International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39501-39508
© Research India Publications. http://www.ripublication.com

Learning Bilingual Lexica from Non-parallel Comparable Corpora Using Perceptron Learning

Jae-Hoon Kim

*Professor,Department of Computer Engineering,Korea Maritime and Ocean University,
727 Taejong-ro, Yeongdo-Gu,Busan 49112, Republic of Korea,*jhoon@kmou.ac.kr

Hyeong-Won Seo

*Ph.D Student,Department of Computer Engineering,Korea Maritime and Ocean University,
727 Taejong-ro, Yeongdo-Gu,Busan 49112, Republic of Korea,*wonn24@gmail.com

Hong-Seok Kwon

*MS Student,Department of Computer EngineeringKorea Maritime and Ocean University
727 Taejong-ro, Yeongdo-Gu,Busan 49112, Republic of Korea,*hong8c@naver.com

Min-Ah Cheon

*MS Student,Department ofComputer EngineeringKorea Maritime and Ocean University
727 Taejong-ro, Yeongdo-GuBusan 49112, Republic of Korea,*minah014@outlook.com

Abstract

In this paper, we propose a new method for finding words that are translations of each other. The method is based on non-parallel comparable corpora and the perceptron learning algorithm. Non-parallel comparable corpora are used for learning input vectors and output vectors of the Perceptron learning algorithm, which learns a function that maps input vectors to output vectors. In this work, the input and output vectors are synonym vectors for source and target languages, respectively. Unlike the general Perceptron learning algorithm, in this approach, there are no desired vectors in the training examples. Instead, the desired vectors are dynamically selected according to thesimilarity between the output vector and the synonym vectors of the target words during learning. We extract bilingual dictionaries by iteratively applying our proposed method. Experiments were conducted on two different language pairs, bi-directional Korean-Spanish and Korean-French. The empirical results show that our proposed method significantly improvesthe performance for the top rank candidate.

Keywords: Bilingual dictionary, Perceptron learning algorithm, Non-parallel comparable corpus.

Introduction

In the study of natural language processing, bilingual lexica that contain source words and their translations as target words are important linguistic resources. For example, bilingual lexica are used to translate source texts into target texts for a statistical machine translation system [1] and are used in the query translation for cross lingual information retrieval [2]. Bilingual lexica, however, are not available for all language pairs. Basically, bilingual lexica can be obtained by manually extracting appropriate translation pairs for each language pair, but this is an extremely time-consuming and labor-intensive process. For these reasons, many researchers have focused on automatic bilingual lexicon extraction (BLE). The most direct and simple automatic BLE aligns words using parallel corpora [3], which contain source texts and their translations. However, collecting a large amount of parallel corpora is onerous and restricted to specific domains in some less-known language pairs. For all these reasons, researchers turn to extracting bilingual lexica from comparable corpora [4−6], which are pairs of corpora in two different languages that are related by certain characteristics such as event, domain, topic, date, and/or subject.

One BLE approach is a context-based approach that uses information retrieval techniques [7−11]. This approach has achieved significantly good performance on high-frequency words, but a large-scale seed dictionary is required to translate context vectors. Recently, Chatterjee *et al.* [12] and Chu *et al.* [13] proposed iterative approaches that extract new translation candidates, use the candidates as a new seed dictionary, and repeat the procedure until convergence. These iterative approaches have been shown to significantly improve their accuracy within a few epochs.

We propose an iterative method for extracting bilingual lexica that is similar to an iterative approach, but is based on the Perceptron learning algorithm [14], which is a type of neural network algorithm. The input and output of the algorithm consist of synonym vectors of source and target words, respectively. That is, the representation of an input word (or output word) is a synonym vector of a source word (or target word). The synonym vector can alleviate somewhat the data sparseness problem, although we discuss this issue further later. The synonym vector on the input layer (or output layer) is estimated by the similarity of context vectors for source words (or target words) generated through a source comparable corpus (or target comparable corpus). Although we use the Perceptron learning algorithm, which is a supervised learning algorithm, we make no use of any training examples that map input vectors to output vectors corresponding to the desired vectors. Thus, we call this method *unsupervised Perceptron learning*. The desired vector is dynamically selected by the similarity between the output vector and synonym vectors of target words during learning. As a result, the most similar synonym vector becomes the desired vector. In this paper, we extract bilingual lexica by iteratively applying the proposed method.

The remainder of this paper is organized as follows. Section 2 reviews work related to the task. Section 3 describes the unsupervised Perceptron algorithm. Section 4 presents the overall architecture of BLE systems. In Section 5 we present details of the experiments on two bi-directional language pairs. Section 6 further discusses the results. Finally, the conclusion and directions for future work are given in the last Section.

Related Work

A. BLE based on comparable corpora

The widely used standard approach to BLE is a context-based approach that uses information retrieval techniques [4, 7]. Generally, the standard approach builds context vectors for each source and target word. A source word is represented by a vector along with its contextual words and a target word is also represented in the same way. Here, contextual words are weighted by their degree of association. However, the source and target vectors cannot be represented in the same space because they are made up of words in different languages. For these reasons, the source vectors have to be translated into the target language using an initial seed dictionary. In this translation process, the volume of the initial seed dictionary is very important. A larger initial seed dictionary represents the source vectors more accurately in the target vector space. Therefore, the source word can be represented in the target vector space and then compared with the target vectors in that space. This approach has achieved very good performance for high-frequency words, but a large-scale seed dictionary is required to translate the context-vectors and can affect the performance of the system.

The standard approach uses comparable corpora and a seed dictionary [4, 7]. Its performance, however, is dependent on the size and quality of the seed dictionary and, moreover, constructing a large, high-quality seed dictionary is tedious and expensive. Kwon et al. [15] proposed the pivot-based approach (PA) that uses two parallel corpora that share a pivot language such as English with more accurate alignment information instead of comparable corpora. The pivot language represents both source context vectors and target context vectors, which are comparable to each other because they have the same dimension when represented in the pivot language. As a result, the PA does not need an initial seed dictionary. In addition, the PA uses a freely available word aligner called Anymalign [16], to construct context vectors. Anymalign can extract translation candidateswith high accuracy for low-frequency words. The PA can be summarized in the following three steps:

(1) Build a source context vector and target context vector for each word in a source language (e.g., Korean) and target language (e.g., Spanish) using the two independent parallel corpora Korean-English and English-Spanish, respectively. All words in the context vector are weighted by Anymalign.
(2) Calculate the cosine similarity between source context vectors and target context vectors.
(3) Sort the target words for a source word based on their similarity scores and select the top k target words as translation candidates for the source word.

Another recent method using comparable corpora extracts new translation candidates from a seed dictionary and then iteratively uses the translation candidates as a new seed dictionary. This method is called an iterative approach, and has been presented in Chatterjee et al. [12] and Chu et al. [13]. Chu et al. proposed a BLE system that uses topical and contextual knowledge in the iterative process. The system consists of two main methods, namely the topic model based method (TMBM) and context based method (CBM). The TMBM measures the similarity of two words on cross-lingual topical distributions, while the CBM measures the similarity of contextual distributions across languages. In their study, exploiting both topical and contextual knowledge can make bilingual extraction more reliable and accurate than only using one knowledge source. Chu et al. conducted experiments on Chinese-English and Japanese-English Wikipedia data, showing that their method can significantly improve its performance in the first few epochs.

B. Perceptron learning algorithm

A perceptron is a type of artificial neural network introduced by Rosenblatt [14] for classification. The Perceptron performs supervised classification of an input $\boldsymbol{x}$ into one of several possible non-binary outputs. It is a linear classifier, which is a function that maps its input $\boldsymbol{x}$ (a real-valued vector) to an output value $y = f(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = f(x_1, \dots, x_n) = \begin{cases} 1, \text{if} \sum_{i=0}^{n} w_i x_i > 0 \\ -1, \text{otherwise} \end{cases} \quad (1)$$

where w is a vector of real-valued weights. Further, w_0 in w is a bias that does not depend on any input value. Thus, x_0 is one. Each w_i is a real-value weight that reflects the importance of input x_i to the Perceptron output. The Perceptron algorithm is called a single-layer Perceptron with one output neuron to distinguish it from the multi-layer Perceptron [17]. Below is the learning algorithm for an input vector $\boldsymbol{x}$ and its desired value d:

(1) Initialize the weights w with small random values.
(2) Calculate the output $y = f(\boldsymbol{x})$ using Eq. (1).
(3) Update the weights according to

$$w_i = w_i + \Delta w_i \quad (2)$$

where $\Delta w_i$is $\alpha(d - y)x_i$and α is a learning rate.

(4) Repeat Steps (2) and (3) until a certain condition holds, for instance, when the iteration error is less than a user-specified error threshold or the predetermined number of iterations has been completed.

To determine the iteration error, the mean squared error E is generally used and measures the average of the squares of the errors. It is computed over all of the training examples (input/output vector pairs$\{(\boldsymbol{x}_i, d_i) \mid i = 1, \ldots, p\}$) and is given by:

$$E = \frac{1}{2}\sum_{i=1}^{p}\|d_i - y_i\|^2 \quad (3)$$

where p is the number of input/output vector pairs in the training examples, $\boldsymbol{x}_i$ and d_i, are the i-th input vector and i-th desired vector in the training examples, respectively, and y_i is the i-th output vector for a given input $\boldsymbol{x}_i$.

Unsupervised Perceptron Learning

In this section, we propose an unsupervised Perceptron learning method for BLE. Fig. 1 shows the Perceptron structure for BLE, which is the same structure as mentioned in Section 2.2, except it has multiple output neurons.

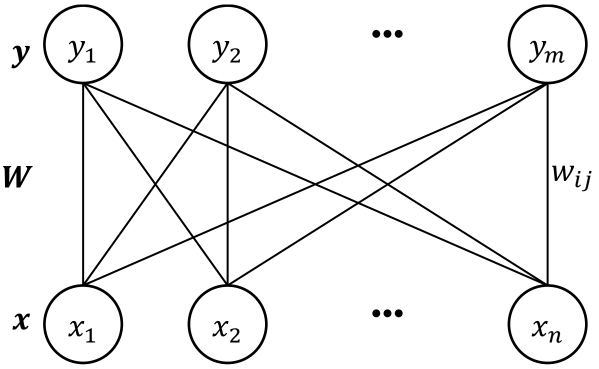

Fig. 1. Perceptron structure for bilingual lexicon extraction

Another significant difference is that there are no desired outputs for the input samples. The desired output $\boldsymbol{d}$ is dynamically labeled as follows:

$$\boldsymbol{d} = \text{argmax}_{t \in T}\text{sim}(\boldsymbol{y}, \boldsymbol{t}), \quad (4)$$

where $\text{sim}(\boldsymbol{y}, \boldsymbol{t})$ is a similarity function, a real-valued function that quantifies the similarity between the output vector $\boldsymbol{y}$ andsynonym vector $\boldsymbol{t}$ from the set of synonym vectors $\boldsymbol{T} = \{\boldsymbol{t}_j | j = 1, \ldots, m\}$. It is possible to compare $\boldsymbol{y}$ with $\boldsymbol{t}$ because output vector $\boldsymbol{y}$ is considered to be a translation equivalent for source word $\boldsymbol{x}$. For a given training input vector $\boldsymbol{x}$, the unsupervised Perceptron learning algorithm for BLE is as follows:

(1) Initialize weights $\boldsymbol{W} = [w_{ij}]_{n \times m}$ with a small seed dictionary, where w_{ij} is an association score between the i-th source word and the j-th target word, similar to word association [18] and word similarity. The seed dictionary can be extracted using one of the previous approaches such as the standard approach [5] or PA [15].

(2) Calculate the output $\boldsymbol{y} = f(\boldsymbol{x})$ using Eq.(5)

$$\boldsymbol{y} = f(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{x}\cdot\boldsymbol{w}_i}} \quad (5)$$

(3) Choose the desired output vector $\boldsymbol{d}$for input $\boldsymbol{x}$ using Eq. (4)

(4) Update the weight between the i-th source word and j-th target word according to

$$w_{ij} = w_{ij} + \alpha\begin{cases}(d_j - y_j)x_i & \text{if}(d_j - y_j) > 0 \\ 0 & \text{otherwise}\end{cases} \quad (6)$$

where α is a learning rate.

(5) Repeat Steps (2) through (4) until either the iteration error is less than a user-specified threshold or the predetermined number of iterations has been completed.

To sum up, there are a few differences between the standard Perceptron and the unsupervised Perceptron: First, there are no desired outputs, as mentioned before. Second, the input vector (or output vector) is the synonym vector for a word in a source language (or target language). That is, the attributes or features are synonyms found from a source comparable corpus (or target comparable corpus). We use synonym vectors to address the data sparseness problem. Suppose that synonyms of the word “father” are “dad,” “daddy,” and “papa,” including “father” itself, and translation candidates for “father” exist in an initial seed lexicon, but translation candidates for “daddy” do not. The weights for the translation candidates for “daddy” are gradually turned on when the synonym vector of “father” is repeatedly exposed on the input layer. As a result, translation candidates for the word “daddy” can be extracted, even though they did not exist in the initial seed bilingual lexicon. Third, the weights of the Perceptron can be initialized using a seed bilingual lexicon extracted using the previous approaches, but not randomly. Fourth, all weights are positive.

Proposed BLE System

In this section, we present the overall architecture for BLE and then explain how to automatically generate bilingual lexica in detail.

A. System architecture

The overall structure of the proposed method is depicted in Fig. 2. Our method consists of two procedures: building a bilingual lexicon using the PA [15, 19] as a seed bilingual lexicon and extracting the final bilingual lexicon using the unsupervised Perceptron learning algorithm described in Section 3.

The first procedure is depicted on the left side of Fig.2**.** Using the PA [15], it builds a bilingual lexicon that is used as the initial weights of the unsupervised Perceptron. The procedure constructs both source context vectors and target context vectors represented by same words in the pivot language. The source and target context vectors are obtained from source-pivot and pivot-target parallel corpora, respectively. The bilingual lexicon is then constructed by comparing the two context vectors.

The second procedure is portrayed on the right side of Fig. 2. Using the unsupervised Perceptron learning algorithm as described in Section 3, it extracts the final bilingual lexicon. The procedure constructs source synonym vectors from a source comparable corpus and target synonym vectors from a target comparable corpus, respectively. The bilingual lexicon is then extracted using the unsupervised Perceptron learning algorithm and the seed bilingual lexicon is updated using the bilingual lexicon. This process is continually repeated until the number of iterations reaches a predetermined limit; hence, we call this process the Perceptron-based iterative approach (PIA). Further details are described in subsequent sections.

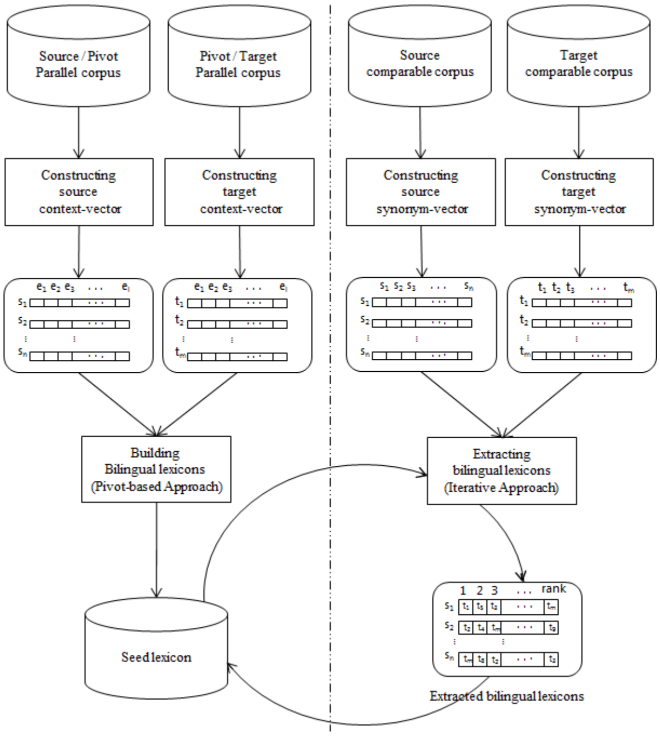

Fig. 2. Overall structure of the proposed method

B. Constructing synonym vectors

To construct a synonym vector for a word, we first build a context vector for it that is obtained from words occurring close to it in texts in a single language in the same way as the context-based approach [7]. In the proposed approach, context is represented as words that occur within a fixed window size of ± 2. The words in a context vector are weighted by X^2 scores and are selected using a critical value threshold of 3.841. The source and target context vectors are respectively constructed from source and target comparable corpora in the same way.

We next construct source synonym vector $s(\mathrm{w})$ for source wordw according to the similarity score between two source context vectors as follows:

$$\boldsymbol{s}(w) = \left[\cos\left(\boldsymbol{c}(w),\boldsymbol{c}(w_1)\right),\cos\left(\boldsymbol{c}(w),\boldsymbol{c}(w_2)\right),\ldots,\cos\left(\boldsymbol{c}(w),\boldsymbol{c}(w_n)\right)\right]^{\mathrm{T}} \quad (7)$$

where $\boldsymbol{c}(w)$ is a context vector for w and n is the number of source words. The target synonym vector $\boldsymbol{t}(w)$ for target word w is formed in the same way as the source synonym vector.

C. Extracting bilingual lexica

Before extracting bilingual lexica, we first construct a bilingual lexicon using the PA, as mentioned before. The lexicon is used for an initial seed bilingual lexicon of the unsupervised Perceptron, which maps source words to target words, just like the seed dictionary in the context-based approach [7]. Therefore, the source synonym vectors are input into the unsupervised Perceptron and can be translated into their corresponding target synonym vectors. The PIA consists of the following steps:

(1) Build source synonym vectors $\boldsymbol{S} = \{\boldsymbol{s_1},\boldsymbol{s_2},\ldots\boldsymbol{s_n}\}$and target synonym vectors $\boldsymbol{T} = \{\boldsymbol{t_1},\boldsymbol{t_2},\ldots,\boldsymbol{t_m}\}$ as described in Section 4.2.

(2) Generate translated vector $\boldsymbol{y}$ using the unsupervised Perceptron algorithm for a given source synonym vector $\boldsymbol{x} \in \boldsymbol{S}$.

(3) Determine the desired synonym vector $\boldsymbol{d}$ of $\boldsymbol{x}$ using Eq. (4)

(4) Update $\boldsymbol{W}$ via the unsupervised Perceptron learning algorithm.

(5) Repeat Steps (2) through (4) until convergence.

(6) Obtain translation candidates $T(w)$ for all source words w as follows:

(6-1) Obtain the synonym vector $\boldsymbol{x} = \boldsymbol{s}(w) \in \boldsymbol{S}$

(6-2) Compute $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \{y_1, y_2, \ldots, y_m\}$

(6-3) Sort $\boldsymbol{y}$ by value and select the top k target words from the sorted list.

Experiment and results

In this section, we bi-directionally evaluate our approach for two different language pairs: Korean-Spanish (KR-ES) and Korean-French (KR-FR). For evaluation metrics, we use accuracy, mean reciprocal rank (MRR), and recall; these are metrics that are widely used in BLE [19].

A. Experimental setup

i. Comparable corpora

We built three comparable corpora for Korean, Spanish, and French. Sentences in the corpora were taken from news articles on the Web (Korean from www.naver.com, Spanish from www.abc.es, and French from www.lemonde.fr) and were supplemented by adding the existing corpora such as the Europarl corpus [20]. All corpora contain 800,000 sentences each and their statistics are shown in TABLE 1 in detail.

TABLE 1. Statistics of comparable corpora

	Korean (KR)	Spanish (ES)	French (FR)
The number of sentence pairs	800,000	800,000	800,000
The average number of wordsper sentence	16.2	15.9	16.1
The number Ofdistinct nouns	16,000	5,900	4,900
Domain	International news (51%)	International news (32%)	International news (61%)
	Not categorized news (49%)	Europarl corpus (68%)	Europarl corpus (39%)

ii. Data pre-processing

All words were tokenized using the following tools: U-tagger [21] for Korean and Tree-Tagger [22] for English, Spanish, and French. In the case of Korean, multiword expressions that were composed of more than four characters were decomposed by U-tagger. For example, the Korean word “인공지능(artificial intelligence)” was decomposed into “인공(artificial)” and “지능(intelligence)” because the proposed system is targeted towards extracting single words. All words in English, Spanish, and French were converted to lower case, and those in Korean were morphologically analyzed into morphemes and POS-tagged by the U-tagger. Next, only content words[1] that occurred more than five times were considered when generating context vectors in all languages. Finally, the comparable corpora comprised about 16,000 distinct nouns in Korean, 5,900 in Spanish, and 4,900 in French.

iii. Building evaluation dictionary

We manually built four evaluation dictionaries (KR-ES, KR-FR, ES-KR, and FR-KR) using the Web dictionary (http://dic.naver.com) to evaluate the performance of the proposed method. Each lexicon is unidirectional, meaning that it lists the meanings of words of one language in another. The evaluation dictionary contains 150 high-frequency words (denoted by HIGH hereafter) and 150 low-frequency words (denoted by LOW hereafter). TABLE 2 shows the average number of translations per source word in each lexicon. The number indicates the degree of ambiguity.

TABLE 2. Average number of the translations per source word in the evaluation dictionaries for IA

Evaluation dictionary	HIGH	LOW
KR-ES	9.1	5.3
KR-FR	8.8	7.0
ES-KR	12.5	8.7
FR-KR	12.0	8.5

B. Performance evaluation

We conducted 60 epochs with a learning rate of α = 0.01 for the KR-ES, KR-FR, ES-KR, and FR-KR language pairs. Accuracy, MRR, and recall were used as evaluation metrics.

i. Performance of accuracy@1

The accuracy@1 means the accuracy of the top 1. The accuracy@1 of HIGH and LOW are shown in Fig. 3 and 4. As shown in the figures, the accuracy@1 of HIGH slightly increased over 60 epochs. The accuracy@1 of KR-ES increased from 0.366 to 0.406, that of KR-FR increased from 0.413 to 0.440, that of ES-KR increased from 0.400 to 0.420, and that of FR-KR increased from 0.500 to 0.506. For the LOW, the accuracy@1 of KR-ES improved from 0.187 to 0.207, that of KR-FR improved from 0.353 to 0.373, that of ES-KR improved from 0.220 to 0.240, and that of FR-KR improved from 0.286 to 0.293.

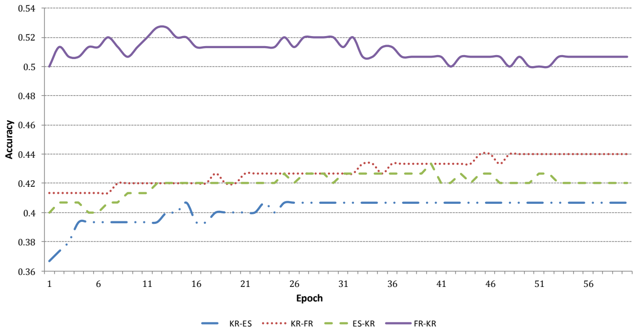

Fig. 3. Accuracy@1 for HIGH

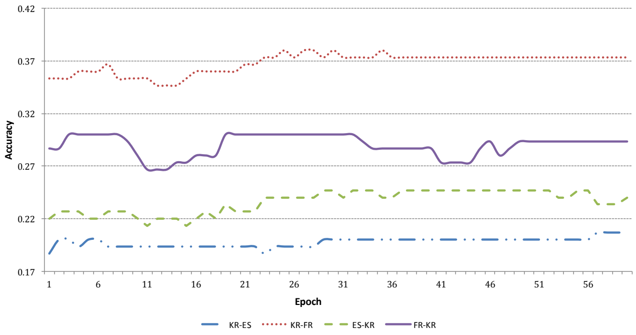

Fig. 4. Accuracy@1 for LOW

[1] Korean (Sejong tagset): NNG, VV, VA, MAG, SL
Spanish (Penn Treebank tagset): NC, NMEA, NP, PE, ACRNM, NMON, ADJ, ADV, UMMX, VCLIger, VCLIinf, VCLIfin, VEadj,VEfin, VEger, VEinf, VHadj, VHfin, VHger, VHinf, VLadj, VLfin, VLger, VLinf, VMadj, VMfin, VMger, VMinf, VSadj, VSfin, VSger, VSinf
French (Penn Treebank tagset): ABR, NOM, ADJ, ADV, INT, VER

ii. Performance of MRR

The MRR of HIGH and LOW for the top five ranks is shown in Fig. 5 and 6, respectively. As shown in the figures, the MRR increased by about 0.014 (KR-ES) and 0.029 (KR-FR). For ES-KR and FR-KR, the MRR decreased by about 0.004 and 0.013, respectively. The MRR of LOW increased by about 0.014 for KR-ES and decreased by about 0.001, 0.001, and 0.01 for KR-FR, ES-KR, and FR-KR, respectively. The reason for decreasing MRR is that the PIA is largely dependent on the synonym vectors. If the synonym vectors are inaccurate, the unsupervised Perceptron learning algorithm can learn incorrectly. As a result, the system cannot find the correct translation candidates. In our experiments, the generated synonym vectors were noisy. Therefore, the system performances of the top two and higher ranks decreased as the number of epochs increased.

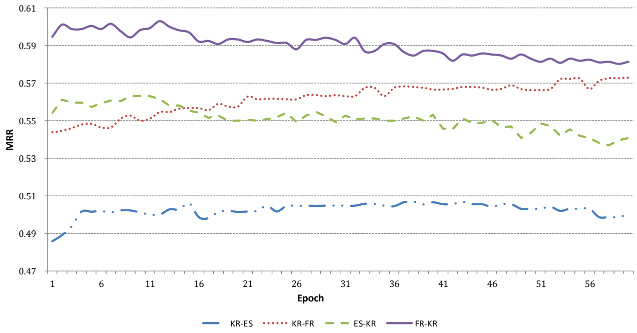

Fig. 5.MRR for HIGH

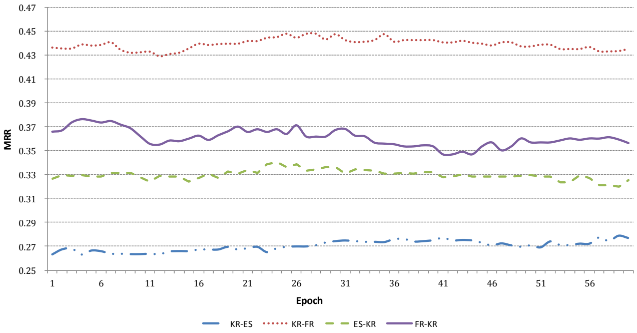

Fig. 6. MRR for LOW

iii. Performance of recall

Lastly, the recalls of the HIGH and LOW words for the top five ranks are presented in TABLE 3. As seen in the table, the best recall is 22.5% for KR-FR for HIGH and the worst recall is 9.0% for FR-KR for LOW. One of the reasons why this has occurred is that words do not belong to various domains and our language data sets (except for Korean) come from only international news articles.

TABLR 3.Recalls of IA for HIGH and LOW for the top five ranks

Language pairs	Top 20 Recall	
	HIGH	LOW
KR-ES	17.0%	15.0%
KR-FR	22.5%	18.3%
ES-KR	12.2%	10.9%
FR-KR	10.9%	9.0%

Discussion

In this section, we discuss several issues raised by our experiments

A. Number of epochs to convergence

In the experiment, we conducted 60 epochs. The accuracy gradually increased during the first several epochs and stabilized afterwards. The reason for this is a characteristic of the Perceptron algorithm. Rosenblatt proved that if the inputs presented from more than two classes are separable, then the Perceptron convergence procedure converges between those classes in finite time. The second reason is that there is a limitation on the performance. After several epochs, the performance approaches that limitation and is difficult to further improve; thus, the performance converges. We conclude that the iteration number at which the performance converges depends on the particular experimental settings such as the synonym threshold, dictionary threshold, number of translated vector attributes, and learning rate of the unsupervised Perceptron algorithm.

B. Performance on different language pairs

In our experiments, in Fig. 3, we can see that the performance on two bi-directional different language pairs, KE-ES and KE-FR, significantly improved. This indicates that the proposed method is language independent.

C. Performance for different synonym thresholds

We conducted the experiments using threshold values that ranged between 0.05 and 0.5. The effects of this variation are presented in TABLE 4.

TABLE 4.Effects of various threshold values

Synonym threshold	Reliability of the information	# of vector attributes	Dimension Of the vector	Recall	Accuracy
θ(↑)	↑	↓	↓	↓	↑
θ(↓)	↓	↑	↑	↑	↓

A higher threshold generates more reliable vectors. This means that the synonym vector becomes more accurate and the initial dictionary may have more accurate translation candidates. It also leads to better accuracy. Moreover, the number of vector attributes and its dimension decrease. Finally, it also reduces the time complexity of the Perceptron algorithm. However, if the threshold is set too high, it can exclude some correct synonyms in the synonym vectors or translation candidates from the initial seed dictionary, thus decreasing recall. On the other hand, a lower threshold keeps more information so that the recall increases but the accuracy decreases. Therefore, we set the synonym threshold to 0.2.

D. Performance on the different number of translated vector attributes

In Section 5.2, we restricted the maximum size of the translated vector attributes. We did this to reduce the computational cost of calculating similarity. Like the threshold, the number of translated vector attributes affects the results. More attributes increase the percentage of words where the correct translation is contained within the top *N* ranks, but it also leads to more noise and is more time consuming. Therefore, we set a small number of attributes, such as 50, and find that this is appropriate for our proposed method.

E.Error analysis

In the proposed method, two problems that affect performance are the inaccurate representation of synonym vectors and semantic distinctions of a word. As seen in Fig. 5, the MRRs decreased over several epochs on some language pairs. The reason for this is that our system represents words in vectors using their synonyms and extracts the translation candidates from the most similar target synonym vector. Therefore, the system performance is very dependent on synonym vectors. However, synonym extraction is a difficult task to achieve and evaluate. TABLE 5 shows a partial example of the synonym vectors for Korean. The synonym vector of the word is noisy, except for itself. Therefore, the performance of ranks 2 and below decreased as the number of epochs increased.

TABLE 5.Partial examples of the Korean synonym vector

Word	Synonym								
학교 (school)	**학교 (school)**	**1.00**	신학교 (seminary)	0.81	각급 (each class)	0.41	학생 (student)	0.38	...
가격 (price)	**가격 (price)**	**1.00**	하락 (fall)	0.47	동급 (same level)	0.42	인하 (reduction)	0.37	…
고객 (client)	**고객 (client)**	**1.00**	증서 (certificate)	0.89	예탁 (deposition)	0.22	만족도 (satisfaction)	0.11	…
경기 (economy)	**경기 (economy)**	**1.00**	장기화 (long period)	0.76	여파 (aftereffect)	0.67	내수 (demand)	0.37	…

The second problem is semantic distinction. When we build the context vectors, we do not consider the meaning of the words. For example, the Korean word "가격(price)" has two meanings, "가격(price)" and "가격(hit)," which are used in different ways but are considered the same when generating the context vector of the Korean word "가격(price)." It makes the context vector noisy and inaccurate. We reserve this problem for future work.

Conclusions and Future Works

This paper presents a novel BLE method from comparable corpora. The proposed method consists of two approaches: PA and PIA. The PA uses parallel corpora, a pivot language, and word alignment tool. The word alignment tool is used to construct context vectors. The pivot language is exploited to represent the context vectors of both source and target languages; thus, an initial seed dictionary is not required to translate a source vector into the target language. The main contribution of the method proposed in this paper is the PIA. The PIA extracts bilingual lexica from comparable corpora and exploits a modified Perceptron algorithm called the unsupervised Perceptron algorithm. Starting from the PA, the PIA iteratively constructs a seed dictionary as weights that are learned by an unsupervised Perceptron algorithm. The basic characteristics of this approach are that it can further improve the BLE accuracy and needs no data provided by the training examples to learn weights via the unsupervised Perceptron algorithm. Our experimental results show that PIA improves accuracy.

There are still several directions for future work under consideration. Currently, the proposed method has many parameters that must be adjusted to improve the performance. In the future, we will adjust these parameters to further improve performance. In addition, we plan to expand the system to different categories in addition to nouns. Lastly, we plan to handle multi-word expressions.

Acknowledgement

This work was supported by the IT R&D program of MSIP/KEIT (10041807, Development of Original Software Technology for Automatic Speech Translation with Performance 90% for Tour/International Event Focused on Multilingual Expansibility and Based on Knowledge Learning).

References

[1] P. Brown, S. Pietra, and R. Mercer, "The mathematics of statistical machine translation: Parameter estimation," Computational Linguistics, vol. 19, no. 2, pp. 263-311, 1993.

[2] G. Grefenstette, The Problem of Cross-Language Information Retrieval, Springer USA, 1998.

[3] H. Wu and H. Wang, "Pivot language approach for phrase-based statistical machine translation," Proceedings of 45th Annual Meeting of the Association for Computational Linguistics (ACL'07), pp. 856-863, 2007.

[4] P. Fung, "Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus," Proceedings of the 3rd Workshop on Very Large Corpora (VLC'95), pp. 173-183, 1995.

[5] K. Yu and J. Tsujii, "Bilingual dictionary extraction from Wikipedia," Proceedings of the Machine Translation Summit XII, pp. 379-386, 2009.

[6] A. Ismail and S. Manandhar, "Bilingual lexicon extraction from comparable corpora using in-domain

terms," Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10), pp. 481-489, 2010.

[7] R. Rapp, "Identifying word translations in non-parallel texts," Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95), pp. 320-322, 1995.

[8] P. Fung, "A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora," Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98), pp. 1-16, 1998.

[9] E. Gaussier, J.-M. Renders, I. Matveeva, C. Goutte, and H. Dejean, "A geometric view on bilingual lexicon extraction from comparable corpora," Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL'04), pp. 526-533, 2004.

[10] A. Hazem and E. Morin, "Adaptive dictionary for bilingual lexicon extraction from comparable corpora," Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12), pp. 288-292, 2012.

[11] J.-H. Kim, H.-S. Kwon, and H.-W. Seo, "Evaluating a pivot-based approach for bilingual lexicon extraction," Computational Intelligence and Neuroscience, vol. 2015, Article ID. 434153, 13 pages, 2015.

[12] D. Chatterjee, S. Sarkar, and A. Mishra, "Co-occurrence graph based iterative bilingual lexicon extraction from comparable corpora," in Proceedings of the 4th International Workshop on Cross Lingual Information Access, pp. 35-42, 2010.

[13] C. Chu, T. Nakazawa, and S. Kurohashi, "Iterative bilingual lexicon extraction from comparable corpora with topical and contextual knowledge," Proceedings of the 15th Conference on Intelligent Text Processing and Computational Linguistics (CICLING'14), pp. 296-309, 2014.

[14] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," Psychological Review, vol. 65, no. 6, pp. 386-408, 1958.

[15] H.-S. Kwon, H.-W. Seo, and J.-H. Kim, "Enhancing performance of bilingual lexicon extraction through refinement of pivot-context vectors," Journal of KIISE: Software and Applications, vol. 41, no. 7, pp. 492-500, 2014.

[16] A. Lardilleux, Y. Lepage, and F. Yvon, "The contribution of low frequencies to multilingual sub-sentential alignment: A differential associative approach," International Journal of Advanced Intelligence, vol. 3, no. 2, pp. 189-217, 2011.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. William, "Learning internal representations by error propagation," D. E. Rumelhart, J. L. McClelland, and the PDP research group (eds), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press, 1986.

[18] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," Computational linguistics, vol. 16, no. 1, pp. 22-29, 1990.

[19] H.-S. Kwon, H.-W. Seo, and J.-H. Kim, "Bilingual lexicon extraction via pivot language and word alignment tools," Proceeding of the 6th International Workshop on Building and Using Comparable Corpora (BUCC'13), pp. 11-15, 2013.

[20] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," Proceedings of the Machine Translation Summit X, pp. 79-86, 2005.

[21] J. Shin and C. Ock, "A Korean morphological analyzer using a pre-analyzed partial word-phrase dictionary," Journal of KIISE: Software and Applications, vol. 39, no. 5, pp. 415-424, 2012.

[22] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," Proceedings of International Conference on New Methods in Language Processing, pp. 44-49, 1994.