A New Feature to Improve Moore's Sentence Alignment Method

Hai-Long Trieu[1], Phuong-Thai Nguyen[2]

[1]Thai Nguyen University of Education, Thainguyen, Vietnam
[2]University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

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 the sparse data problem. We have proposed an algorithm which not only exploit advantages of M-Align but overcomes the weakness of this baseline method by using a new feature in sentence alignment, word clustering. The effectiveness of this proposal is illustrated by results of experiments in both highly recall and reasonable precision rates.

***Index Terms*—Sentence alignment, parallel corpora, word clustering, natural language processing.**

I. Introduction

Online parallel texts, an ample and substantial resource, have a considerable growth today. Nevertheless, in order to apply these materials into useful applications like machine translation, they need to be processed through some stages. Before learning word correspondences from parallel texts, these resources have to be aligned at sentence level, a task known as sentence alignment.This process maps sentences in the text of the source language to their corresponding units in the text of the target language. After processed by sentence alignment, the bilingual corpora are greatly useful in machine translation and many other important applications. Efficient and powerful sentence alignment algorithms, therefore, become increasingly important.

The sentence alignment approach proposed by Moore [10] is an effective method which gets a relatively high performance especially in precision rate. Nonetheless, this method has a drawback that it usually gets a low recall especially when dealing with the sparse data problem. In any real text, sparseness of data is an inherent property, and it is a problem that aligners always encounter in collecting frequency statistics on words. This may lead to an inadequately estimate 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 have proposed a method which overcomes weaknesses of Moore's approach by using a new feature in sentence alignment, bilingual word clustering, to deal with the sparse data issue. The lack of necessary items in the dictionary used in the lexical stage is supplemented by applying word clustering data sets. This approach obtains a high recall rate while the accuracy still gets a reasonable ratio, which gains a considerably better overall performance than above-mentioned methods.

The rest of this paper is structured as follows. Previous works are described in section II. In section III, we present our approach as well as the sentence alignment framework we use and focus on description of applying word clustering feature in our algorithm. Section IV indicates experimental results and evaluations on our algorithm compared with the baseline method (Moore [10]). Finally, Section V gives several conclusions and future works.

II. 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 (Brown, et al. [2], Gale and Church [6]), lexical correspondence (Chen [4], Melamed [9]), and a combination of these two methods (Moore [10], Varga, et al. [13]).

Sentences are aligned by length-based algorithms based on their lengths (measured by character or word). Some of these algorithms are proposed by Brown, et al. [2], and Gale and Church [6]. These methods are based on the idea that long sentences will be translated into long sentences and short ones into short ones. A probabilistic score is assigned to each proposed correspondence of sentences, based on the scaled difference of lengths of the two sentences and the variance of this difference. These algorithm may produce a good alignment on language pairs with high length correlation like French-English and perform in a high speed. Nevertheless, they are not robust since they only use the sentence length information. When there is too much noise in the input bilingual texts, sentence length information will be no longer reliable.

The second method tries to overcome the disadvantages of the length-based approach by using lexical information from translation lexicons, and/or through the recognition of cognates. This approach could be illustrated in Kay and Röscheisen [7], Ma [8], Melamed [9], and Wu [14]. Chen [4], meanwhile, constructs a word-to-word translation model during alignment to assess the probability of an alignment. The approach based on word correspondences is usually more robust than the length-based one because it uses lexical information from source and translation lexicons rather than only sentence length to determine the translation relationship between sentences in the source text and the target text. Nevertheless, algorithms based on a lexicon are slower than those based on sentence length since they require considerably more

expensive computation. In addition, some of them sometimes require extra resources of languages.

Sentence length and lexical information are also combined to achieve more efficient algorithms. Moore [10] describes an algorithm with two major phases. Firstly, a length-based approach is used for an initial alignment. This first alignment plays as training data for a translation model. Moore uses IBM Model 1 to make a bilingual word dictionary. Finally, length-based model is combined with dictionary to make a hybrid approach. Braune and Fraser [1] also propose an algorithm that has some modifications with Moore [10]. Nonetheless, this approach built more 1-to-many and many-to-1 alignments rather than only making 1-to-1 alignments as Moore [10]. The hybrid approaches achieve a relatively high performance that overcome limits of the first two methods and combine their advantages. Nonetheless, there are still drawbacks with some approaches like a quite low precision and memory restriction in the method of Varga, et al. [13] or a low recall in the Moore's method.

In addition to these above-mentioned algorithms, some new methods have been proposed like Sennrich and Volk [11] and Fattah [5]. While Sennrich and Volk [11] 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. In general, the approach based on combination of sentence length and word correspondences is relatively effective in overall like the proposal of Moore [10] whose idea has been referred and expanded by some researches as Varga, et al. [13] and Braune and Fraser [1]. We have applied the sentence alignment framework proposed by Moore's [10] in order to develop our idea, which is presented in the next section.

III. Our Proposal

This section mentions the method we proposed. Our model is based on the framework of Moore [10], which is presented in section A. Section B illustrates our analyses and evaluations about the impacts of dictionary quality to the performance of sentence aligner. We briefly introduce about word clustering,a new feature we propose to use in sentence alignment and give our algorithm using this feature described in section 3.3. An example is also included in this section to illustrate our algorithm.

A. Sentence Alignment Framework

We use the framework of Moore [10] with some modifications based on a combination of length-based and word correspondences, which consists of two phases. Firstly, the corpus is aligned by sentence-length. After extracting sentence pairs with the highest probabilities, a lexical model is trained with sentences just chosen using IBM Model 1. In the second phase, the corpus is aligned based on the hybrid of the first model with lexical information combined with word clustering. Our approach is illustrated in the Fig. 1.

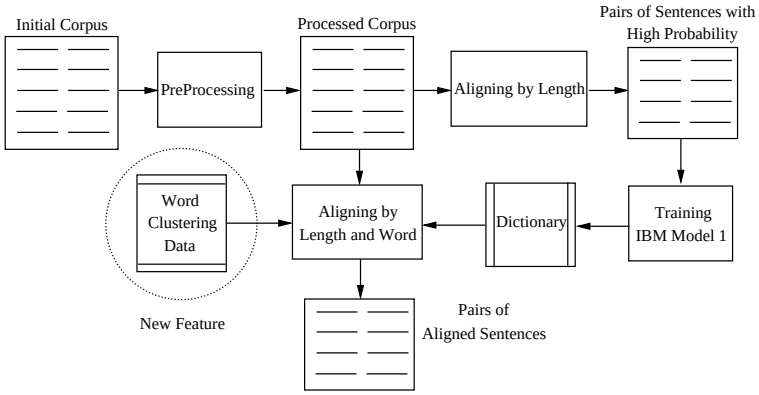

Fig 1. Framework of our sentence alignment algorithm.

B. The Effect of Dictionary

Bilingual word dictionary is usually used in aligners based on combination length-based and word correspondences. Varga, et al. [13] uses an extra dictionary in their framework or train IBM Model 1 to make a dictionary when without such a dictionary. Moore [10] uses IBM Model 1 to make a bilingual dictionary. Suppose that there is a pair of sentences (*s, t*) where *s* is one sentence in the text of source language, *t* is one sentence in the text of target language:

s = (s_1, s_2, ..., s_l), where s_i is a word of sentence *s*.

t = (t_1, t_2, ..., t_m), where t_j is a word of sentence *t*.

To estimate alignment probability of this sentence pair, there has to be a look up each of the word pairs (s_i, t_j) in the dictionary. However, a dictionary do not contain all word pairs, and this is increasingly evident when processing sparse data. Moore [10] deals with this issue by assigning every word not found in the dictionary to a common word "*other*". In the Moore's mehthod, word translation is applied to evaluate alignment probability as formula below:

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

Where:

m is the length of sentence t of target language;

l is the length of sentence s of source language;

$tr(t_j|s_i)$ indicates word translation probability of the word pair (t_j, s_i) looked up from bilingual dictionary.

According to Moore's method, when each word s_i or t_j is not included in dictionary, it is all replaced by an unique word "*other*"; In other word, the word pair (t_j, s_i) is replaced by one of these 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 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

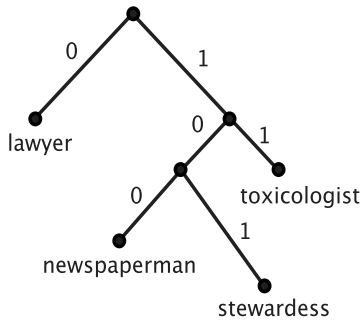

Fig 2. An example of Brown's cluster algorithm

error of this estimation $\varepsilon_i \in \{\epsilon_1, \epsilon_2, \epsilon_3\}$ effects to the correct estimation by a total error ω:

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

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

$$\omega = (k * \mu)^m; \tag{4}$$

When dealing with sparse data, the range of vocabulary lacks considerably to create a reliable dictionary. Therefore, there are more word pairs are not included in dictionary. The bigger the number of lacking word pairs *k*, the bigger the total error ω. This is not really smooth and could lead to declining the quality of sentence aligner. We have addressed this issue by using word clustering introduced in the next sections.

C. 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 the aims of word clustering is the problem of predicting a word from previous words in a sample of text. This algorithm counts the similarity of a word based on its relations with words on its left and words on its right. The input to the algorithm is a corpus of unlabeled data which consist 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 pair 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. The 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 subordinate words, each subordinate word has the same bit string and corresponding frequency.

D. Algorithm

When aligning sentences based on a dictionary, word pairs forming corresponding sentences are looked up in the bilingual word dictionary. All of them, however, do not always appear in the dictionary especially when processing with sparse data. We propose using word clustering data sets to supplement lexical information for bilingual dictionary and improve alignment quality. It is the fact that words in the 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 by ones in the same cluster rather than assigning all of those words to a common word as the method of Moore [10]. We use two word clustering data sets corresponding to the two languages in the corpus. This idea is indicated at the Algorithm. 1.

Algorithm. 1: Sentence Alignment Using Word Clustering

Input: A pair of words (*e*, *v*);
Dictionary *D*, two clusters C_e and C_v
Output: *P*, word translation probability of (*e*, *v*)

if (*e*, *v*) contained in D **then**
  P ⟵ *Pr*(*e*, *v*)
else
  if (*e* contained in *D*) and (*v* contained in *D*) **then**
    looking for all (e_1, e_2, ..., e_n) in C_e
    looking for all (v_1, v_2, ..., v_m) in C_v
    P ⟵ **avg**(*Pr*(e_i,*v*), *Pr*(*e*,v_j)),$1 \leq i \leq n$, $1 \leq j \leq m$
  else
    if (*e* contained in *D*) or (*v* contained in *D*) **then**
      if (*e* contained in *D*) **then**
        looking for all (v_1, v_2, ..., v_m) in C_v
        if (*e*, v_j) contained in *D*) **then**
          P ⟵ **avg**(*Pr*(*e*, v_j)), $1 \leq j \leq m$
        else
          P ⟵ **avg**(*Pr*(*e*) , *(other)*)
      else
        looking for all (e_1, e_2, ..., e_n) in C_e
        if (e_i, *v*) contained in *D*) **then**
          P ⟵ **avg**(*Pr*(e_i, *v*)), $1 \leq i \leq n$
        else
          P ⟵ **avg**(*Pr*((*other)*, *v*))
    else
      P ⟵ *Pr*("*other*", "*other*")
return *P*

In this algorithm, *D* is the dictionary which is created by training IBM Model 1. The dictionary *D* contains word pairs (*e*, *v*), where *e* is the word of the text of source language and *v* is the word of the text of target language, and *Pr*(*e*, *v*) is their word translation probability.

In addition, two data sets clustered by word are used in this algorithm, which contain words of the texts of source language and target language respectively. Words in these two data sets have been divided into clusters in which C_e is a cluster containing word *e*, and C_v is a cluster containing word *v*. When (*e*, *v*) is not contained in the dictionary, each word of

this pair is replaced by all words in its cluster before looking up these new word pairs in the dictionary. The probability of (*e*, *v*) is counted by the function **avg** calculating the average value of probabilities of all word pairs looked up according to the approach in the above-mentioned algorithm. Consider an English-Vietnamese sentence pair as indicated in Table I.

Table I
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 .

Several word pairs in the Dictionary created by training IBM Model 1 can be listed as in Table II:

Table II
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
natural	tự_nhiên	0.44
...		

There are a number of word pairs which are not contained in the Dictionary such as the word pair (*act*, *chức_năng*). Thus, in the first step, algorithm searches cluster of each word in this word pair. Two clusters containing words "*act*" and "*chức_năng*" are shown in Table III and Table IV.

Table III
CLUSTER OF *act*

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

Table IV
CLUSTER OF *chức_năng*

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

In these clusters, the bit strings "0110001111" and "11111110" indicate the identification of the clusters. Word pairs of these two clusters are then looked up in the Dictionary as indicated in Table V.

The next step of the algorithm is to calculate the average value of these probabilities, and the probability of word pair (*act*, *chức_năng*) would be shown in Table VI.

This word pair along with its probability just calculated can be used as a new item in the Dictionary.

Table V
WORD PAIRS ARE LOOKED UP IN DICTIONARY

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

Table VI
PROBABILITY OF THE WORD PAIR (*act*, *chức_năng*)

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

IV. EXPERIMENTS

In this section, we assess the performance of our sentence aligner and conduct experiments to compare with the baseline method (M-Align). First of all, we describe data sets used in experiments. The metrics to evaluate the performance of sentence aligners then are introduced. Finally, we illustrate results of the experiments and give evaluations about performance of the methods.

A. Data

1) Bilingual Corpora

We conduct experiments on 66 pairs of bilingual files English-Vietnamese extracted from websites of World Bank, Science, WHO, and Vietnamtourism, which consist of 1800 English sentences with 39526 words (6333 different words) and 1828 Vietnamese sentences with 40491 words (5721 different words) as shown in Table VII. We align this corpus at the sentence level manually and gain 846 sentences pairs. Moreover, to achieve more reliable result in experiments, we use over 100,000 English-Vietnamese sentence pairs with 1743040 English words (36149 different words) and 1681915 Vietnamese words (25523 different words). These corpora are available at website[1], which consists of 80,000 sentence pairs in Economics-Social topics and 20,000 sentence pairs in information technology topic.

Table VII
BILINGUAL CORPORA

	Number of Sentences English/Vietnamese	**Vocabulary size English/Vietnamese**
Training Data	100038/100038	36149/25523
Test Data	1800/1828	6333/5721

In pre-processing data, we identify the discrimination between lowercase and upper case to make experiments more accurate. This is reasonable because whether a word is lower case or upper case, it is basically similar in the meaning to the other. Thus, we convert all words in these corpora into an unique form, lowercase. In addition to this, in Vietnamese, there are many compound words which may affect to the quality of alignment. As a result, compound words have to be recognized rather than keeping all of them by single words. Therefore, we conduct tokenizing the Vietnamese corpus by using an effective tool for this task available at website[1].

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

2) Word Clustering Data.

Related to applying word clustering feature in our approach, we use two word clustering data sets of English and Vietnamese in experiments which are indicated in Table VIII. We create those data sets by using Brown's word clustering algorithm conducting on two input data sets. The input English data set is extracted from British National Corpus (BNC) with 1044285 sentences (approximately 22 million words). The input Vietnamese data set, meanwhile, is the Viettreebank data set consisting of 700,000 sentences with somewhere in the vicinity of 15 million words including Political-Social topics from 70,000 sentences of Vietnamese treebank[1] and the rest from topics of websites laodong, tuoitre, and PC world.

Table VIII
INPUT CORPORA FOR TRAINING WORD CLUSTERING

	Number of Input Sentences	Vocabulary size
English	1044285	223841
Vietnamese	700000	180099

These corpora is used to create bilingual word clustering data sets by applying word cluster algorithm of Brown, et al. [3] with the number of clusters set by 700 clusters in both data sets. We get two word clustering data sets in which word items of these data sets cover those of these input corpora with a relatively range by 82.96% of English corpus and 81.09% of Vietnamese corpus, indicated in Table IX. These word clustering data sets also cover a large part of vocabulary of Dictionary by 90.31% English vocabulary and 91.82% Vietnamese vocabulary.

Table IX
WORD CLUSTERING DATA SETS.

	Number of Clusters	Dictionary Coverage Scale	Corpus Coverage Scale
English	700	90.31%	82.96%
Vietnamese	700	91.82%	81.09%

B. Metrics

Metrics used to evaluate aligners in our experiments are common ones: *Precision*, *Recall* and *F-measure*.

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

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

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

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.

C. Evaluations

We conduct experiments and compare our approach implemented in Java with the baseline algorithm: M-Align (Bilingual Sentence Aligner, Moore [10]). As mentioned in the previous sections, we apply the hybrid sentence alignment framework along with using word clustering feature. This hybrid method based on the combination of length-based and word translation. A bilingual dictionary is created by training sentence pairs extracted from the initial alignment through a length-based phase. 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 dictionary covers reliable items, the better the alignment result is. A parameter that may influence to this range is the threshold to choose sentence pairs in the initial alignment. A lower threshold gets more sentence pairs which may supplement more word pairs that still ensure reliability and accuracy into dictionary. Therefore, we conduct experiments and set thresholds on a range from 0.5 to 0.99 in the initial alignment in order to not only evaluate approaches fairly but also examine effects of dictionary to alignment result. The threshold of the final alignment is set by 0.9 to ensure a highly reliable result.

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 rate, Fig. 3. 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 correspond with the raise of threshold in the initial alignment. These approaches get the highest precision, 62.55% of our approach and 72.46% of M-Align, when the threshold is set as 0.9. The precision rate of Moore's method is higher than that of our approach in any situation; however, EVS is lower than M-Align with an inconsiderable range. The high precision rate of Moore's method means that its result gets a highly accurate scale regardless of the number of correct sentence pairs, which relates to another important metric, recall rate that is mentioned in the next evaluation.

The corpus we use in experiments is extracted from a number of bilingual websites English-Vietnamese, which data are relatively sparse. Moore's method encounters an ineffective performance especially in term of recall, Fig. 4. Results from Moore's method show a quite low recall rate. In the threshold of 0.5, the recall of M-Align is 51.77%, and it gradually reduces when the threshold is set higher, which it gets 43.74% in the threshold of 0.99.

In our method, instead of replacing probability of the word pair (t_j, s_i) by one of probabilities of the word pairs $(t_j, "other"), ("other", s_i), ("other", "other")$ as Moore's method, we do not use an unique word *"other"* but refer the word t_j or s_i if it is not included in dictionary by words in its cluster. By using the proposed strategy, we not only exploit some characteristics of word clustering for sentence alignment but expect to reduce estimated error of the technique that Moore proposed. Using word clustering data sets whose

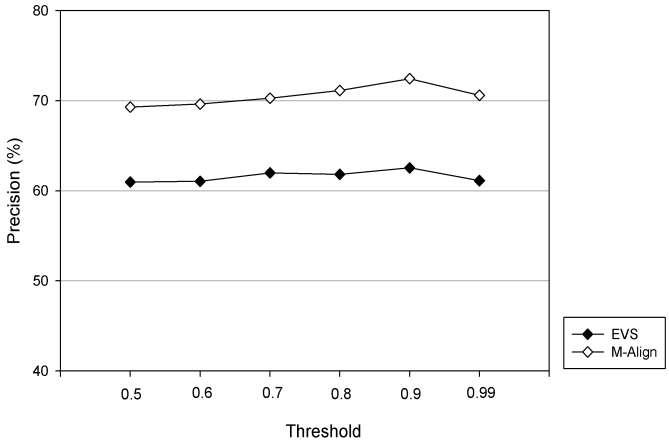

Fig 3. Comparision in Precision (EVS: our approach; M-Align: (Bilingual Sentence Aligner, Moore [10]))

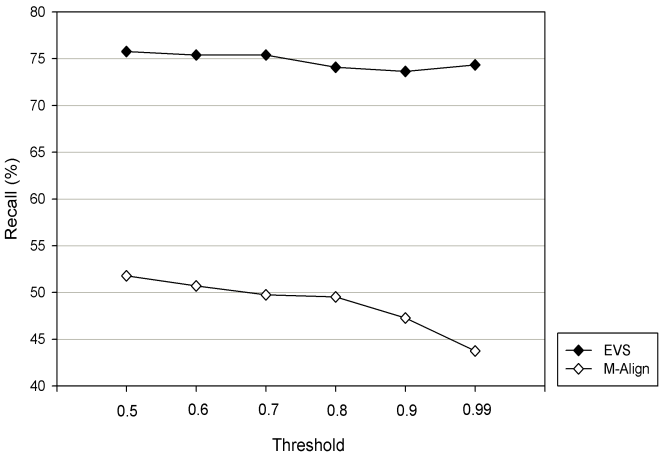

Fig 4. Comparision in Recall (EVS: our approach; M-Align: (Bilingual Sentence Aligner, Moore [10]))

vocabulary covers more than 80% of input corpora and more than 90% of dictionary, the estimated error of our strategy considerably reduces that made by Moore's method.

We verify this strategy by experiments, and one of results is presented in Fig. 4 by evaluating recall rate. Our approach gets the recall significantly higher than that of M-Align, even 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 in processing the lack of lexical information. Meanwhile, when the threshold is increased, the recall of M-Align drops rather considerably. We perform the experiment by decreasing the threshold of the length-based from 0.99 to 0.5 of the initial alignment to evaluate the impact of a dictionary to the quality of alignment. It is an indisputable fact that when using a lower threshold, the number of word items in the dictionary will increase that lead to a growth of recall rate. M-Align usually gets a high precision rate; however, the weakness of this method is the quite low recall ratio, particularly when facing a sparseness of data. This kind of data results in a low accuracy of the dictionary, which is the key factor of a poor recall rate in the approach of Moore [10] because of using only word translation model - IBM Model 1. Our approach, meanwhile, deals with this issue flexibly. If the quality of the dictionary is good enough, a reference to IBM Model 1 also gains a rather accurate output. Moreover, using word clustering data sets which assist to give more translation word pairs by mapping them through their clusters resolved sparse data problem rather thoroughly.

The identification of words not found in the dictionary into a common word as in Moore [10] results in quite a low accuracy in lexical phase that many sentence pairs, therefore, are not found by the aligner. Instead of that, using word clustering feature assists to improve the quality of lexical phase, and thus the performance increases significantly.

Consider the English-Vietnamese sentence pair below as shown in Table X. In experiment, this sentence pair is an alignment in our result, but Moore's method does not explore it.

Table X
AN ENGLISH-VIETNAMESE SENTENCE PAIR

English Sentence	Vietnamese 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** .	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 .

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 XI.

Table XI 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 Moore's method. There are some notations in Table XI:

- (): 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, *(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)*)

Table XI
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

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 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 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 XII, XIII, XIV.

Table XII
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 XIII
P(*well*, *ngựa_ô*)

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

Because our approach significantly improves the recall rate compared with 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. 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 rate (13.08%) when F-measure are 67.09% and 54.01% of EVS and M-Align respectively.

V. CONCLUSIONS AND FUTURE WORKS

The quality of the dictionary significantly impacts the performance of sentence aligners which are based on lexical information. When aligning corpus with sparse data, the dictionary, which is created by training sentence pairs extracted from length-based phase, would lack a great number of word pairs. This leads to a low quality of the dictionary with which declines the performance of the aligners. We have dealt with this issue by using a new feature which is

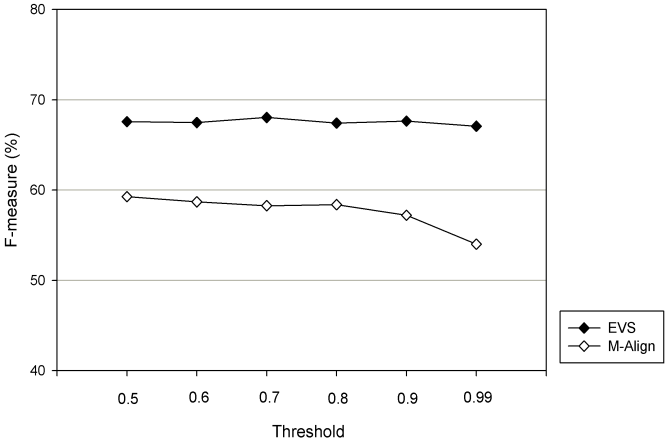

Fig 5. Comparision in F-measure (EVS: our approach; M-Align: (Bilingual Sentence Aligner, Moore [10]))

Table XIV
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

word clustering in our algorithm. The lack of many necessary items in the dictionary is effectively handled by referring to clusters in word clustering data sets sensibly. We associate this feature with the sentence alignment framework proposed by Moore [10] in our method. The experiments indicated that our approach produces a better performance than Moore's method. Word clustering is a useful application and could be utilized in sentence alignment to improve performance of aligners.

In the near future, we try not only to improve the quality of sentence alignment by assessing the correlation of sentence pairs based on word phrases or using better word translation model like IBM Model 4 but to tackle the difficult issues like handling the noisy data.

ACKNOWLEDGMENT

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)

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

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

[9] 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)

[10] 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)

[11] 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)

[12] 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)

[13] 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)

[14] 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)

PLACE PHOTO HERE

Hai-Long Trieu Biography text here.

Phuong-Thai Nguyen Biography text here.