Fundamenta Informaticae 130 (2014) 179–199
DOI 10.3233/FI-2014-987
IOS Press

An Efficient Framework for Extracting Parallel Sentences from Non-Parallel Corpora

Cuong Hoang[*], **Anh-Cuong Le**, **Phuong-Thai Nguyen**, **Son Bao Pham**
University of Engineering and Technology
Vietnam National University, Hanoi, Vietnam
cuongh.mi10@vnu.edu.vn

Tu Bao Ho
Japan Advanced Institute of Science and Technology, Japan and
John von Neumann Institute
Vietnam National University at Ho Chi Minh City, Vietnam

Abstract. Automatically building a large bilingual corpus that contains millions of words is always a challenging task. In particular in case of low-resource languages, it is difficult to find an existing parallel corpus which is large enough for building a real statistical machine translation. However, comparable non-parallel corpora are richly available in the Internet environment, such as in Wikipedia, and from which we can extract valuable parallel texts. This work presents a framework for effectively extracting parallel sentences from that resource, which results in significantly improving the performance of statistical machine translation systems. Our framework is a bootstrapping-based method that is strengthened by using a new measurement for estimating the similarity between two bilingual sentences. We conduct experiment for the language pair of English and Vietnamese and obtain promising results on both constructing parallel corpora and improving the accuracy of machine translation from English to Vietnamese.

Keywords: Parallel sentence extraction; non-parallel comparable corpora; statistical machine translation.

[*] Address for correspondence: University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

1. Introduction

Statistical Machine Translation (SMT) is currently the most successful approach to large vocabulary text translation. All SMT systems share the basic underlying principle of applying a translation model to capture the lexical translations and taking a language model to quantify fluency of the target sentence. SMT is a data-driven approach in which the parameters of the translation models are estimated by iterative maximum-likelihood training based on a large parallel corpus of natural language texts. Hence, the quality of an SMT system is heavily dependent on the "quantity, quality, and domain" of the bilingual training data which contains a large set of parallel sentences, known variously as parallel text, bitext, or multitext [24]. Therefore, constructing a training corpus which contains a set of parallel sentences becomes one of the most important tasks in building any SMT system.

There are mainly two kinds of resources which can be used to construct the training data. The first resource is the collections of parallel texts, such as the Verbmobil Task, Canadian or European Parliament, which are quite large for some languages. However, the parallel corpora for some other language pairs are extremely scarce. It is found from those languages that non-parallel but comparable corpora are much more available from various resources in different domains, such as from Wikipedia, news websites, etc. In fact, these corpora contain some parallel sentences which are the target for extracting in our work.

For the parallel corpora construction, it is recognized from previous studies that automatically obtaining pairs of aligned sentences is simple [8, 10, 16]. In contrast, the task of extracting bitext from comparable non-parallel corpora is completely not a trivial task. Up to the "noise", there may be only a few parallel sentences per pair of candidate documents and it is quite hard to obtain a high recall while keeping the soundness or precision of each clarifying decision. In our opinion, building an efficient similarity measurement is the core for every framework of extracting parallel sentences (from now, this framework is called in short the extracting framework). Moreover, a better measurement gives us not only good recall and precision for each extracting iteration but also an efficient way to apply the bootstrapping scheme to exploit more data.

Previously, some researches deployed a log-linear classifier to clarify whether candidates are parallel or not [25, 29]. The log-linear model is built from many features which are estimated by the IBM alignment Models. Basically, each of the features, such as the longest contiguous connected words, the number of aligned/un-aligned words, the largest fertilities, or the e longest unconnected substring, etc., is used as a filtering condition to remove non-parallel pairs. Deploying many features helps us gain a good precision for the clarifying decision. However, using lots of filtering conditions also unintentionally reduces many other parallel pairs since many pairs do not satisfy one or more of those requirements.

Later, with the availability of some high quality phrase-based SMT frameworks such as MOSES [21], the research in the field has been moving to focus on building the measurement based on the N-grams matching paradigm [1, 2, 3, 12, 13]. That trend offers many advantages including the simplicity, recall or computational performance when comparing to the classification approach. In this work, for measuring the similarity between two bilingual sentences we will use a phrase-based SMT system to translate source sentences to the target languages. We then measure the similarity between the translation sentence and the target sentence using some N-grams evaluation metrics such as BLEU [26], NIST [14] or especially TER [30].

However, the traditional N-grams matching scheme basically focuses on how many N-grams are matched, but do not pay enough attention on validating these N-grams. When dealing with a "noise" non-parallel corpus, according to its complexity, recognizing and removing the "noise" N-grams is the vital point that decides the precision or recall of the detecting method. As the result the traditional N-grams scheme does not provide us a good "noise" filtering capacity which we especially need for our task.

Research on building a better N-grams similarity measurement forces to boost the performance of the extracting framework and this work will focus on that problem. We propose an improved high quality N-grams similarity measurement which is calculated based on the *phrasal* matching with significant improvements. This will help us to determine what N-grams matching could be recognized and what could be not. Hence, our similarity measurement shows a superior in quality when comparing it to other traditional N-grams methods. Based on that similarity metric, the detecting component will classify which candidate pair is parallel. We also integrate the bootstrapping scheme into the extraction framework to extend the training data and hence improve the quality of the SMT system consequently.

To present the performance of the proposed framework, we focus on the Vietnamese language, which is quite scarce lacking of parallel corpora. We choose Wikipedia as the resource for our comparable non-parallel corpora extracting task. It is a big challenge to build a framework which automatically extracts parallel sentences from this resource. In addition, it is the much more difficult task for the low-resource languages since the "noise" on those non-corpora resources are very complex. We present the efficiency of the framework on two aspects. First, from an initial small training corpus, which is not of very high quality, how can we extract as many as possible parallel texts from non-parallel corpora with high quality. Second, how can we expand repeatedly the training corpus by extracting and using previous results. Together, we especially focus on the new domain knowledge exploring capacity from our learning framework.

Various experiments are conducted to testify the performance of our system. We will show that our similarity measurement significantly gives a better performance than TER or other N-grams methods. As the result, the system could extract a large number of parallel sentences with significantly higher quality in the recall. Thanks to the quality of the detecting component, the integration of the bootstrapping scheme into the framework helps us obtain more than 5 millions of words bitext data. In addition, the quality of SMT upgrades gradually together with its "boosting" ability of translation, special for incrementally and automatically covering new domains of knowledge.

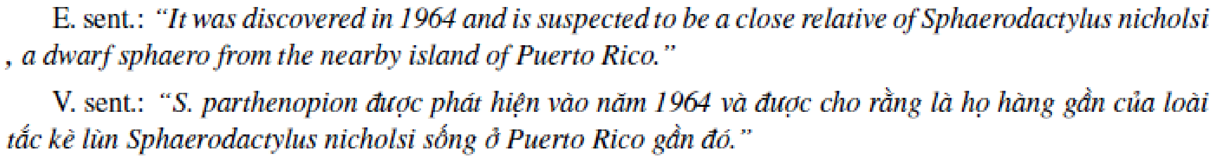
2. The English-Vietnamese Wikipedia System

Wikipedia resource is the typical non-parallel corpora in which we could easy detect the equivalent articles via the *interwiki* link system. There are more than $100,000$ articles which are written in Vietnamese from the Wikipedia System[1]. This number is really small in comparison with the number of English articles. In addition, these Vietnamese documents are quite shorter than the corresponding English articles. The content in the Vietnamese Wikipedia sites is usually partially translated from the corresponding English sites. It is a challenge to build a framework which automatically extracts parallel sentences from here since the "noise" on that non-corpora resource is very complex. We take some examples for the English and Vietnamese Wikipedia systems[2].

[1] This statistics used from the official Wikipedia Statistics: http://meta.wikimedia.org/wiki/List_of_Wikipedias.

[2] These examples are the texts from the corresponding articles: http://en.wikipedia.org/wiki/Virgin_Islands_dwarf_sphaero and http://vi.wikipedia.org/wiki/T?c_ke_lun_qu?n_đ?o_Virgin.

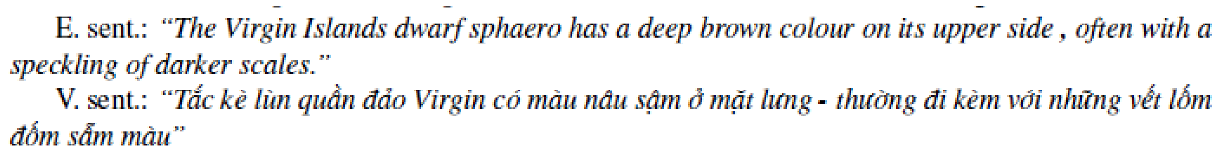
Basically, the equivalent sentences are often written in a different cognate structures of different languages. Therefore it is not safe to use only cognate structures as previous studies, such as [28] to filter “noise” data. The bellow example is a pair of equivalent sentences which are the same meaning but the Vietnamese sentence misses the comma translating cognate.



E. sent.: *“It was discovered in 1964 and is suspected to be a close relative of Sphaerodactylus nicholsi , a dwarf sphaero from the nearby island of Puerto Rico.”*

V. sent.: *“S. parthenopion được phát hiện vào năm 1964 và được cho rằng là họ hàng gần của loài tắc kè lùn Sphaerodactylus nicholsi sống ở Puerto Rico gần đó.”*

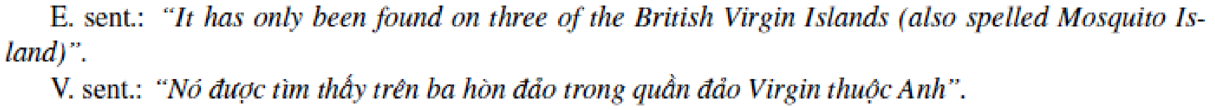
We take another example, two equivalent sentences have the same meaning but have different cognate structures. The bellow pair is an example in which these sentences are different in cognates (**“-"** and **“,"**).



E. sent.: *“The Virgin Islands dwarf sphaero has a deep brown colour on its upper side , often with a speckling of darker scales.”*

V. sent.: *“Tắc kè lùn quần đảo Virgin có màu nâu sậm ở mặt lưng - thường đi kèm với những vết lốm đốm sẫm màu”*

Last but not least, the Vietnamese sentence is usually partial translated from the original English sentence. Similarly, the bellow example is a pair of the partial equivalent sentences. That is, the partial “*(also spelled Mosquito Island)*” translation is missed in the Vietnamese sentence.



E. sent.: *“It has only been found on three of the British Virgin Islands (also spelled Mosquito Island)”.*

V. sent.: *“Nó được tìm thấy trên ba hòn đảo trong quần đảo Virgin thuộc Anh”.*

The examples above just illustrate some of many “noise” phenomena which deeply reduce the performance of any parallel sentence extraction system when encountering with our task. In addition, they are not only the problems for extracting from English-Vietnamese Wikipedia but also for all of the languages. In practice, if we ignore these problems and try to use the cognate condition, the recall is extremely low. Previously, the research on that topic usually just focuses on extracting corresponding words from Wikipedia [5, 15]. Some others propose some basic extracting methods which result a small number of extracted parallel words [31].

To overcome those problems, for the first one we add to the list of sentences another one which purely does not have any comma and other symbols[3]. For the second phenomenon, we split all sentences to some other parts based on the cognates and add them to the list of sentences, too[4]. Deploying these improved schemes creates more candidates because by which the traditional strong cognate condition does not filter much “noise” pairs. We take an example about our statistics from a small set which contains $10,000$ pairs of article links. We have processed $58,313,743$ candidates and obtained around $53,998$ pairs ($\approx$ 1080 candidates/1 bitext pair). We extremely need a significantly better N-grams which is integrated to the extracting framework. In the section Experiment, we will point out that our framework does it well.

[3] For the first case, we add other sentences: *“It was discovered in 1964 and is suspected to be a close relative of Sphaerodactylus nicholsi a dwarf sphaero from the nearby island of Puerto Rico.”* and *“The Virgin Islands dwarf sphaero has a deep brown colour on its upper side often with a speckling of darker scales.”*

[4] For example, for the last pair we add more sentences: *“It has only been found on three of the British Virgin Islands."* and *“also spelled Mosquito Island."*

3. The Extracting Framework Description

The general architecture of a parallel sentence extraction system is very common. At first, the extracting system selects pairs of similar documents. From each of such document pairs, it generates all possible sentence pairs and passes them through a *similarity measurement*. Note that, the step of finding similarity document emphasizes recall rather than precision. It does not attempt to find the best-matching candidate documents. It rather prefers to find a set of similar parallel documents. The key point here, like the core of any extraction system, is how to build a *detecting component* to classify the set of candidate sentences and decide the degree of parallelism between bilingual sentences.

Figure 1 shows the architecture of our extracting framework that deals with two tasks: extracting parallel texts from candidates and improving the corresponding SMT system by applying the bootstrapping scheme.

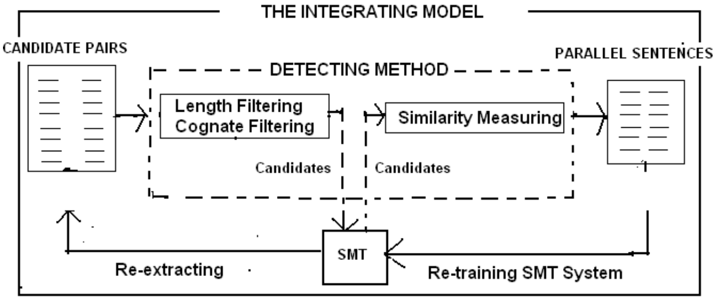


Figure 1. Architecture of the proposed model.

3.1. The General Framework

The general architecture of our parallel sentence extraction system could be described as follows: Starting with the candidates of comparable documents extracted from corpora, we generate a set $\mathcal{C}$ of all possible candidates of sentence pairs $(f^{(c)}, e^{(c)}) : c = 1, \ldots, \mathcal{C}$. Hence, these pairs pass through the detecting component for clarifying the parallel texts.

The detecting component consists of two consecutive sub-components. The first one aims to filter the candidates based on the conditions of length ratio between the source and target sentences. In addition, the candidates is also filtered by the cognate condition, but do not base much on closed similarity as mentioned above. If the pair c passes those above conditions, its source sentence will be translated by the SMT system and the obtained translation sentence will be compared to the target sentence. The similarity measurement component, as the last one, tries to estimate the *similarity* between the translation sentence and the target sentence, and assign it as the similarity to the pair c.

3.2. The Parallel Text Detecting Component

As the core of our extracting framework, the detecting component includes two checking steps as follows:

- *Step 1* - Filtering candidates based on the ratio of lengths and cognate structure of the sentences in each candidate.

- *Step 2* - Measuring the similarity between candidates based on the following algorithm. Consequently, it will determine whether a candidate is parallel or not.

The Similarity Measuring Algorithm.

```
Input: Candidate $\mathcal{C}(f_s, e_s)$
Return: The similarity: $sim_{overlap,phrase}(\mathcal{C}(f_s, e_s))$
1. Sentence $t_s$ = decoding(SMT System, $e_s$)
2. Return the similarity:
    $sim_{overlap,phrase}(\mathcal{C}(f_s, t_s)) = tanh(\frac{overlap_{phrase}(t,f))}{|t|+|f|})$
```

As one of the first studies for sentence alignment [16] has used the rate of lengths between two bilingual sentences for measuring the similarity between them. That is, the length measurement could be "used in a dynamic programming framework in order to find the maximum likelihood alignment of sentences". This method is successful in a very "clean" environment (purely parallel corpora). There is a remarkable point in which we further concern with the length condition for reducing as much as possible the computational cost. However, it is worth to recognize that SMT does not represent the length condition as a major factor in its translation models. Consider the original equation for IBM Models 3-5 (it is similar to IBM Models 1-2) [9], which describes the "joint likelihood" for a tableau, τ, and a permutation, π,:

$$\begin{aligned} Pr(f,a|e) = Pr(\tau,\pi|e) \;\; = \;\; & \prod_{i=1}^{I} Pr(\phi_i|\phi_1^{i-1},e) \times Pr(\phi_0|\phi_1^I,e) \times \\ & \prod_{i=0}^{I}\prod_{k=1}^{\phi_i} Pr(\tau_{ik}|\tau_{i_1^{k-1}},\tau_0^{i-1},\phi_0^I,e) \times \\ & \prod_{i=1}^{I}\prod_{k=1}^{\phi_i} Pr(\pi_{ik}|\pi_{i_1^{k-1}},\pi_1^{i-1},\tau_0^I,\phi_0^I,e) \times \\ & \prod_{k=1}^{\phi_0} Pr(\pi_{0k}|\pi_{0_1^{k-1}},\pi_1^I,\tau_0^I,\phi_0^I,e) \qquad (1) \end{aligned}$$

There is no factor represented the length translation probability from equation (1). In other words, there is no requirement (or constraint) about the relationship between the lengths of parallel sentences. The condition as [16] pointed out is not revealed or exhibited explicitly by IBM Models and consecutively to the statistical phrase-based translation. It means that by using only an SMT system (without deploying the length condition constraint) to extract parallel texts, we lack one of the most interesting features from the bilingual text data. Therefore, by integrating the length condition, our framework is hoped to improve the result of extracting parallel sentences.

Finally, in the third step, the precision is paramount. We estimate a score based on our similarity measurement between two sentences in a candidate. If this score is greater or equal a threshold called λ, we will obtain a new pair of parallel sentences. More detail about this step will be described in Section 4.

3.3. The Bootstrapping Component

The prior target of parallel sentence extraction at each specific time t (corresponding to the system's translation ability at that time $\mathcal{C}_t$) could not extract all the parallel sentence pairs from a comparable non-parallel corpora resource. Alternatively, the highest priority task at each specific time is extracting all possible candidates based on the system's translation ability at that time. After that, we will append all the new parallel sentence pairs to the SMT system's training set and re-train the SMT system. Hence, we have a better translation system to re-extract the resource again.

4. The Similarity Measuring Algorithm

To determine the similarity score of two bilingual sentences (in the source language and the target language), we firstly use a complete phrase-based SMT system to translate the source sentence and obtain its translation in the target language. Second, by applying a *phrasal overlap* similarity measurement to estimate the similarity between the translation sentence and the target sentence (both are now in the same language), we try to clarify that pair as the parallel texts or not.

4.1. The Phrase-based SMT System

As the result of deploying a phrase-based SMT system, we can utilize more accurate information from the translation outputs. Our phrase-based SMT system is based on the MOSES framework [21]. The basic formula for finding e_{best} (decoding step) in a statistical phrase-based model (mixing several components which contributes to the overall score: the phrase translation probability ϕ, reordering model d and the language model p_{LM}), which gives all $i = 1, \ldots, I$ input phrases f_i and output phrase e_i and their positions $start_i$ and end_i:

$$e_{best} = argmax_e \prod_{i=1}^{I} \phi(\overline{f_i}|\overline{e_i}) d(start_i - end_{i-1} - 1) p_{LM}(e) \tag{2}$$

In the corpora of Wikipedia, almost sentences are long with about 20-25 words. However, we usually limit the length of phrase extracted to only some words because it gains top performance, as mentioned in [22]. Using longer phrases does not yield much improvement, and occasionally leads to worse results. In addition, the tri-gram model is shown as the most successful language model [11, 23]. It is also usually used as the default language model's parameter for a phrase-based SMT system [22].

For training a SMT system, if we don't have a very large training set, we could set a large value for n in n-grams. However, this can encounter with the over-fitting problem. As n increases, the accuracy of the n-grams model increases, but the reliability of our parameter estimates decreases, drawn as they must be from a limited training set [7]. We see that P_{LM} is actually *not clue enough* to be existed a relationship between all phrases e_i. Thus we can assume that in the output of the decoding step of a complete phrase-based SMT system, each phrase element e_i is *independent with other elements*, or there is *no or rarely* relationship between these elements.

4.2. The Similarity Measurement

Normally, a baseline measurement for computing the similarity between sentence t and sentence e is the proportion of words in t that also appears in e, which is formulated in the following:

$$sim(t,e) = \frac{2 \times |t \cap e|}{|t| + |e|} \tag{3}$$

where $|t \cap e|$ is the number of words/terms appeared in both sentences t and e.

In addition, there is a Zipfian relationship between the lengths of phrases and their frequencies in a large corpus of text: the longer the phrase, the less its frequency. Therefore, to emphasize the long phrase overlap in the similarity measurement some studies assigned an n-word overlap the score of n^2 (or in some similar ways). In fact, [6] introduced a "multi-word phrases overlap" measurement based on Zipf's law between the length of phrases and their frequencies in a text collection. Together, [27] used the sum of sentence lengths, and applied the hyperbolic tangent function to minimize the effect of the outliers. More detail, for computing phrasal overlapping measurement between the sentence *t* and *e* (i.e. the content-based similarity), the formulate denoted in [6, 27] is described as follows:

$$sim_{overlap,phrase}(t,e) = tanh(\frac{overlap_{phrase}(t,e)}{|t| + |e|}) \tag{4}$$

where

$$overlap(t,e) = \sum_{n}^{N} \sum_{m} n^2 \tag{5}$$

here m is a number of n-word phrases that appear in both sentences.

We will apply this scheme in our case with some appropriate adaptations based on outputs of the decoding process of the SMT system. From the results of the MOSES's decoding process we can split the translation sentence into separate segments. For example, a translation_sentence_with_trace[5] has format sequence of segments as follows:

$$\hat{t} = |\ldots||w_1 w_2 \ldots w_k||w_{k+1} w_{k+2} \ldots w_n||\ldots|$$

Generally, if we treat these segments independently, we can avoid measuring the overlap on the phrases such as $w_k w_{k+1}$, or $w_{k-1} w_k w_{k+1}$, etc. As we analysed previously, the word w_k and the word w_{k+1} seem not co-occur more often than would be expected by chance. It means we will not take the phrases in which their words appear in different translation segments. Note that in a "noisy" environment this phenomenon may cause many wrong results for sentence alignment.

[5] Running the MOSES decoder with the segmentation trace switch using -t option

4.3. Recognizing N-word Overlapping Phrases

From our observation, long overlapping phrases take a large proportion in the score of overlapping measurement. For example, if a 3-word overlapping phrase is counted, it also contains the two 2-word overlapping sub-phrases, and the three 1-word overlapping sub-phrases. Therefore, the total value $overlap(t, e)$ always obtains:

$$overlap(t, e) \geq 3^2 + 2 \times (2^2) + 3 \times (1^1) = 20$$

Hence, the appearance of overlapping phrases in non-parallel sentences may cause much mis-detection of parallel sentence pairs. In a very "noisy" environment as our task, there easily exists a lot of overlapping phrases which tend to occur randomly. To our knowledge, this phenomenon has not been mentioned in previous studies.

To overcome this drawback, we add a *constraint rule* for recognizing an N-word overlapping phrase with $N \geq 2$. An overlapping phrase with N-words ($N \geq 2$) (called N-word overlapping phrase) will be counted or recognized if and only if there at least exists N different overlapping phrases (or words) with their lengths are shorter than N. Together, these "smaller" phrases must not be the fragments (or sub-parts) of the N-word overlapping phrase.

We take an example as below. There are four translation sentences (from T.1 to T.4) of English from a source sentence. The reference sentence is also given bellow:

English sentence: *"shellshock 2 blood trails* ***is a first-person*** *shooter video game developed by rebellion developments"*

T.1: | ***shellshock*** | | - | | - ***trails*** | | ***is a first-person*** - | | - | | - | | - | | - | | ***developments*** |

T.2: | ***shellshock*** | | - | | ***blood trails*** | | ***is a first-person*** - | | - | | ***game*** | | - | | - | | - |

T.3: | ***shellshock*** | | - | | ***blood trails*** | | ***is a first-person*** - | | - | | - | | ***developed by*** | | - | | - |

T.4: | - | | - | | - | | ***is a first-person shooter***| | - | | -| | -| | -| | -|

We want to determine how the part "**is a first-person**" is recognized as a 3-grams matching. According to our constraint rule, if a 3-word overlapping phrase is counted, there at least, for example, 3 different overlapping words or 2 different 2-word overlapping phrases together with 1 different overlapping word, etc. Hence, the sub-part "**is a first-person**" can be recognized as a 3-grams matching from T1[6], T2[7] or T3[8].

Importantly, for T4, because there does not exist any other overlapping phrase, therefore the phrase "**is a first-person shooter**" is not recognized as an 4-grams matching. Similarly to the two 3-word overlapping phrases "**is a first-person**" or "**a first-person shooter**". However, three 2-word overlapping phrases ("**is a**", "**a first-person**" or "**first-person shooter**") will be recognized as three 2-grams matchings since when we remove them out, there still exists 2 other separated words.

A 2-grams matching consists two 1-gram matchings. Similarly, a 3-grams matching consists two 2-grams and three 1-gram matchings (the total n-grams we have is 5). A 4-grams matching consists two 3-grams, three 2-grams and four 1-gram matchings (the total n-grams is 9). Similarly for the others, we use these rules to implement our constraint rule as the pseudo-code below:

[6] There are three 1-word overlapping phrases (**shellshock**, **trails** and **developments**).

[7] There are four 1-word overlapping phrases (**shellshock**, **game**, **blood**, **trails**) and one 2-word overlapping phrase (**blood trails**).

[8] There are five 1-word overlapping phrase (**shellshock**, **blood**, **trails**, **developed**, **by**) and two 2-word overlapping phrases (**blood trails**, **developed by**).

Algorithm: Recognize N_grams ($TOTAL$, N)

/* $TOTAL$ is the total of all m-grams with m < N. N is the value of N-grams which we check */

```
switch ($N$)
    case 2: $TOTAL$ -= 2;
    case 3: $TOTAL$ -= 5;
    case 4: $TOTAL$ -= 9;
    case 5: $TOTAL$ -= 14;
    case 6: $TOTAL$ -= 20;
    case 7: $TOTAL$ -= 27;
    if ($TOTAL$ >= $N$)
        return true;
    else
        return false;
```

Basically, we recognize n-grams ($n \geq 2$) matchings in an iterative method. Before recognizing an n-grams matching, we enumerate all m-grams (m < n) matchings previously (for the pseudocode, the representing variable is set as $TOTAL$). Hence, we ignore all the m-grams which are counted as the sub-parts of the n-grams matching. If the final result is greater or equal than N we then accept it as a legal n-grams matching.

The combination between the phrase-based overlapping movement and our proposed constraint rule creates an effective influence in both accuracy and performance aspects of the extracting framework, in comparison with lexicon based methods. We will delve into more detail about them in later sections.

5. Data Preparation and Performance Evaluation

In this work, all experiments are deployed on an English-Vietnamese phrase-based SMT project, using MOSES framework [22]. We use an initial bilingual corpus for training our SMT systems, which is constructed from Subtitle resources as credited in [17]. Note that almost all parallel sentences from this data consists of the normal conversations between characters on films and its content is far different from the Wikipedia resource.

To process the English data, we use the Stanford Tokenizer – an efficient, fast, deterministic tokenizer[9]. In addition, for estimating the similarity between two sentences, and we use F-score which is the harmonic mean of precision and recall. The general formula for computing F-score is:

$$F_1 = 2 \cdot \{precision \cdot recall\}/\{precision + recall\} \tag{6}$$

We train the initial corpora and perform the experimental evaluations to account three major contributions of our proposed framework:

[9] Available at: http://nlp.stanford.edu/software/tokenizer.shtml

- The experiment for Parallel Sentence Extraction: We will show the ability with high precision and recall of extracting a large number of parallel sentences from a "noise" comparable non-parallel corpus.
- The experiment for analyzing Similarity Measurement: We will show the effectiveness of the proposed similarity measurement method through some experimental analyses.
- The experiment for Bootstrapping and Statistical Machine Translation: We will show the ability of expanding the training corpus, and improving the SMT system under the bootstrapping scheme.

Because of guaranteeing the best performance of applying the bootstrapping scheme, the vital requirement we need is to ensure the precision of the extracting system. In more detail, our goal is to achieve a precision around 95%. Together, we also have to achieve an enough high recall to obtain more parallel data. This is an important point which previous studies do not concern with. In the experimental sections, we will point out that by deploying our extracting framework, it is feasible to satisfy those requirements.

5.1. Wikipedia Resource Data

In this work, we use Wikipedia resource for extracting parallel sentence pairs of English and Vietnamese. A Wikipedia page (in source language) will connect to (if exists) another Wikipedia page (in target language) via an "interwiki" link in the Wikipedia's hyperlink structure. Based on this information we can collect a large set of bilingual pages in English and Vietnamese. Hence, from a pair of pages denoted as A (containing *m* sentences) and B (containing *n* sentences) in the candidate set, we have $n \times m$ pairs of the parallel sentence candidates (the Cartesian product).

It is also worth to emphasize again that the domain knowledge gaining from the Wikipedia resource is far different from Subtitle domain. The bilingual sentences obtained from Subtitle resources are usually simple, short and contains a lot of abbreviations. However, the bilingual sentences getting from our parallel text detecting method are longer, more complex in structure and extremely far diverse in the content information.

5.2. Artificial Resource Data

We also want to generate some fully difficult test cases to point out the capacity in which our framework could efficiently exploit from some very difficult (i.e. "noise") environments (for example: the World Wide Web). From $10,000$ pairs of parallel sentences we sort them by the alphabet order for Vietnamese sentences and obtain the new set. In this set of parallel sentence pairs, two "neighbor" pairs may be very similar for Vietnamese sentences but quite different for English sentences. Hence, for a pair of parallel sentence, we create other candidates by choosing an English sentence from itself with the Vietnamese sentence from its Vietnamese sentence's "neighbors". These candidate pairs fully have a lot of matching n-grams. Some of them are even very similar in meaning. However, they are actually not the parallel texts. This idea of constructing that noise non-parallel corpora is similar to [25] but the method is much simpler.

We will use about $100,000$ pairs of candidates (they are both satisfied the conditions on length and cognation) for evaluating the similarity measuring algorithm. Our task is to try to extract parallel sentences from this "noise" environment, in which the original N-grams matching method is failed for the task. By deploying this experiment, we will show out the improvement of our detecting component. To check the consistency and reliability of the experimental result, we will use three training sets with different sizes, which are the sets of $10,000$, $50,000$ and $90,000$ bilingual sentences collected from the Subtitle resources. Note that the extracted parallel sentence pairs will be checked manually to obtain the performance result.

6. Results for Parallel Data Extraction

6.1. Wikipedia Parallel Text Extraction

We start with a set of $50,000$ available parallel sentence pairs collected from the Subtitle resource. These experiments come with a set of about $1,124,169$ parallel sentence candidates getting from Wikipedia resource (they are both satisfied the conditions on length and cognation) and to be going through the similarity measuring component. With the obtained parallel sentence pairs, we will manually check them. Table 1 shows the results when using the original N-gram overlapping similarity measurement as described by [4, 6], with our setting N=1. When we try to use a higher N-grams matching method (2-grams, 3-grams, 4-grams[10]), we found that the result is not significantly changed. Some of them are even significantly worse.

Table 2 presents the results when we deploy our similarity measurement, which is the combination of the phrasal overlapping, independent fragments, and the constraint rule.

λ	Total	True	Wrong	Error(%)
0.45	778	**582**	196	**25.19**
0.5	474	389	85	**17.93**
0.55	266	231	35	**13.15**
0.6	156	140	16	**10.25**

Table 1. The data extracting result using N-gram overlapping similarity measurement (N=1).

λ	Total	True	Wrong	Error(%)
0.3	893	**851**	42	**4.70**
0.35	595	**586**	9	**1.51**
0.4	404	401	3	**0.74**
0.45	272	272	0	**0.00**
0.5	172	172	0	**0.00**

Table 2. Result of parallel sentence extraction using our similarity measurement.

Notice that the λ thresholds in the word-based similarity measurement are different from the λ thresholds used in our similarity measurement. The obtained results in Table 1 and Table 2 have depicted our

[10] Basically, the idea of BLEU is the 4-grams matching approach with some adaptations. This is also similar to other evaluation methods.

	10,000				**50,000**				**90,000**			
λ	Total	Recall	Preci.	F-1	Total	Recall	Preci.	F-1	Total	Recall	Preci.	F-1
0.4	1727	12.95	74.99	22.09	3376	27.19	80.54	40.66	6374	52.35	82.13	63.94
0.45	873	7.29	83.51	13.41	1987	17.40	87.57	29.03	4529	39.89	88.08	54.91
0.5	706	5.85	82.86	10.93	1624	14.26	87.81	24.54	3858	34.43	89.24	46.69
0.55	306	2.72	88.89	5.28	810	7.58	93.58	14.02	2356	22.10	93.80	35.77
0.6	171	**1.63**	**95.32**	**3.21**	522	**4.96**	**95.02**	**9.43**	1676	**15.90**	**94.87**	**27.24**

Table 3. Extracting parallel text pairs with three SMT systems using 1-gram overlapping measurement.

	10,000				**50,000**				**90,000**			
λ	Total	Recall	Preci.	F-1	Total	Recall	Preci.	F-1	Total	Recall	Preci.	F-1
0.3	699	**6.64**	**95.00**	**12.41**	1830	**17.46**	**95.41**	**29.52**	3792	35.85	94.54	51.99
0.35	483	4.66	96.48	8.89	1298	12.57	96.84	22.25	3086	**29.49**	**95.56**	**45.07**
0.4	337	3.29	97.63	6.37	982	9.58	97.56	17.45	2621	25.21	96.18	39.95
0.45	217	2.12	97.70	4.15	700	6.87	98.15	12.84	2155	20.89	96.94	34.37
0.5	143	1.41	98.60	2.78	546	5.38	98.53	10.02	1794	17.48	97.44	29.64

Table 4. Extracting parallel text pairs with three SMT systems using our similarity measurement.

similarity measurement significantly feasible for extracting parallel texts from the “noise” non-parallel corpus. The error-rate bearing in the case of using the normal overlapping method is deeply higher in this environment. However, by using our detecting component, we can achieve much lower error-rates.

Especially, the detecting component even did not make any wrong clarifying decision when the λ threshold is greater than 0.4 (for this particular case). It is interesting that, by applying our similarity measurement, we even be able to extract a larger number of the parallel sentence pairs (586 pairs in comparison with 582 pairs) while we still obtain significantly higher accurate result (25.19 vs 1.51). In addition, we could still satisfy our requirement about the precision of around 95% with a large extracted parallel sentences (851 pairs).

6.2. Artificial Data Extraction Result

From the above experiment, we see that the performance of our similarity measurement, which comes from the combination between the recall and error-rate controlling ability, is deeply better in comparison with the traditional N-grams matching approach. Hence, this experimental evaluation comes to confirm the significantly better in quality of our framework’s extracting ability when we deploy the detecting component on a much more difficult environment (i.e. on the artificial data).

Table 3 presents the results conducting on the artificial non-parallel data when we deploy the N-gram overlapping measurement for each threshold level. Similarly, Table 4 presents in detail the extracting results when we use our similarity measurement for exploiting the normal non-parallel corpus. To satisfy the requirement of the precision performance, our detecting component also obtains a significantly better result in comparison with the normal N-grams approach (6.64 vs 1.63, 17.46 vs 4.96 and 29.49 vs 15.90 in recall and 12.41 vs 3.21, 29.52 vs 9.43, 45.07 vs 27.24 in F-score).

7. Experimental Results for Similarity Measurement

This section will use the artificial resource data for providing experimental evidences to confirm our observation to the effectiveness of proposed similarity measurement method.

For each section below, we will show the result in which the appropriate λ value is taken to obtain the precision around 95%.

7.1. Independent Assumption

This evaluation shows in more detail the difference between applying and without applying the independent assumption of translation segments for computing the similarity. What we want to exhibit in our evaluation is the wrong results we have to bear from using the similarity measurement without applying the independent assumption. More results are described in Table 5:

System	Without Deploying			Baseline		
	Preci.	Recall/$\Delta(\%)$	F-1/$\Delta(\%)$	Preci.	Recall	F-1
10,000	94.64	5.65/**-0.99**	10.66/**-1.75**	**95.00**	**6.64**	**12.41**
50,000	95.35	15.17/**-2.29**	26.18/**-3.34**	**95.41**	**17.46**	**29.52**
90,000	95.17	22.87/**-6.62**	36.88/**-8.19**	**95.56**	**29.49**	**45.07**

Table 5. Extracting parallel text pairs with three SMT systems without applying independence assumption property.

From Table 5, we could see the fact that the recall and the F-score are quite abate and the decrease gaps increase larger for a better SMT extracting system (5.65 vs 6.64, 15.17 vs 17.46 and 22.87 vs 29.49 in recall and 10.66 vs 12.41, 26.18 vs 29.52, 36.88 vs 45.07 in F-score).

7.2. Zipf's Law Evaluation

This evaluation presents the difference between the extraction systems with or without applying the Zipf law for computing the similarity. Table 6 describes more detail about the parallel text extraction results by the systems without applying Zipf law.

System	Without Deploying			Baseline		
	Preci.	Recall/$\Delta(\%)$	F-1/$\Delta(\%)$	Preci.	Recall	F-1
10,000	94.61	5.09/**-1.55**	9.66/**-**	**95.00**	**6.64**	**12.41**
50,000	95.69	12.34/**-5.12**	21.86/**-**	**95.41**	**17.46**	**29.52**
90,000	95.14	26.41/**-3.08**	41.34/**-**	**95.56**	**29.49**	**45.07**

Table 6. The difference between applying Zipf Law and without applying Zipf Law assumption.

Similarly to the above experimental evaluation, we could see the similar phenomenon about recall and F-score. That is the performance is quite abate and the decrease gaps increase larger for a better SMT extracting system (5.09 vs 6.64, 12.34 vs 17.46 and 26.41 vs 29.49 in recall and 9.66 vs 12.41, 21.86 vs 29.52, 41.34 vs 45.07 in F-score).

7.3. Alignment Position

For the task of detecting parallel sentences, there is a common problem in many related studies, as pointed out by [1] that we usually face the case in which two sentences share many common words, but actually convey different meaning. Some studies use the alignment position constraint which is a constraint of the position of the overlapping words. For example, if a candidate of parallel sentences is a pair of parallel texts, firstly, it must satisfy the condition in alignment, in which at least two words overlapping have to lie on two halves of the target sentence. Another trigger constraint: at least two words overlapping have to lie on three parts of the target sentence, etc.

In this evaluation, we will demonstrate the influence happened by changing the alignment position, to show the features which are selected by previous methods are not always accurate, especially for some language pairs of very different in linguistics such as English and Vietnamese. Table 7 shows more detail about the condition of alignment positions in which they include 2 parts, 3 parts, or 4 parts and the difference between them and our original method. The result is quite clear, when we add a constraint on the alignment position, it does not improve the extracting system for the language pair of English and Vietnamese.

	10,000 (Original: **6.64)**		50,000 (Original: **17.46)**		90,000 (Original: **29.49)**	
	Recall(%)	Δ(%)	Recall(%)	Δ(%)	Recall(%)	Δ(%)
2-parts	6.53	**-0.11**	17.11	**-0.35**	35.38	**-0.47**
3-parts	5.22	**-1.42**	13.33	**-4.13**	28.3	**-7.55**
4-parts	2.87	**-3.77**	7.53	**-9.93**	16.39	**-19.46**

Table 7. Different results between adding Alignment Position Constraint and without applying Alignment Constraint.

This comes from the fact that English and Vietnamese languages are quite different in word order [20, 19, 18]. Because of that, we cannot create a constraint in which the position from the first sentence has to match with the position in a determined range from the second one. This is the key point which explains to us the fact that applying the alignment position may obtain a worse result.

7.4. Comparison with Translation Edit Rate

Translation **E**dit **R**ate (or **TER**) [30, 1] is a well-known criterion. TER is applied by [1, 2, 3] and it has been proved that deploying TER instead of the traditional N-grams approach such as BLEU, NIST gives us a significantly better performance. In other words, TER is currently state-of-the-art N-grams uses as the similarity measurement which we integrate to the extracting framework.

Basically, TER was defined as the minimum number of edits needed to change a hypothesis so that it exactly matches one of the references, normalized by the average length of the references. Since we are concerned with the minimum number of edits needed to modify the hypothesis, we only measure the number of edits to the closest reference (as measured by the TER score). Put n_e as the number of edits, and n_a as the average number of reference words. Specifically, we have:

$$TER = \frac{n_e}{n_a} \tag{7}$$

Possible edits include the insertion, deletion, and substitution of single words as well as shifts of word sequences. A shift moves a contiguous sequence of words within the hypothesis to another location within the hypothesis. All edits, including shifts of any number of words, by any distance, have equal cost. In addition, punctuation tokens are treated as normal words and mis-capitalization is counted as an edit.

Table 8 presents experimental results of extracting parallel sentence pairs obtained by using TER criterion. As shown in this table, by using TER criterion we usually obtain worse results of recall in comparison with using our similarity measurement (3.93 vs 6.64, 9.90 vs 17.46, 23.00 vs 29.49). This is an important point which certifies the significantly better performance of our extracting framework for applying the bootstrapping scheme in the next section.

System	**TER**			**Baseline**		
	Preci.	Recall/$\Delta(\%)$	F-1/$\Delta(\%)$	Preci.	Recall	F-1
10,000	94.93	3.93/**-2.71**	7.55/**-4.86**	**95.00**	**6.64**	**12.41**
50,000	94.38	9.90/**-7.56**	17.92 /**-11.6**	**95.41**	**17.46**	**29.52**
90,000	95.34	23.00/**-6.49**	37.06/**-8.01**	**95.56**	**29.49**	**45.07**

Table 8. Extracting parallel text pairs using TER similarity measurement.

7.5. Performance Analysis

There are some points we found to reason about the significantly better result of using our similarity measurement. We make a statistics on the result of the decoding process of $924,529$ English sentences. We list all of the segments and count the number of words from them. As the mention above, the translation output of a sentence has the format as follows:

$$\hat{t} = |\ldots||w_1 w_2 \ldots w_k||w_{k+1} w_{k+2} \ldots w_n||\ldots|$$

For example, for the segment $|w_1 w_2 \ldots w_k|$, the number of word is k. Similarly, for the other one $|w_{k+1} w_{k+2} \ldots w_n|$, the number of word is $(n-(k+1)+1)$. We hence count all the length and make a statistics. The total result in overall is presented in Table 9 below:

Segments	Total of segments	Percent($\%$)
1	$10,178,608$	81.19
2	$1,795,099$	14.32
3	$444,090$	3.54
4	$88,053$	0.70
5	$22,060$	0.18
6	$5,772$	0.05
7	$2,690$	0.02
8	0	0.00

Table 9. Statistics of the Length of Segments.

From Table 9, we analysize some points which explain for the better obtained results. Firstly, an overlapping 1-gram or 2-grams, which are major portions in overall (81.19% and 14.32%), does not deeply affect the measurement score. Secondly, these n-grams ($n \leq 2$) is mainly used for the constraint rule. This reduces a lot of “noisy” pairs. Thirdly, deploying the independent assumption helps us limit a lot of other n-words overlapping phrases which are the “noisy” matchings. Last but not least, adding a “zooming” similarity value n^2 is an appropriate way to enhance the similarity value for the bitext pairs in a “noise” non-parallel corpus.

8. Experimental Results of Using Bootstrapping Scheme

Our main goal is to extract, from an in-domain training corpus (the data collected from the Subtitle resource), parallel training data to use for improving the performance of an out-of-domain-trained SMT system (the data collected from Wikipedia resource). Thus, we evaluate our extracted corpora by showing that when adding them to the out-of-domain training data of a baseline MT system it can improve the MT system’s performance significantly. We start with a small parallel data. By using the bootstrapping technique, at each iteration we obtain a new training set of parallel data and consequently improve the performance of the MT system.

8.1. Applying the Bootstrapping Scheme - Evaluation 1

This evaluation tests the re-training with the re-extracting capacity of our proposed model. We extract 9,998 parallel links from Wikipedia (via Wikipedia’s hyperlink structure) and use this resource for evaluating the scalability of extracting new parallel sentences. Note that by using the first step and the second step of the parallel sentence detecting algorithm, we remove a large number of “noise” candidates from the whole candidates (tens millions of parallel sentence pair candidates). Consequently, there are only about 958,800 remaining candidates to be used for the third step.

Hence, in this model we apply the bootstrapping scheme and use our method for detecting parallel sentences. Each time when the training corpus is extended, we re-train the SMT system again. Consequently, we apply the new SMT system (it is now stronger than the older) to re-check the candidates again to find out more new parallel sentence pairs. We manually build a test set containing 10,000 parallel sentence pairs getting from Wikipedia resource to test the improvement of the BLEU score of the phrase-based SMT system. Table 10 shows the experimental result:

Iterator	Training	BLEU(%)	Extracted
Iter. 1	50,000	**8.92**	22,835
Iter. 2	72,835	21.80	16,505
Iter. 3	89,340	23.01	4,742
Iter. 4	94,082	23.96	1,130
Iter. 5	**95,212**	**24.07**	0

Table 10. The results of using Bootstrapping scheme with our detecting method.

At first, the BLEU score is 8.92% obtained by using the initial training set. Hence, at the first iteration of the process of extending parallel corpus, we achieve 22,835 new parallel sentence pairs. To get more parallel sentences, we re-train our SMT system with the new training set containing 72,835 parallel sentence pairs, and consequently the BLEU score is improved up to 21.80%. Then, the SMT system continues extracting more 16,505 new parallel sentence pairs which were not extracted at the previous iterations according to the lack of the capacity of the SMT system. This result expresses that the SMT system is now upgrading its translation ability. At the end, we can extract in the total of 45,212 new parallel sentence pairs and the BLEU score reaches to 24.07% that is far from the beginning score.

8.2. Applying Bootstrapping Scheme - Evaluation 2

This evaluation is performed in the same condition as the previous evaluation. However, we try to extract parallel texts from a larger test set of candidates so that we can obtain more parallel sentences. Table 11 presents more clearly the result with two major achievements. Firstly, from the limited knowledge domain in the initial corpus, after the learning process we can obtain a new data set of parallel sentences which stay in some new knowledge domains. The experiment shows a high improvement of BLEU score when we use the test set that belongs to the same domain of the extracted corpora. Secondly, this result also emphasizes again that by this method we can yield a large corpus of parallel sentence pairs with high quality.

Iterator	Training	BLEU(%)	Extract
Iter. 1	50,000	**8.92**	60,151
Iter. 2	110,151	25.85	37,287
Iter. 3	147,438	29.31	12,297
Iter. 4	159,735	31.11	3,157
Iter. 5	162,892	31.43	151
Iter. 6	**163,043**	**31.43**	0

Table 11. The results of integrating reinforcement learning with our detecting method (larger test set).

For both Evaluation 1 and Evaluation 2 we can see the number of extracted parallel sentence pairs is significantly larger than the number of parallel sentence pairs in the initial corpus. Note that these extracted sentences are much longer than the sentences in the initial training data. Interestingly and importantly, they cover a lot of diversity fields without concerning the Subtitle resource. As a result, quality of the new SMT is far better than that of the initial system (24.07 and 31.3 vs 8.92).

9. Conclusion

This work proposes a framework for automatically exploiting comparable non-parallel corpora. First, we propose a high quality similarity measurement which is used as the core for the extracting framework that gives a significantly better performance than previous works. We then integrate that method of detecting parallel texts into the Bootstrapping learning scheme for extending the parallel corpus, and hence improving the SMT system.

Various experiments have been conducted and the obtained results have shown the proposed algorithm of detecting parallel sentences obtains a significantly better recall in comparison with the previous studies. By combining the detecting component and the Bootstrapping learning scheme, we could be able to allow the SMT system efficiently extract more than 5 millions of words bitext English-Vietnamese data. In addition, the SMT system upgrades gradually with its improved ability of translation, especially for covering new domains of knowledge.

Acknowledgment This work is supported by NAFOSTED (Vietnam's National Foundation for Science and Technology Development) under project number 102.99.35.09.

References

[1] AbduI-Rauf, S., Schwenk, H.: On the use of comparable corpora to improve SMT performance, *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009.

[2] Abdul-Rauf, S., Schwenk, H.: Exploiting comparable corpora with TER and TERp, *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*, BUCC '09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, ISBN 978-1-932432-53-4.

[3] Abdul Rauf, S., Schwenk, H.: Parallel sentence generation from comparable corpora for improved SMT, *Machine Translation*, **25**(4), December 2011, 341–375, ISSN 0922-6567.

[4] Achananuparp, P., Hu, X., Shen, X.: The Evaluation of Sentence Similarity Measures, *Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, DaWaK '08, Springer-Verlag, Berlin, Heidelberg, 2008, ISBN 978-3-540-85835-5.

[5] Adafre, S. F., de Rijke, M.: Finding Similar Sentences across Multiple Languages in Wikipedia, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006, 62–69.

[6] Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness, *Proceedings of the 18th international joint conference on Artificial intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[7] Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., Lai, J. C.: Class-Based n-gram Models of Natural Language, *Computational Linguistics*, **18**, 1992, 467–479.

[8] Brown, P. F., Lai, J. C., Mercer, R. L.: Aligning sentences in parallel corpora, *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, ACL '91, Association for Computational Linguistics, Stroudsburg, PA, USA, 1991.

[9] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., Mercer, R. L.: The mathematics of statistical machine translation: parameter estimation, *Comput. Linguist.*, **19**, June 1993, 263–311, ISSN 0891-2017.

[10] Chen, S. F.: Aligning sentences in bilingual corpora using lexical information, *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, Association for Computational Linguistics, Stroudsburg, PA, USA, 1993.

[11] Chen, S. F., Goodman, J.: An empirical study of smoothing techniques for language modeling, *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, Association for Computational Linguistics, Stroudsburg, PA, USA, 1996.

[12] Do, T. N., Besacier, L., Castelli, E.: A Fully Unsupervised Approach for Mining Parallel Data from Comparable Corpora, *European COnference on Machine Translation (EAMT) 2010*, Saint-Raphael (France), June 2010.

[13] Do, T. N., Besacier, L., Castelli, E.: UNSUPERVISED SMT FOR A LOW-RESOURCED LANGUAGE PAIR, *2d Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU 2010)*, Penang (Malaysia), May 2010.

[14] Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[15] Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: Improving the extraction of bilingual terminology from Wikipedia, *ACM Trans. Multimedia Comput. Commun. Appl.*, **5**(4), November 2009, 31:1–31:17, ISSN 1551-6857.

[16] Gale, W. A., Church, K. W.: A program for aligning sentences in bilingual corpora, *Comput. Linguist.*, **19**, March 1993, 75–102, ISSN 0891-2017.

[17] Hoang, C., Cuong, L. A., Thai, N. P., Bao, H. T.: Exploiting Non-Parallel Corpora for Statistical Machine Translation, *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on*, 2012.

[18] Hoang, C., Le, C.-A., Pham, S.-B.: Improving the Quality of Word Alignment by Integrating Pearson's Chi-Square Test Information, *Proceedings of the 2012 International Conference on Asian Language Processing*, IALP '12, IEEE Computer Society, Washington, DC, USA, 2012, ISBN 978-0-7695-4886-9.

[19] Hoang, C., Le, C. A., Pham, S. B.: Refining lexical translation training scheme for improving the quality of statistical phrase-based translation, *Proceedings of the Third Symposium on Information and Communication Technology*, SoICT '12, ACM, New York, NY, USA, 2012, ISBN 978-1-4503-1232-5.

[20] Hoang, C., Le, C. A., Pham, S. B.: A Systematic Comparison between Various Statistical Alignment Models for Statistical English-Vietnamese Phrase-Based Translation, *Proceedings of the 2012 Fourth International Conference on Knowledge and Systems Engineering*, KSE '12, IEEE Computer Society, Washington, DC, USA, 2012, ISBN 978-0-7695-4760-2.

[21] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation, *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, Association for Computational Linguistics, Stroudsburg, PA, USA, 2007.

[22] Koehn, P., Och, F. J., Marcu, D.: Statistical phrase-based translation, *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003.

[23] Lau, R., Rosenfeld, R., Roukos, S.: Adaptive language modeling using the maximum entropy principle, *Proceedings of the workshop on Human Language Technology*, HLT '93, Association for Computational Linguistics, Stroudsburg, PA, USA, 1993, ISBN 1-55860-324-7.

[24] Lopez, A.: Statistical machine translation, *ACM Comput. Surv.*, **40**(3), 2008.

[25] Munteanu, D. S., Marcu, D.: Improving Machine Translation Performance by Exploiting Non-Parallel Corpora, *Comput. Linguist.*, **31**, December 2005, 477–504, ISSN 0891-2017.

[26] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002.

[27] Ponzetto, S. P., Strube, M.: Knowledge derived from wikipedia for computing semantic relatedness, *J. Artif. Int. Res.*, **30**, October 2007, 181–212, ISSN 1076-9757.

[28] Simard, M., Foster, G. F., Isabelle, P.: Using cognates to align sentences in bilingual corpora, *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing - Volume 2*, CASCON '93, IBM Press, 1993.

[29] Smith, J. R., Quirk, C., Toutanova, K.: Extracting parallel sentences from comparable corpora using document level alignment, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, ISBN 1-932432-65-5.

[30] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation, *In Proceedings of Association for Machine Translation in the Americas*, 2006.

[31] Tyers, F., Pienaar, J.: Extracting Bilingual Word Pairs from Wikipedia, *in Proceedings of the SALTMIL Workshop at Language Resources and Evaluation Conference, LREC08*, 2008.