VNU Journal of Science: Comp. Science & Com. Eng. Vol. 31. No. 1 (2015) 32–44

A New Feature to Improve Moore's Sentence Alignment Method

Hai-Long Trieu[1] Phuong-Thai Nguyen[2] Le-Minh Nguyen[1]

[1]*Japan Advanced Institute of Science and Technology, Ishikawa, Japan*
[2]*VNU University of Engineering and Technology, Hanoi, Vietnam*

Abstract

The sentence alignment approach proposed by Moore, 2002 (M-Align) is an effective method which gets a relatively high performance based on combination of length-based and word correspondences. Nevertheless, despite the high *precision*, M-Align usually gets a low *recall* especially when dealing with sparse data problem. We propose an algorithm which not only exploits advantages of M-Align but overcomes the weakness of this baseline method by using a new feature in sentence alignment, word clustering. Experiments shows an improvement on the baseline method up to 30% *recall* while *precision* is reasonable.

Manuscript communication: received 17 June 2014, revised 4 january 2015, accepted 19 January 2015
Corresponding author: Trieu Hai Long, trieulh@jaist.ac.jp

Keywords: Sentence Alignment, Parallel Corpora, Word Clustering, Natural Language Processing

1. Introduction

Online parallel texts are ample and substantial resources today. In order to apply these materials into useful applications like machine translation, these resources need to be aligned at sentence level. This is the task known as sentence alignment which maps sentences in the text of the source language to their corresponding units in the text of the target language. After aligned at sentence level, the bilingual corpora are greatly useful in many important applications. Efficient and powerful sentence alignment algorithms, therefore, become increasingly important.

The sentence alignment approach proposed by Moore, 2002 [14] is an effective method which gets a relatively high performance especially in *precision*. Nonetheless, this method has a drawback that it usually gets a low *recall* especially when dealing with sparse data problem. In any real text, sparseness of data is an inherent property, and it is a problem that aligners encounter in collecting frequency statistics on words. This may lead to an inadequate estimation probabilities of rare but nevertheless possible words. Therefore, reducing unreliable probability estimates in processing sparse data is also a solution to improve the quality of aligners. In this paper, we propose a method which overcomes weaknesses of the Moore's approach by using a new feature in sentence alignment, word clustering. In the Moore's method, a bilingual word dictionary is built by using IBM Model 1, which mainly effects on performance of the aligner. However, this dictionary may lack a large number of vocabulary when input corpus contains sparse data. Therefore, in order to deal with this problem, we propose an algorithm which applies monolingual word clustering to enrich the dictionary in such case. Our approach obtains a high *recall* while the accuracy is still relatively high, which leads to a considerably better overall performance than the baseline

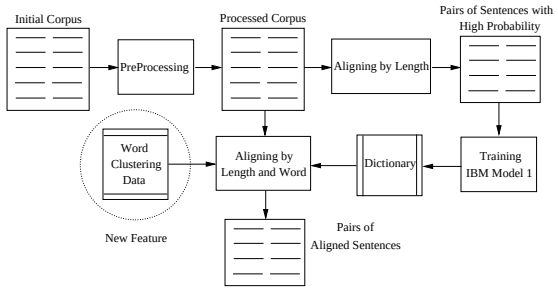

Fig 1: Framework of our sentence alignment algorithm.

method [14].

In the next section, we present our approach and sentence alignment framework. Section 3 indicates experimental results and evaluations on our algorithm compared to the baseline method. Section 4 is a survey of related works. Finally, Section 5 gives conclusions and future works.

2. Our Method

Our method is based on the framework of the Moore's algorithm [14], which is presented in section 2.1. Section 2.2 illustrates our analyses and evaluations impacts of dictionary quality to performance of the sentence aligner. We briefly introduce to word clustering (Section 2.3) and using this feature to improve the Moore's method (Section 2.4). An example is also included in this section to illustrate our algorithm more detail.

2.1. Sentence Alignment Framework

We use the framework of the Moore's algorithm [14] with some modifications. This framework consists of two phases. Firstly, input corpus is aligned based on a sentence-length model in order to extract sentence pairs with high probability to train word alignment model (IBM Model 1). In the second phase, the corpus is aligned again based on a combination of length-based and bilingual word dictionary. Word clustering is used in the second phrase to improve sentence alignment quality. Our approach is illustrated in the Fig. 1.

2.2. Effect of Bilingual Word Dictionary

Sentence aligners based on the combination length-based and word correspondences usually use bilingual word dictionary. Moore [14] uses IBM Model 1 to make a bilingual word dictionary. Varga, et al. [20] use an extra dictionary or train IBM Model 1 to make a dictionary in the case of absence such a resource. Let $(s,\ t)$ is a pair of sentences where s is a sentence of source language, t is a sentence of target language.

$s = (s_1, s_2, ..., s_l)$, where s_i is words of sentence s.

$t = (t_1, t_2, ..., t_m)$, where t_j is words of sentence t.

To estimate alignment probability for this sentence pair, all word pairs (s_i, t_j) are searched in bilingual word dictionary. However, the more input corpus contains sparse data, the more these word pairs are not contained in the dictionary. In the Moore's method [14], words which are not included in the dictionary are simply replaced by an only term "*(other)*".

In the Moore's method, word translation is applied to evaluate alignment probability as formula below:

$$P(s,t) = \frac{P_{1-1}(l,m)}{(l+1)^m}\left(\prod_{j=1}^{m}\sum_{i=0}^{l} t(t_j|s_i)\right)\left(\prod_{i=1}^{l} f_u(s_i)\right) \quad (1)$$

Where m is the length of t, and l is the length of s; $t(t_j|s_i)$ is word translation probability of word pair (t_j, s_i); and f_u is the observed relative unigram frequency of the word in the text of corresponding language.

In the below section, we will analyse how the Moore's method makes errors when word pairs are absent in dictionary, or sparse data problem.

According to the Moore's method, when s_i or t_j is not included in dictionary, it is replaced by one of pairs: $(t_j$,"*(other)*"), ("*(other)*", s_i), or ("*(other)*","*(other)*"). Suppose that the correct translation probability of the word pair (t_j, s_i) is ρ, and the translation probabilities

Algorithm 1: Generating Bilingual Word Dictionary

Input : set of sentence pairs (***s,t***)
Output: translation prob. $t(e, f)$

```
1  begin
2      initialize t(e|f) uniformly
3      while not converged do
4          //initialize
5          count(e|f) = 0 for all e, f
6          total(f) = 0 for all f
7          for all sentence pairs (s,t) do
8              //compute normalization
9              for all words e in s do
10                 total(e) = 0
11                 for all words f in t do
12                     total(e)+ = t(e|f)
13             //collect counts
14             for all words e in s do
15                 for all words f in t do
16                     count(e|f)+ = t(e|f) / total(e)
17                     total(f)+ = t(e|f) / total(e)
18         //estimate probabilities
19         for all words f do
20             for all words e do
21                 t(e|f) = fraccount(e|f)total(f)
22     return t(e|f)
```

of the word pair $(t_j$,"$(other)$"), ("$(other)$", $s_i)$, ("$(other)$","$(other)$") are ρ_1, ρ_2, ρ_3 respectively. These estimations make errors as follows:

$$\epsilon_1 = \rho - \rho_1; \epsilon_2 = \rho - \rho_2; \epsilon_3 = \rho - \rho_3; \quad (2)$$

Therefore, when (t_j, s_i) is replaced by one of these word pairs: $(t_j$,"$(other)$"), ("$(other)$", $s_i)$, ("$(other)$","$(other)$"), the error of this estimation $\varepsilon_i \in \{\epsilon_1, \epsilon_2, \epsilon_3\}$ effects to the correct estimation by a total error ω:

$$\omega = \prod_{j=1}^{m} \sum_{i=0}^{l} \varepsilon_i \quad (3)$$

If (t_j, s_i) is contained dictionary, $\varepsilon_i = 0$; suppose that there are k, $(0 \leq k \leq l + 1)$, word pairs which are not included in dictionary, and the error average is μ; then the total error is:

$$\omega = (k * \mu)^m; \quad (4)$$

The more word pairs which are not included in dictionary, the more the number of word pairs k, or total error ω.

2.3. Word Clustering

Brown's Algorithm. Word clustering Brown, et al. [3] is considered as a method for estimating the probabilities of low frequency events that are likely unobserved in an unlabeled data. One of aims of word clustering is the problem of predicting a word from previous words in a sample of text. This algorithm counts the

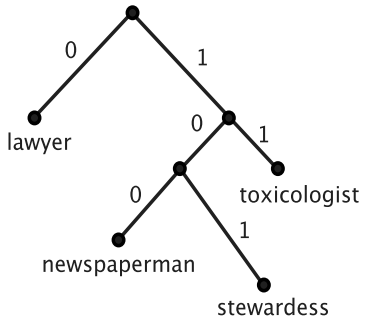

Fig 2: An example of Brown's cluster algorithm

similarity of a word based on its relations with words on left and the right of it. Input to the algorithm is a corpus of unlabeled data which consists of a vocabulary of words to be clustered. Initially, each word in the corpus is considered to be in its own distinct cluster. The algorithm then repeatedly merges pairs of clusters that maximizes the quality of the clustering result, and each word belongs to exactly one cluster until the number of clusters is reduced to a predefined number. Output of the word cluster algorithm is a binary tree as shown in Fig. 2, in which the leaves of the tree are the words in the vocabulary. A word cluster contains a main word and several subordinate words. Each subordinate word has the same bit string and corresponding frequency.

2.4. Proposed Algorithm

We propose using word clustering data to supplement lexical information for bilingual word dictionary and improve alignment quality. We use the hypothesis that same cluster have a specific correlation, and in some cases they are able to be replaced to each other. Words that disappear in the dictionary would be replaced other words of their cluster rather than replacing all of those words to an only term as in method of Moore [14]. We use two word clustering data sets corresponding to the two languages in the corpus. This idea is indicated at the *Algorithm 2*.

In *Algorithm 2*, D is bilingual word dictionary created by training IBM Model 1. The dictionary D contains word pairs (e, v) in which each word belongs to texts of source and target languages correspondingly, and $t(e, v)$ is their word translation probability.

In addition, C_e and C_v are two data sets clustered by word of texts of source and target languages respectively. C_e is the cluster of the word e, and C_v is the cluster of the word v.

When the word pair (e, v) is absent in the dictionary, e and v are replaced by all words of their cluster. A combined value of probability of new word pairs is counted, and it is treated as alignment probability for the absent word pair (e, v). In this algorithm, we use average function to get this combined value.

Consider an English-Vietnamese sentence pair as indicated in Table 1.

Some word pairs of bilingual word dictionary are listed in Table 2.

Consider a word pairs which is not contained in the Dictionary: (*act*, *chức_năng*). In the first step, our algorithm returns clusters of each word in this pair. The result is shown in Table 3 and Table 4.

Table 1: An English-Vietnamese sentence pair

damodaran ' s solution is gelatin hydrolysate , a protein known to act as a natural antifreeze .
giải_pháp của damodaran là chất thủy_phân gelatin , một loại protein có chức_năng như chất chống đông tự_nhiên .

Table 2: Several word pairs in Dictionary

damodaran	damodaran	0.22
's	của	0.12
solution	giải_pháp	0.03
is	là	0.55
a	một	0.73
as	như	0.46

Table 3: Cluster of *act*

0110001111	**act**
0110001111	society
0110001111	show
0110001111	departments
0110001111	helps

Algorithm 2: Sentence Alignment Using Word Clustering

Input : A word pair (e, v), Dictionary D, Clusters C_e and C_v
Output: Word translation prob. of (e, v)

begin
  if *(e, v) contained in D* **then**
    $P = t(e, v)$
  else
    if *(e contained in D) and (v contained in D)* **then**
      with all $(e_1,..., e_n)$ in C_e
      with all $(v_1,..., v_m)$ in C_v
      if *$((e_i, v)$ contained in D) or $((e, v_j)$ contained in D)* **then**
        $P = \frac{1}{n+m}(\sum_{i=1}^{n} t(e_i, v) + \sum_{j=1}^{m} t(e, v_j))$
      else
        $P = t(\text{"}(other)\text{"}, \text{"}(other)\text{"})$
    else
      if *(e contained in D) or (v contained in D)* **then**
        if *(e contained in D)* **then**
          with all $(v_1,..., v_m)$ in C_v
          if *(e, v_j) contained in D* **then**
            $P = \frac{1}{m}\sum_{i=1}^{m} t(e, v_j)$
          else
            $P = \frac{1}{m}\sum_{i=1}^{m} t(e, \text{"}(other)\text{"})$
        else
          with all $(e_1,..., e_n)$ in C_e
          if *(e_i, v) contained in D* **then**
            $P = \frac{1}{n}\sum_{i=1}^{n} t(e_i, v)$
          else
            $P = \frac{1}{n}\sum_{i=1}^{n} t(\text{"}(other)\text{"}, v)$
      else
        $P = t(\text{"}(other)\text{"}, \text{"}(other)\text{"})$
  return P

Table 4: Cluster of *chức_năng*

11111110	**chức_năng**
11111110	hành_vi
11111110	phạt
11111110	hoạt_động
...	

The bit strings “0110001111" and “11111110" are identification of the clusters. Word pairs of these two clusters are then searched in the Dictionary as shown in Table 5.

Table 5: Word pairs are searched in Dictionary

departments	**chức_năng**	9.15E-4
act	hành_vi	0.43
act	phạt	7.41E-4
act	hoạt_động	0.01

In the next step, the algorithm returns a translation probability for the initial word pair (*act*, *chức_năng*).

Table 6: Probability of the word pair (*act*, *chức_năng*)

Pr(*act*,*chức_năng*) 0.43,	**=average of** (9.15E-4, 7.41E-4, 0.01) = 0.11

3. Experiments

In this section, we evaluate performance of our algorithm and compare to the baseline method (M-Align).

3.1. Data

3.1.1. Bilingual Corpora

The test data of our experiment is English-Vietnamese parallel data extracted from some websites including World Bank, Science, WHO, and Vietnamtourism. The data consist of 1800 English sentences (En Test Data) with 39526 words (6333 distint words) and 1828 Vietnamese sentences (Vi Test Data) with 40491 words (5721 distinct words). These data sets are shown in Table 7. We align this corpus at the sentence level manually and obtain 846 bilingual sentences pairs. We use data from VLSP project available at[1]including 100,836 English-Vietnamese sentence pairs (En Training Data and Vi Training Data) with 1743040 English words (36149 distinct words) and 1681915 Vietnamese words (25523 distinct words). The VLSP data consists of 80,000 sentence pairs in Economics-Social topics and 20,000 sentence pairs in information technology topic.

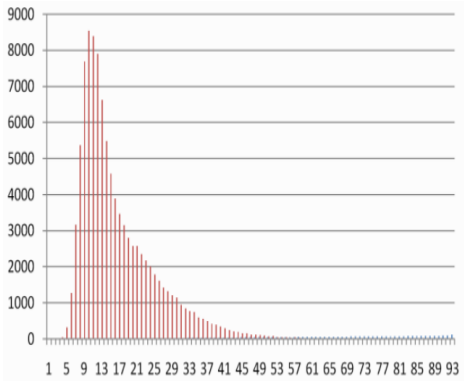

Fig 3: Frequencies of Vietnamese Sentence Length

Table 7: Bilingual Corpora

	Sentences	**Vocabularies**
En Training Data	100038	36149
Vi Training Data	100038	25523
En Test Data	1800	6333
Vi Test Data	1828	5721

We conduct lowercase, tokenize, word segmentation these data sets using the tool of[1].

3.1.2. Sentence Length Frequency

The frequencies of sentence length are described in Fig. 3 and Fig. 4. In these figures, the horizontal axis describe sentence lengths, and the vertical axis describe frequencies. The average sentence lengths of English and Vietnamese are 17.3 (English), 16.7 (Vietnamese), respectively.

[1]`http://vlsp.vietlp.org:8080/demo/?page=resources`

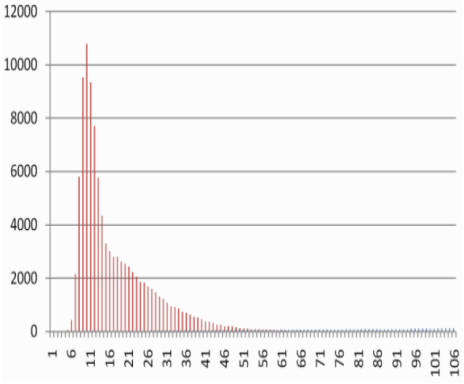

Fig 4: Frequencies of English Sentence Length

3.1.3. Word Clustering Data.

We use the two word clustering data sets of English and Vietnamese as indicated in Table 8. To get these data sets, we use two monolingual data sets of English (BNC corpus) and Vietnamese (crawling from the web) and apply Brown's word clustering. English BNC corpus (British National Corpus) we use including 1044285 sentences (approximately 22 million words). We get Vietnamese data set from the Viettreebank data including 700,000 sentences (about 15 million words) of topics Political-Social, and the rest of data is crawled from websites laodong, tuoitre, and PC world.

Table 8: Input Corpora for Training Word Clustering

	Sentences	Vocabularies
En Data	1044285	223841
Vi Data	700000	180099

We apply word cluster algorithm (Brown, et al. [3]) with 700 clusters for both English and Vietnamese monolingual data. Vocabulary of clustering data sets cover 82.96% and 81.09% of English and Vietnamese sentence alignment corpus respectively, indicated in Table 9. Vocabulary of these word clustering data sets cover 90.31% and 91.82% of English and Vietnamese vocabulary in bilingual word dictionary created by training IBM Model 1.

Table 9: Word clustering data sets.

	Clusters	Dictionary Coverage	Corpus Coverage
En Data	700	90.31%	82.96%
Vi Data	700	91.82%	81.09%

3.2. Metrics

We use the following metrics for evaluation: *precision*, *recall* and *F-measure* to evaluate sentence aligners. The metric *precision* is defined as the fraction of retrieved documents that are in fact relevant. The metric *recall* is defined as the fraction of relevant documents that are retrieved by the algorithm. The *F-measure* characterizes the combined performance of *recall* and *precision* [7].

$$precision = \frac{CorrectSents}{AlignedSents}$$

$$recall = \frac{CorrectSents}{HandSents}$$

$$F\text{-}measure = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Where:

CorrectSents: number of sentence pairs aligned by the algorithm match those manually aligned.

AlignedSents: number of sentence pairs aligned by the algorithm.

HandSents: number of sentence pairs manually aligned.

3.3. Evaluations

We conduct experiments and compare our approach (EVS) to the baseline algorithm: M-Align (Bilingual Sentence Aligner[2], Moore [14]). As mentioned in the previous sections, the range of vocabulary in this dictionary considerably affects to the final alignment result because it is related to translation probabilities estimated in this dictionary. The more vocabulary in dictionary, the better the alignment result. The Moore's method sets the threshold 0.99 for the

[2] http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656

length-based phrase. We evaluate the impact of size of dictionary by setting a range of threshold of length-based phrase, from 0.5 to 0.99. We use the same threshold 0.9 as in the Moore's method to ensure the high reliability.

Firstly, we assess our approach compared with the baseline method (M-Align) in term of *precision*. M-Align is usually evaluated as an effective method with high accuracy; it is better than our approach about 9% in *precision*, Fig. 5. In the threshold 0.5 of the length-based phase, EVS gets a *precision* by 60.99% while that of M-Align is 69.30%. In general, the *precision* gradually increases according to thresholds of the initial alignment. When the threshold is set as 0.9, both approaches get the highest *precision*, 62.55% (our approach) and 72.46% (M-Align). The *precision* of the Moore's method is generally higher than that of our approach; however, the difference is not considerable.

As mentioned in the section of metrics, *precision* is counted by ratio of number of true sentence pairs (sentence pairs aligned by aligner match with those aligned manually) and the total of sentence pairs aligned by aligner. Let a_1 and b_1 be true sentence pairs and total sentence pairs created by M-align, respectively. Also, let a_2 and b_2 be true sentence pairs and total sentence pairs created by EVS, respectively. Then, the *precision* of these two methods are:

a_1/b_1 (M-Align), a_2/b_2 (EVS)

In our method, because of using word cluster features, the aligner discovers much more sentence pairs than that of M-Align both of a_2 and b_2. In other word, a_1 and b_1 are really lower than a_2 and b_2, which leads to the difference in the ratio between them (a_1/b_1 and a_2/b_2). In this method, our goal is to apply word cluster to deal with problem of sparse data that improves *recall* considerably while the *precision* is still reasonable. We will describe the improvement in term of *recall* below.

The corpus we use in experiments is crawled from English-Vietnamese bilingual websites, which contains sparse data. The Moore's method encounters an ineffective performance especially in term of *recall*, Fig. 7. At the threshold of 0.5,

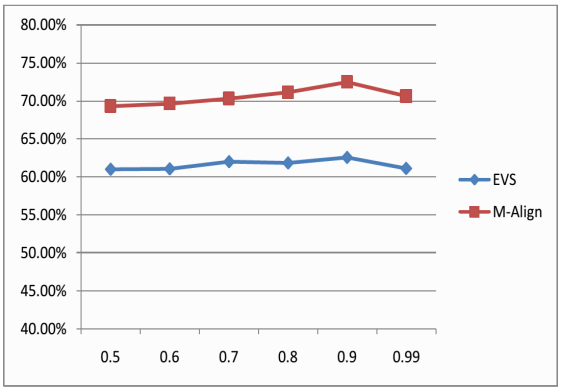

Fig 5: Comparision in Precision of proposed and baseline approaches.

the *recall* of M-Align is 51.77%, and it gradually reduces at higher thresholds.

By using word clustering data, we not only exploit some characteristics of word clustering for sentence alignment but reduce error of the Moore's method. The comparison between our method and the baseline method is shown in Fig. 7. Our approach gets a *recall* significantly higher than that of M-Align, up to more than 30%. In the threshold of 0.5, the *recall* is 75.77% of EVS and 51.77% of M-Align while that is 74.35% (EVS) and 43.74% (M-Align) in the threshold of 0.99. In our approach, the *recall* fluctuates insignificantly with the range about 73.64% to 75.77% because of the contribution of using word clustering data. Our approach deals with the sparse data problem effectively. If the quality of the dictionary is good enough, the algorithm can get a rather high performance. Otherwise, using word clustering data can contribute more translation word pairs by mapping them through their clusters, and help to resolve sparse data problem rather thoroughly.

Because our approach significantly improves *recall* compared to M-Align while the *precision* of EVS is inconsiderably lower than that of M-Align, our approach obtains the *F-measure* relatively higher than M-Align (Fig. 8). In the threshold of 0.5, *F-measure* of our approach is 67.58% which is 8.31% higher than that of M-Align (59.27%). Meanwhile, in the threshold of 0.99, the increase of *F-measure* attains the highest

English Sentence	scientists conducted a series of tests to see how **horseflies** also known as **tabanids** , reacted to the light reflected by solid black , **brown-grey** and white horses , as well as the vertical stripes of a **zebra** .
Vietnamese Sentence	các nhà_khoa_học thực_hiện một loạt các thí_nghiệm để thấy tại_sao các ruồi_trâu phản_ứng với ánh_sáng phản_chiếu từ những con **ngựa_ô** (đen) , **nâu-xám** và **ngựa_bạch** (trắng) , cũng_như những vằn thẳng_đứng của ngựa_vằn .

Fig 6: An English-Vietnamese sentence pair

rate (13.08%) when *F-measure* are 67.09% and 54.01% of EVS and M-Align respectively.

We will discuss contribution of using word clustering by an example described below. Consider the English-Vietnamese sentence pair as shown in Fig. 6. This sentence pair is a correct result of our algorithm, but the Moore's method can not return it.

In these two sentences, words are not contained in Dictionary including: **horseflies**, **tabanids**, **brown-grey**, **zebra** (English); **ngựa_ô**, **nâu-xám**, **ngựa_bạch** (Vietnamese).

In counting alignment probability of a sentence pair, there has to look up each word in the English sentence to all word the Vietnamese sentence and vice versa. We describe this by analyzing word translation probabilities of all words of the English sentence to the Vietnamese word **ngựa_ô** which is indicated in Table 10.

Table 10 illustrates word probabilities of all word pairs (e_i, *ngựa_ô*) looked up from Dictionary where e_i is one word of the English sentence, $1 \leq i \leq 40$. P_1 describes word translation probability produced by our approach while P_2 describes that produced by the Moore's method. There are some notations in Table 10:

- (): means that this probability made by using word clustering. (replacing *ngựa_ô* by words in cluster of *ngựa_ô*)
- *: means that this probability made by referring probability of the word pair (e_i,

Table 10: $P(e_i, ngựa\_ô)$

i	e_i	P_1	P_2
1	scientists	(0.1277)	0
2	conducted	0	0
3	a	*0.0017	0
4	series	*0.0508	*0.003
5	of	*0.0080	0
6	tests	(0.0032)	0
7	to	0	0
8	see	0	0
9	how	0	0
10	horseflies	**0.6327	**
11	,	*0.004	*0.0049
12	also	(0.002)	*0.0007
13	known	(0.072)	*0.0003
14	as	*5.3991E-4	*0.0001
15	tabanids	**0.633	**0.123
16	,	*0.004	*0.0049
17	reacted	0	**0.123
18	to	0	0
19	the	*0.006	0
20	light	*0.007	0
21	reflected	(0.017)	0
22	by	0	0
23	solid	0	0
24	black	(0.0076)	*0.0017
25	,	*0.004	*0.0049
26	brown-grey	**0.633	**0.123
27	and	*1.9661E-4	*4.714E-5
28	white	(0.0076)	0
29	horses	0	0
30	,	*0.004	*0.0049
31	as	*5.3991E-4	*0.0001
32	well	(0.0137)	0
33	as	*5.3991E-4	*0.0001
34	the	*0.006	0
35	vertical	*0.0495	0
36	stripes	(0.0511)	**0.123
37	of	*0.0080	0
38	a	*0.0017	0
39	zebra	**0.633	**0.123
40	.	*0.0055	0

(other)) in Dictionary. (replacing *ngựa_ô* by *(other)*)

- **: means that this probability made by referring probability of the word pair (*(other)*, *(other)*) in Dictionary. (replacing both e_i and *ngựa_ô* by *(other)*)

In this table, from the column of P_1 (probabilities produced by our approach),there are probabilities of 40 word pairs including probabilities of 9 word pairs produced by using word clustering, 18 word pairs produced by replacing *ngựa_ô* by *(other)*, 4 word pairs produced by replacing both e_i *ngựa_ô* by *(other)*, and 9 word pairs by zero (probability by zero means that the word pair (e_i, v_j) is not contained in Dictionary even when replacing e_i, v_j by *(other)*). Meanwhile, from the column of P_2 (probabilities produced by the Moore's method),there are probabilities of 12 word pairs produced by replacing *ngựa_ô* by *(other)*, 6 word pairs produced by replacing both e_i and *ngựa_ô* by *(other)*, and 22 word pairs by zero. There are a large number of word pairs that probabilities by zero produced by the Moore's method (22 word pairs) while we use word clustering to count probabilities of these word pairs and get 5 word pairs from word clustering and 9 word pairs from replacing *ngựa_ô* by *(other)*). By using word clustering, we overcome major part of word pairs that probabilities are by zero, which effect alignment result. We show some of word pairs using word clustering to count translation probabilities as Table 11, 12, 13.

Table 11: Word Cluster of *ngựa_ô*

01100101110	**ngựa_ô**	1
01100101110	bé_tí	1
01100101110	ruồi_trâu	1
01100101110	binh_lính	1
01100101110	lạc_đà	1
01100101110	dương_cầm	2
01100101110	gia	12
01100101110	giỏi	181
01100101110	gọi_là	2923
...		

Table 12: P(*well*, *ngựa_ô*)

well	ruồi_trâu	0.0137
well	gia	0.0137
well	giỏi	0.0137
P(*well*, *ngựa_ô*)	=	0.0137

Table 13: P(*known*, *ngựa_ô*)

known	bé_tí	0.0049
known	ruồi_trâu	0.0724
known	gia	0.0724
known	gọi_là	0.1399
P(*known*, *ngựa_ô*)	=	0.0724

4. Related Works

In various sentence alignment algorithms which have been proposed, there are three widespread approaches which are based on respectively a comparison of sentence length, lexical correspondence, and a combination of these two methods.

The length-based approach is based on modeling the relationship between the lengths (number of characters or words) of sentences that are mutual translations. This method is based on the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences. The algorithms of this type were first proposed in (Brown, et al., 1991 [2]) and (Gale and Church, 1993 [6]). These algorithms use sentence-length statistics in order to model the relationship between groups of sentences that are translations of each other. Wu (Wu, 1994) also uses the length-based method by applying the algorithm proposed by Gale and Church, and further uses lexical cues from corpus-specific bilingual lexicon to improve alignment. These algorithms are based solely on the lengths of sentences, so they require almost no prior knowledge. Furthermore, when aligning texts whose languages have a high length correlation such as English, French, and German, these approaches are especially useful and work remarkably well. The Gale and Church

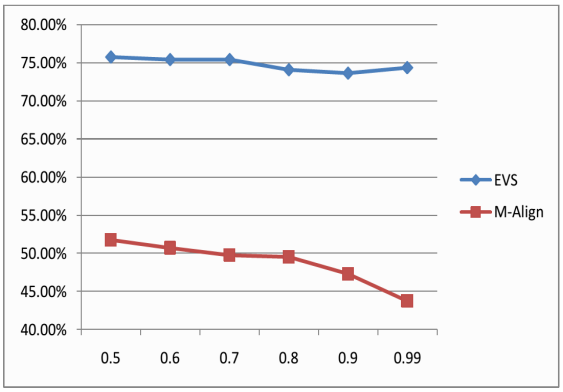

Fig 7: Comparision in Recall of proposed and baseline approaches.

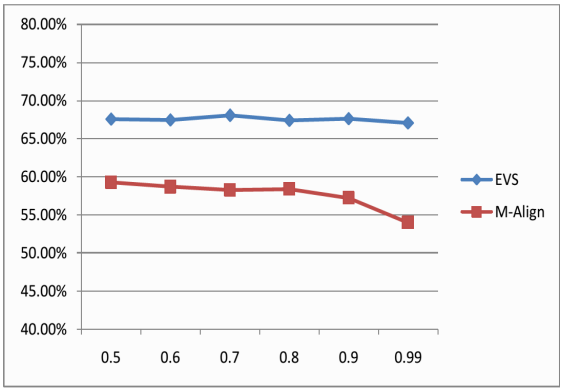

Fig 8: Comparision in F-measure of proposed and baseline approaches.

algorithm is still widely used today, for instance to align Europarl (Koehn, 2005). Nevertheless, this method is not robust and will no longer be reliable if there exists too much noise in input bilingual texts. The algorithm of Brown et al., 1991 requires corpus-dependent anchor points while the method proposed by Gale and Church, 1993 depends on prior alignment of paragraphs to constrain searching alignment. When length correlation of texts breaks down, such as Chinese-English parallel texts, performance of length-based algorithms declines quickly.

Another approach tries to overcome disadvantages of length-based methods by using lexical information from translation lexicons, and/or through the recognition of cognates. Most algorithms match content in one text with their correspondences in the other text, and use these matches as anchor points to align sentences. Meanwhile, some algorithms use cognates (words in language pairs that resemble each other phonetically) rather than the content of word pairs to determine alignments. This method is shown in Kay and Röscheisen, 1993 [8]; Chen, 1993 [4]; Melamed, 1996 [12]; and Ma, 2006 [11]. Kay's work has not proved efficient enough to be suitable for large corpora while Chen constructs a word-to-word translation model during alignment to evaluate probability of an alignment. Word correspondence was further developed in IBM Model (Brown et al., 1993) for statistical machine translation. Melamed, 1996 proposes using geometric correspondence for sentence alignment. The method of word correspondences gets higher accuracy than the length-based method because of using lexical information from source and translation lexicons rather than only sentence length parameter. Nevertheless, in term of speed, this method is slower since it requires considerably more expensive computation. In addition, the method depends on cognates or a bilingual lexicon, for instance the algorithm of Chen requires an initial bilingual lexicon while Melamed's algorithm depends on cognates in the two languages to suggest word correspondences.

The third method is a combination of length-based and word correspondences. This method is proposed in Moore, 2002 [14]; Varga et al., 2005 [20]; and Braune and Fraser, 2010 [1]. Moore, 2002 proposes a two-phase algorithm that combines sentence length (word count) and word correspondences by training a bilingual word dictionary using IBM Model-1. Length-based method is used for the first alignment which subsequently serves as training data for a translation model. Finally, the length-based and translation model are combined in a complex similarity score. Varga et al., 2005 sentence length and word correspondences using a dictionary-based translation model in which the dictionary can be manually expanded. The proposal of Braune and Fraser, 2010 is similar to the Moore's except building 1-to-many and many-to-1 alignments rather than focus only on

1-to-1 alignment as the Moore's method. The hybrid method gets a high performance because of combining advantages and overcomes limits of the first two methods.

Some other methods have been proposed for sentence alignment as shown in Sennrich and Volk [16] and Fattah [5]. While Sennrich and Volk [16] use a variant of BLEU in measuring similarity between all sentence pairs, the approach of Fattah [5] is based on classifiers: Multi-Class Support Vector Machine and Hidden Markov Model.

Word/phrase cluster is also effective features to improve performance in many common natural language processing tasks. This type of feature is applied in works such as in named entity recognition (Miller, et al. [13]; Tkachenko and Simanovsky [17]; Lin and Wu [10]), query classification Lin and Wu [10], part-of-speech tagging Owoputi, et al. [15]. Word clustering is also applied in machine translation. Zhao, et al. [22] propose a variant of a spectral clustering algorithm for bilingual word clustering. This method is to build bilingual word clusters using eigenstructure in bilingual feature (word's bilingual context). Meanwhile, our method applies algorithm proposed by Brown, et al. [3] on monolingual word clustering (word's monolingual context) to enrich bilingual lexical table built from using IBM Model 1.

In conclusion, our method uses word clustering (Brown, et al. [3]) on monolingual corpus to improve the hybrid sentence alignment method (Moore [14]) presented in the next section.

5. Conclusions and Future Works

Sentence alignment is an important task in creating bilingual corpora, a valuable resource for many applications. There have been a number of methods proposed to resolve this task in which hybrid method of length-based and word correspondences gets high performance as the Moore's method [14]. A general problem in sentence alignment is existence of sparse data in corpus. Although the Moore's method has a high performance in term of *precision*, it still does not overcome the problem of sparse data effectively leading to a low *recall*. We propose using word clustering data to enrich bilingual word dictionary and help to deal with sparse data problem. The result from experiments shows a significant improvement *recall* and overall performance in our method compared to the baseline (M-Align). This shows that word clustering data can be utilized in sentence alignment to improve performance of aligners.

In future works, we will try to improve the quality of sentence alignment by other methods including using word phrases or better word translation model (IBM Model 4). In addition, we study how to tackle with noisy data in sentence alignment. Exploiting word clustering data in other fields is also a promising direction.

Acknowledgement

This paper has been supported by the VNU project "Exploiting Very Large Monolingual Corpora for Statistical Machine Translation" (code QG.12.49).

References

[1] Braune, Fabienne and Fraser, Alexander, Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora, In Proceedings of the 23rd International Conference on Computational Linguistics: Posters. 81–89 (2010)

[2] Brown, Peter F. and Lai, Jennifer C. and Mercer, Robert L, Aligning sentences in parallel corpora, In Proceedings of the 29th annual meeting on Association for Computational Linguistics. 169–176. Berkeley, California (1991) proceeding2-BrownEtal1991

[3] Brown, Peter F and Desouza, Peter V and Mercer, Robert L and Pietra, Vincent J Della and Lai, Jenifer C., Class-based n-gram models of natural language, Computational linguistics. vol. 18, 4, 467–479 (1992)

[4] Chen, Stanley F., Aligning sentences in bilingual corpora using lexical information, In Proceedings of the 31st annual meeting on Association for Computational Linguistics. 9–16 (1993)

[5] Fattah, Mohamed Abdel, The Use of MSVM and HMM for Sentence Alignment, Journal of Information Processing Systems. vol. 8, 2, 301–314 (2012)

[6] Gale, William A and Church, Kenneth W., A program for aligning sentences in bilingual corpora, Computational linguistics. vol. 19, 1, 75–102 (1993)

[7] Huang, Yuanpeng J., Robert Powers, and Gaetano T. Montelione, Protein NMR recall, precision, and F-measure scores (RPF scores): structure quality assessment measures based on information retrieval statistics, Journal of the American Chemical Society 127.6 (2005): 1665-1674.

[8] Kay, Martin, and Martin Röscheisen, Text-translation alignment, computational Linguistics. vol. 19, 1, 121–142 (1993)

[9] Koehn, P., Statistical machine translation. Cambridge University Press. (2009).

[10] Lin, D., Wu, X., Phrase clustering for discriminative learning. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2 (pp. 1030-1038). Association for Computational Linguistics. (2009, August)

[11] Ma, Xiaoyi., Champollion: a robust parallel text sentence aligner, In LREC 2006: Fifth International Conference on Language Resources and Evaluation. 489–492 (2006)

[12] Melamed, I. D., A geometric approach to mapping bitext correspondence, In Proceedings of the Conference on Empirical Methods in Natural Language Processing. 1–12 (1996)

[13] Miller, S., Guinness, J., Zamanian, A., Name Tagging with Word Clusters and Discriminative Training. In HLT-NAACL (Vol. 4, pp. 337-342). (2004, May).

[14] Moore, Robert C., Fast and Accurate Sentence Alignment of Bilingual Corpora, In Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users. 135–144 (2002)

[15] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N. A., Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In HLT-NAACL (pp. 380-390). (2013).

[16] Sennrich, R., and Volk, M., 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. (2010)

[17] Tkachenko, M., Simanovsky, A., Named entity recognition: Exploring features. In Proceedings of KONVENS (Vol. 2012, pp. 118-127). (2012).

[18] Trieu, H. L., Nguyen, P. T., and Nguyen, K. A., Improving Moore's Sentence Alignment Method Using Bilingual Word Clustering, In Knowledge and Systems Engineering. Springer International Publishing. 149–160 (2014)

[19] Trieu, H.L, Nguyen, T.P.T, Nguyen, P.T, "An Effective Sentence Alignment Algorithm for English-Vietnamese", In Proceeding The 15th National Symposium of Selected ICT Problems, 262-267, 2012 (in Vietnamese).

[20] Varga, Dániel and Németh, László and Halácsy, Péter and Kornai, András and Trón, Viktor and Nagy, Viktor., Parallel corpora for medium density languages, In Proceedings of the RANLP 2005. 590-–596 (2005)

[21] Wu, Dekai., Aligning a parallel English-Chinese corpus statistically with lexical criteria, In Proceedings of the 32nd annual meeting on Association for Computational Linguistics. 80–87 (1994)

[22] Zhao, B., Xing, E. P., Waibel, A., Bilingual word spectral clustering for statistical machine translation. In Proceedings of the ACL Workshop on Building and Using Parallel Texts (pp. 25-32). Association for Computational Linguistics. (2005, June).