future.

Keywords: Sentence Alignment, Sentence Segmentation, Pivot-based Machine Translation

1. Introduction

Sentence alignment is a task that consists in aligning the parallel sentences in a translated article pair, which are crucial for machine translation (MT) (Koehn et al., 2003). Previous studies first split the source and target articles into sentences respectively using punctuation information, and then align the source and target sentences based on sentence length and/or bilingual lexicons (Ma, 2006). However, the monolingually determined sentence boundaries are not optimized for sentence alignment, because translation equivalents might cross the monolingual sentence boundaries. In this paper, we propose a method to perform sentence boundary detection and alignment simultaneously, which significantly improves the alignment accuracy.

Sentence alignment methods are generally based on two kinds of algorithms: length-based algorithms align sentences according solely to their lengths (Brown et al., 1991), while lexicon-based algorithms use lexical information to calculate similarity between source and target sentences (Ma, 2006). Length-based algorithms are typically faster but not suited for processing noisy corpus (corpus containing article pairs with omitted or wrong translations). Lexicon-based algorithms, like the one used in Champollion (Ma, 2006)[1] are more robust. However their performance highly depends on the coverage and quality of the lexicon, and large-scale lexicon with high quality is not easy to obtain especially for low resource language pairs. In this paper, we propose to use lexicons generated by a pivot-based MT system, which could be constructed even for low resource languages (Dabre et al., 2015). Experiments conducted on Chinese-Japanese scientific articles verify the effectiveness of our proposed method.

2. Simultaneous Sentence Boundary Detection and Alignment

Our proposed alignment method consists in the following steps

1. Split source and target articles into sentence candidates.
2. Normalize words in sentence candidates to maximize matching rate between words in source, target, and lexicons.
3. Compute alignment path with the highest similarity between source and target. An alignment path is composed of pairs of article segments.[2]
4. Adjust sentence boundaries by merging sentence candidates that are part of the same segment into one sentence.

Steps 2 and 3 are the basic steps used in existing sentence aligners such as Champollion (Ma, 2006). The additional steps 1 and 4 allow to treat the sentence segmentation issue in a more flexible way than what can be done with standalone sentence splitters.

2.1. Sentence Splitting

One issue in sentence alignment is to determine sentence boundaries. Some delimiters (such as periods in Chinese and Japanese) can be considered with a very high probability as sentence boundaries while others, like commas in Chinese or semicolons and colons in Chinese and Japanese may or not correspond to boundaries depending on the context. In the following we call the former type of characters “hard delimiters”, and the latter “soft delimiters”. For example Chinese side of the Chinese-Japanese article pairs we used for evaluation contains many sentences composed of comma-separated fragments, and can be split into smaller groups of fragments without losing meaning or grammatical correctness.

In our proposed method we use a list of hard and soft delimiters to define boundary candidates, and use the sentence alignment result to adjust these sentence boundaries. Only boundaries defined by soft delimiters are adjusted. We use

[1] http://champollion.sourceforge.net

[2] A segment is a group of one or more sentences.

Language	Hard Delimiters	Soft Delimiters
Japanese	period, dot	semicolon
Chinese	period, dot	semicolon, colon, comma

Table 1: Hard and soft sentence delimiter sets for Japanese and Chinese.

hard and soft delimiters for Japanese and Chinese as defined in Table 1.

The use of soft delimiters makes it possible to maximize the number of 1-to-1, as-small-as-possible sentence pairs, by aligning source language (typically Chinese) boundaries on target language (typically Japanese) boundaries. This is important because 1-to-1 pairs of small sentences are generally more valuable when building parallel corpus (e.g., for MT systems) than long, multi-sentence pairs.

In order to reduce the number of sentence candidates, which has a direct impact on alignment computation time, we do not split sentences in the following cases:

- Dot in strings of alphanumeric or symbol characters.
- Dot in well-known abbreviations ("Dr.", "U.S.A.", "Mr.", "i.e.", etc.).
- Punctuation inside parenthesis and quotation blocks.
- Punctuation followed by Japanese words that have a very low probability to be at the start of a sentence.

2.2. Normalization

We normalize sentence candidates and dictionaries in order to maximize the matching rate between source, target, and dictionary. Japanese sentences are lemmatized using Juman (Kurohashi et al., 1994), while no lemmatization is performed for Chinese. Characters in Japanese and Chinese sentences with an equivalent ascii character are then converted into ascii, and letters are converted to lowercase.

2.3. Alignment Path Computation

In order to extract the best alignment path, we calculate the similarity between segments in source and target, and search for the path with maximum similarity using a dynamic programming algorithm.

2.3.1. Similarity Function

Our similarity function only uses the total length (in characters) of matched expressions, and can be expressed as a simple Jaccard coefficient.

$$similarity = \frac{total\ length\ of\ matched\ expressions}{sentence\ length} \tag{1}$$

We use both direct matches and matches through dictionary lookup, between source and target. The similarity function does not use segment length information or other language-specific heuristics, which makes it easier to use on different language pairs.

2.3.2. Matching Unit

Lexicon-based matching algorithms use tokens (from a tokenizer output) as matching unit (Ma, 2006). However, this prevents matching against dictionary terms[3] made of multiple tokens, and may also reduce the number of exact matches between source and target, in case tokenizers for both languages (e.g., Japanese and Chinese) do not follow the same word segmentation rules. In order to match multi-token terms, our algorithm ignores tokenizing information and simply tries to find the longest matches (up to 100 characters in one match) in segments on the source side, starting from the head.

2.3.3. Dynamic Programming Algorithm

We use a dynamic programming algorithm similar to Champollion (Ma, 2006), combined with the similarity function described above, to calculate the alignment path with the maximum global similarity (score). The algorithm uses a recurrence relation that expresses the alignment score of the first m source sentences and n target sentences as the sum of one of the prefixes of these sentences plus the similarity between the remaining suffix. The recurrence relation is defined as follows.

$$\begin{aligned} score(m,n) = \max_{0\le i\le I, 0\le j\le J} (score(m-i, n-j) \\ + similarity(SSeg_{m-i+1,m}, TSeg_{n-j+1,n})) \end{aligned} \tag{2}$$

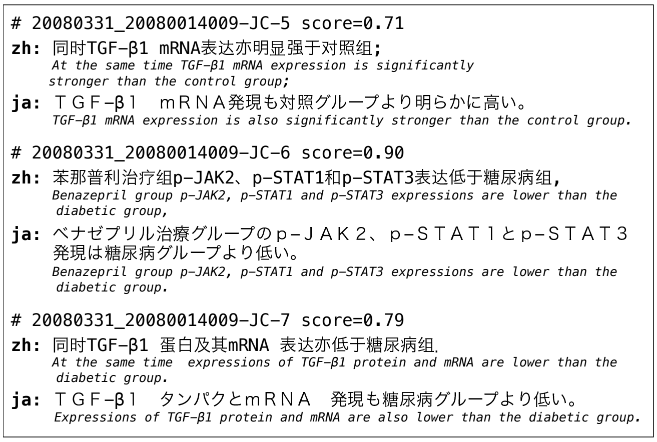
$score(a,b)$ represents the score of the alignment path up to source sentence a and target sentence b. $SSeg_{a,b}$ and $TSeg_{a,b}$ represent article segments from sentence a to b for the source and target, respectively. I and J represent the maximum number of sentence fragments, respectively on the source and target side, that can be concatenated into one alignment. Unlike Champollion which only allows up to 4 concatenated sentences, I and J can be set as parameters, with no upper limit. Since we split Chinese articles on soft delimiters like commas, we need to try more sentence combinations on the Chinese side than on the Japanese side. In our evaluation experiment we concatenate at most $I = 30$ sentence candidates for Chinese and $J = 2$ candidates for Japanese. We also allow single-side pairs 0-to-N and N-to-0, but give them a similarity score of -0.1, similarly to Champollion. We do not give penalty scores to any of the other alignment patterns, since good alignments made of many soft-delimited fragments are not rare.

2.4. Output

We output each segment pair along with their similarity score in the best alignment path. Sentences that were extracted using hard delimiters are outputted as multiple sentences, on multiple lines, while sentence candidates extracted using soft delimiters are merged into one sentence on a single line.

Below is an example of alignment output, with three segment pairs extracted from the same article. In the second pair, the two Chinese sentence candidates "苯那普利治疗组p-JAK2、" and "p-STAT1和p-STAT3表达低于糖尿病

[3] In this paper, we call entries in the dictionary terms. A term consists of one or multiple tokens.



```
# 20080331_20080014009-JC-5 score=0.71
zh: 同时TGF-β1 mRNA表达亦明显强于对照组;
     At the same time TGF-β1 mRNA expression is significantly
    stronger than the control group;
ja: ＴＧＦ－β１　ｍＲＮＡ発現も対照グループより明らかに高い。
     TGF-β1 mRNA expression is also significantly stronger than the control group.

# 20080331_20080014009-JC-6 score=0.90
zh: 苯那普利治疗组p-JAK2、p-STAT1和p-STAT3表达低于糖尿病组,
     Benazepril group p-JAK2, p-STAT1 and p-STAT3 expressions are lower than the
      diabetic group,
ja: ベナゼプリル治療グループのｐ－ＪＡＫ２、ｐ－ＳＴＡＴ１とｐ－ＳＴＡＴ３
     発現は糖尿病グループより低い。
     Benazepril group p-JAK2, p-STAT1 and p-STAT3 expressions are lower than the
      diabetic group.

# 20080331_20080014009-JC-7 score=0.79
zh: 同时TGF-β1 蛋白及其mRNA 表达亦低于糖尿病组.
     At the same time  expressions of TGF-β1 protein and mRNA are lower than the
      diabetic group.
ja: ＴＧＦ－β１　タンパクとｍＲＮＡ　発現も糖尿病グループより低い。
      Expressions of TGF-β1 protein and mRNA are also lower than the diabetic group.
```

Figure 1: Sentence alignment output example (English translations added for reference).

组,” separated by the soft delimiter “、” have been merged into one sentence which perfectly matches the corresponding Japanese sentence. Without using soft delimiters, a 1-to-2 segment pair would have been generated instead of two 1-to-1 pairs.

3. Pivot-based MT Generated Lexicons

To address the lexicon coverage and quality problem, we apply the pivot-based MT dictionary construction method proposed in (Dabre et al., 2015).

Pivot-based MT has been shown to be a possible way of constructing a dictionary for language pairs that have scarce parallel data (Dabre et al., 2015). The assumption of this method is that there is a pair of large-scale parallel data: one between the source language and an intermediate resource rich language (henceforth called pivot), and one between that pivot and the target language. We can use the source-pivot and pivot-target parallel data to develop a source-target term translation model for dictionary construction. This method can address the data sparseness problem by directly merging the source-pivot and pivot-target terms, because it can use the portion of terms to generate new terms.

(Dabre et al., 2015) constructed a large Chinese-Japanese dictionary ($3.6M$ terms) manually evaluated to be 90% accurate. They addressed the noisy nature of pivoting large phrase tables by statistical significance pruning (Johnson et al., 2007). They also exploited linguistic knowledge of common characters (Chu et al., 2013) shared in Chinese-Japanese to further improve the translation model. In addition, they used bilingual neural network language model features for reranking the n-best list produced by the pivot-based system, and achieved a further accuracy improvement. They also used character based neural MT to eliminate the out-of-vocabulary (OOV) terms, which further improved the quality.

4. Experiments

We conducted sentence alignment experiments on Chinese-Japanese scientific data to verify the effectiveness of our proposed method.

Set	# articles	# pairs	# good pairs
Development	100	582	423
Test	100	601	436

Table 2: Number of articles and segment alignment pairs in the development and test reference sets.

Delimiter	Development		Test	
	Japanese	Chinese	Japanese	Chinese
period	546	121	560	144
dot	0	238	0	216
semicolon	5	51	3	33
colon	3	9	2	12
comma	4	148	4	175
other	4	11	4	15

Table 3: Number of sentences ending with each type of delimiters in the development and test reference sets. “other” corresponds to a missing punctuation or suspension points at the end of some articles.

4.1. Test Set

We built the test set by randomly selecting two sets of 100 articles out of the 1,082,345 pairs in the LCAS corpus provided by JST[4], for tuning and testing, respectively. We asked two professional Chinese-Japanese translators to create and annotate sentence alignment reference data. Annotation consisted in marking “good” segment pairs, defined as pairs with no major word-level translation omission. We then asked the translators to check and fix reference data in two steps:

- Cross-check and fix each other’s reference data.
- Compare good pairs from the reference data with automatic alignment data produced by the alignment tool. If the translators considered some of the automatic alignment data better than the reference alignment, translators fixed the reference alignment.

Table 2 shows the number of alignments pairs for each of the obtained reference sets, “Development” and “Test”, while Table 3 shows the number of sentences ending with each type of punctuation delimiters.

4.2. Settings

In our experiments, we compared alignment results of our proposed method with a baseline composed of a sentence pre-segmentation step followed by a sentence alignment step using Champollion (Ma, 2006). We used the same sentence segmentation algorithm for the baseline pre-segmentation step and the sentence candidates extraction (step 1 in Section 2.) of our proposed method, with the following settings: for the baseline we split either on “hard delimiters only”, or both “hard and soft delimiters”, as described in Table 1. The proposed method uses both “hard and soft delimiters”, as described in 2.1., but we also tried a version without soft delimiters, for direct comparison with the baseline “hard delimiters only”.

[4] http://www.jst.go.jp

We also compared six types of dictionaries:

- None: Did not use any dictionaries, and thus only identical characters will be used for matching.
- EDR: 298,857 Japanese-Chinese entries extracted from the EDR Electronic Dictionary.
- MT-Noun: Extracted the noun strings in the 100 Japanese sample articles with Juman, and then translated them using the pivot-based MT system in (Dabre et al., 2015), obtaining 4,263 entries.
- MT-NVAA: Extracted the strings that only consist of noun, verb, and adjective in the 100 Japanese sample articles with Juman, and then translated them using the pivot-based MT system in (Dabre et al., 2015), obtaining 7,004 entries.
- EDR+MT-Noun: Merged entries from EDR and MT-Noun, obtaining 302,180 entries.
- EDR+MT-NVAA: Merged entries from EDR and MT-NVAA, obtaining 304,305 entries.

For each of the above boundary detection and dictionary settings, we extracted from the alignment result segment pairs with a similarity score above a certain threshold. We used three types of thresholds:

- Null: Extract all pairs with a positive score (i.e., pairs that are not omissions).
- F measure optimization: Threshold giving the highest F measure.
- Precision optimization: Threshold giving the highest F measure with a precision ≥ 0.9.

The last two thresholds were tuned on the development set. We report for each setting the F measure, together with precision and recall. Since the input data is not sentence-segmented, we use fragment pairs as unit to calculate precision and recall. Fragment pairs are generated as follows, from both the automatic alignment result and the reference alignment data.

1. Split each article on all possible punctuation characters.
2. Number each fragment with its position in the article.
3. Group together source and target fragments that belong to the same alignment pair.
4. For each alignment (group), generate fragment pairs using the cartesian product between source and target fragment positions.

Precision and recall are then calculated by matching "good" alignment pairs in the reference data and in the automatic alignment result.

4.3. Results

Table 4 shows the sentence alignment results of the proposed method and the baseline (Champollion, (Ma, 2006)), using the alignment methods and sentence delimiter settings described in Section 4.2. "Hard" and "Hard+Soft" denote splitting sentences on "hard delimiters only" and both "hard and soft delimiters", respectively. Only the pairs with omissions (i.e., a negative score) are ignored when comparing with the annotated "good" pairs of the reference data.

The best results are obtained by the proposed method using hard and soft delimiters. The proposed method also improves alignment accuracy over the baseline even when using a sentence splitting strategy similar to the baseline. This can be explained by a higher matching ratio between source and target, with or without lexicons: the proposed method can match terms made of multiple words or an arbitrary number of characters, ignoring word boundaries, while the baseline only performs word by word matching.

We can also see that splitting Chinese on soft delimiters is enough to significantly improve the baseline accuracy, even though Champollion, used for the baseline, only supports alignments up to 4-to-1, which may not be enough when splitting sentences on frequent delimiters like comma.

Figure 2 shows an example of alignments produced by the two baseline methods and the proposed method, on one article of the test set. The baseline generates an alignment that is either less precise than the proposed method (single 1-to-2 alignment vs. two 1-to-1 alignment), or too fragmented and inaccurate due to the 4-to-1 limitation of the alignment algorithm.

Pivot-based MT generated lexicons give a better F measure than EDR, despite fewer number of entries (4,263 and 7,004, vs. 298,857). This can be explained by the higher coverage on the test set, as shown in Table 6. EDR+MT generated lexicons give similar or lower scores than MT generated lexicons alone with tuning.

Table 5 shows alignment results after tuning the proposed method to maximize F measure, with and without a lower limit on the precision. The highest F measure is obtained with the largest dictionary, EDR+MT-NVAA, scoring 3% higher than EDR, but with no significant improvement over MT-Noun and MT-NVAA. The highest F measure with a precision ≥ 0.9 is obtained by the smallest lexicon, MT-Noun.

5. Related Work

Recent studies on sentence alignment methods focus on improving speed and accuracy by combining length-based and lexicon-based algorithms (Moore, 2002; Li et al., 2010).

These methods require the input corpus to be segmented into sentences, considering sentence segmentation as a mostly solved issue. However sentence segmentation is not trivial for languages like Chinese, where commas may or not be used as sentence delimiters. (Jin et al., 2004) and (Xue and Yang, 2011) proposed a method to categorize commas into delimiters and non-delimiters based on features extracted from text surrounding them. (Jin et al., 2004) obtained a detection score (F measure) of 93% for non-delimiters and 70% for actual sentence delimiters. Our proposed method, less complex, tackles both the sentence

Alignment Method	Delimiters	Dictionary	Precision	Recall	F
Baseline (Sentence split preprocessing + Champollion)	Hard	None	0.4423	0.8849	0.5898
		EDR	0.4293	0.9180	0.5850
		MT-Noun	0.4311	0.9076	0.5845
		MT-NVAA	0.4385	0.9569	0.6014
		EDR+MT-Noun	0.4242	0.9189	0.5804
		EDR+MT-NVAA	0.4332	**0.9619**	0.5973
	Hard+Soft	None	0.6687	0.7571	0.7102
		EDR	0.6572	0.7426	0.6973
		MT-Noun	0.7042	0.7960	0.7473
		MT-NVAA	0.7031	0.7973	0.7473
		EDR+MT-Noun	0.7027	0.7945	0.7458
		EDR+MT-NVAA	**0.7062**	0.7956	0.7482
Proposed (Simultaneous sentence split and alignment)	Hard	None	0.4892	0.8966	0.6330
		EDR	0.4896	0.9027	0.6349
		MT-Noun	0.4863	0.8996	0.6313
		MT-NVAA	0.4970	0.9068	0.6421
		EDR+MT-Noun	0.4957	0.9016	0.6397
		EDR+MT-NVAA	0.4960	0.9018	0.6400
	Hard+Soft	None	0.6709	0.8906	0.7653
		EDR	0.6717	0.8773	0.7608
		MT-Noun	0.6988	0.8873	0.7819
		MT-NVAA	0.7005	0.8903	**0.7840**
		EDR+MT-Noun	0.6958	0.8847	0.7789
		EDR+MT-NVAA	0.6961	0.8845	0.7791

Table 4: Sentence alignment results without tuning.

Tuning Method	Dictionary	Similarity Threshold	Precision	Recall	F
maximum F measure @tuning	None	0.14	0.6901	0.8572	0.7646
	EDR	0.20	0.6883	0.8616	0.7653
	MT-Noun	0.17	0.7111	**0.8841**	0.7883
	MT-NVAA	0.28	0.7361	0.8600	0.7932
	EDR+MT-Noun	0.32	**0.7544**	0.8414	0.7955
	EDR+MT-NVAA	0.32	0.7543	0.8443	**0.7968**
precision $\geq$ 0.9 @tuning	None	0.46	0.7842	0.3480	0.4820
	EDR	0.63	**0.9566**	0.1354	0.2372
	MT-Noun	0.50	0.8976	**0.4090**	**0.5619**
	MT-NVAA	0.55	0.8843	0.3879	0.5393
	EDR+MT-Noun	0.59	0.8987	0.2871	0.4352
	EDR+MT-NVAA	0.66	0.9272	0.1824	0.3049

Table 5: Sentence alignment results of the proposed method “Hard+Soft” with similarity threshold tuned on the development set for maximum F measure and precision $\geq$ 0.9.

Dictionary	Coverage
None	0.27
EDR	0.39
MT-Noun	0.42
MT-NVAA	0.45
EDR+MT-Noun	0.46
EDR+MT-NVAA	0.48

Table 6: Lexicons coverage on the test set.

segmentation and alignment issues, taking advantage of intermediate alignment data to adjust sentence boundaries.

Lexicon-based algorithms require a dictionary that may be generated offline (Wu, 1994; Ma, 2006), or online, for example by comparing the number of occurrences and distribution of words on both sides of the bilingual corpus (Kay and Röscheisen, 1993). Our proposed method follows an intermediate approach, where lexicons are generated automatically using a pivot-based MT system. Since lexicons are generated from words in the corpus, they achieve a high coverage ratio with a low number of entries. MT systems have previously been used for sentence alignment, by calculating similarity scores between target and MT translation of the source. (Adafre and De Rijke, 2006) uses word-level Jaccard coefficient measure, while (Sennrich

```
Baseline (with hard delimiters)

zh: 据陈宗懋报导,茶茎溃疡病(Phomopsis theae)最早是1917年在斯里兰卡发现,印度、
    肯尼亚、坦桑尼亚、马拉维、乌干达和我国浙江相继报道.
    According To Chen Zong Mao, tea stem canker (Phomopsis theae) was first
    discovered in 1917 in Sri Lanka, it was reported in India, Kenya, Tanzania,
    Malawi, Uganda, and the Zhejiang province in China.

ja: Ｃｈｅｎ　Ｚｏｎｇ　Ｍａｏによると，茶ノ木枝枯病（Ｐｈｏｍｏｐｓｉｓ　ｔｈｅａｅ）は
    １９１７年にスリランカで発見された。
    According To Chen Zong Mao, tea stem canker (Phomopsis theae) was
    discovered in 1917 in Sri Lanka.

ja: 後にインド，ケニア，タンザニア，マラウイ，ウガンダと中国の浙江省で報告された。
    Then it was reported in India, Kenya, Tanzania, Malawi, Uganda, and the
    Zhejiang province in China.
```

```
Baseline (with hard and soft delimiters)

zh: 据陈宗懋报导,
    According To Chen Zong Mao,
________________________________________________________________
zh: 茶茎溃疡病(Phomopsis theae)最早是1917年在斯里兰卡发现,
    tea stem canker (Phomopsis theae) was first discovered in 1917 in Sri Lanka,
ja: Ｃｈｅｎ　Ｚｏｎｇ　Ｍａｏによると，茶ノ木枝枯病（Ｐｈｏｍｏｐｓｉｓ　ｔｈｅａｅ）は
    １９１７年にスリランカで発見された。
    tea stem canker (Phomopsis theae) was discovered in 1917 in Sri Lanka.
________________________________________________________________
zh: 印度、
    India,
________________________________________________________________
zh: 肯尼亚、
    Kenya,
zh: 坦桑尼亚、
    Tanzania,
zh: 马拉维、
    Malawi,
zh: 乌干达和我国浙江相继报道.
    it was reported in Uganda and the Zhejiang province in China.

ja: 後にインド，ケニア，タンザニア，マラウイ，ウガンダと中国の浙江省で報告された。
    Then it was reported in India, Kenya, Tanzania, Malawi, Uganda, and the
    Zhejiang province in China.
```

```
Proposed (with hard and soft delimiters)

zh: 据陈宗懋报导,茶茎溃疡病(Phomopsis theae)最早是1917年在斯里兰卡发现,
    According To Chen Zong Mao, tea stem canker (Phomopsis theae) was first
    discovered in 1917 in Sri Lanka,

ja: Ｃｈｅｎ　Ｚｏｎｇ　Ｍａｏによると，茶ノ木枝枯病（Ｐｈｏｍｏｐｓｉｓ　ｔｈｅａｅ）は
    １９１７年にスリランカで発見された。
    According To Chen Zong Mao, tea stem canker (Phomopsis theae) was
    discovered in 1917 in Sri Lanka.
________________________________________________________________
zh: 印度、肯尼亚、坦桑尼亚、马拉维、乌干达和我国浙江相继报道.
    It was reported in India, Kenya, Tanzania,
    Malawi, Uganda, and the Zhejiang province in China.

ja: 後にインド，ケニア，タンザニア，マラウイ，ウガンダと中国の浙江省で報告された。

    Then it was reported in India, Kenya, Tanzania, Malawi, Uganda, and the
    Zhejiang province in China.
```

Figure 2: Sentence alignment output of baseline and proposed methods

and Volk, 2010) uses modified BLEU score as similarity score.

Current lexicon-based algorithms also require the input to be segmented into words (Ma, 2006). However, like sentence segmentation, word segmentation is not a trivial issue especially for Asian languages like Chinese or Japanese with no explicit word boundaries (spaces), and matching rate between the input words and lexicon entries decreases if different segmentation rules are used. Our proposed method does not require the input to be segmented into words, using the longest match against lexicons instead as matching unit.

Finally our method also outputs the similarity score for each alignment pair, which can be used to filter out pairs with low similarity, i.e., pairs with bad or omitted word-level translation. Current studies do not take into account the level of similarity of alignment pairs when evaluating alignment methods.

6. Conclusion

In this paper, we proposed a simultaneous sentence boundary detection and alignment method with pivot-based MT generated lexicons. We verified the effectiveness of our proposed method on a Chinese-Japanese scientific domain sentence alignment task. The alignment tool used in this paper will be available as an open source software. In future work, we plan to apply the method on large sentence alignment tasks and evaluate the MT performance trained on the aligned sentences.

7. Bibliographical References

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

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176. Association for Computational Linguistics.

Chu, C., Nakazawa, T., Kawahara, D., and Kurohashi, S. (2013). Chinese-japanese machine translation exploiting chinese characters. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):16:1–16:25.

Dabre, R., Chu, C., Cromieres, F., Nakazawa, T., and Kurohashi, S. (2015). Large-scale dictionary construction via pivot-based statistical machine translation with significance pruning and neural network features. In *Proceedings of the 29th Pacific Asia Conference on Language, Information, and Computation*, Shanghai, China, October.

Jin, M., Kim, M.-Y., Kim, D., and Lee, J.-H. (2004). Segmentation of chinese long sentences using commas. In Oliver Streiter et al., editors, *ACL SIGHAN Workshop 2004*, pages 1–8, Barcelona, Spain, July. Association for Computational Linguistics.

Johnson, H., Martin, J., Foster, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June. Association for Computational Linguistics.

Kay, M. and Röscheisen, M. (1993). Text-translation alignment. *Computational linguistics*, 19(1):121–142.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.

Li, P., Sun, M., and Xue, P. (2010). Fast-champollion: a fast and robust sentence alignment algorithm. In *Pro-*

ceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 710–718. Association for Computational Linguistics.

Ma, X. (2006). Champollion: A robust parallel text sentence aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pages 489–492.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *AMTA*, volume 2499, pages 135–144.

Sennrich, R. and Volk, M. (2010). Mt-based sentence alignment for ocr-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010), Denver, Colorado*.

Wu, D. (1994). Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 80–87. Association for Computational Linguistics.

Xue, N. and Yang, Y. (2011). Chinese sentence segmentation as comma classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 631–635. Association for Computational Linguistics.