Translating Collocation using Monolingual and Parallel Corpus

蔣明撰 Ming-Zhuan Jiang, 顏孜羲 Tzu-Xi Yen, 黃仲淇 Chung-Chi Huang,
陳玫樺 Mei-Hua Chen, 張俊盛 Jason S. Chang

國立清華大學資訊工程系
Dept of Computer Science
National Tsing Hua Univ.
{raconquer, joseph.yen, u901571, chen.meihua, jason.jschang}@gmail.com

Abstract

In this paper, we propose a method for translating a given verb-noun collocation based on a parallel corpus and an additional monolingual corpus. Our approach involves two models to generate collocation translations. The combination translation model generates combined translations of the collocate and the base word, and filters translations by a target language model from a monolingual corpus, and the bidirectional alignment translation model generates translations using bidirectional alignment information. At run time, each model generates a list of possible translation candidates, and translations in two candidate lists are re-ranked and returned as our system output. We describe the implementation of using method using Hong Kong Parallel Text. The experiment results show that our method improves the quality of top-ranked collocation translations, which could be used to assist ESL learners and bilingual dictionaries editors.

Keyword: collocation, statistical machine translation, computer-assisted translation

1 INTRODUCTION

A collocation is a recurrent combination of words that co-occur more frequently than expected by chance. Collocations can be classified into lexical and grammatical by the nature of their constituents. Another way of classifying collocations uses word positions to distinguish between rigid collocations and elastic collocations. Typically, a collocation consists of a base word and a collocate. Since collocations are used extensively, knowing the a right collocate for the base word plays an important role in second language learning as well as in machine translation. Translation of collocations is difficult for English as Second Language learners (ESL) because collocations are not always translated literally. For instance, the English collocation “delegate authority” can not be translated into “委託 機關”.

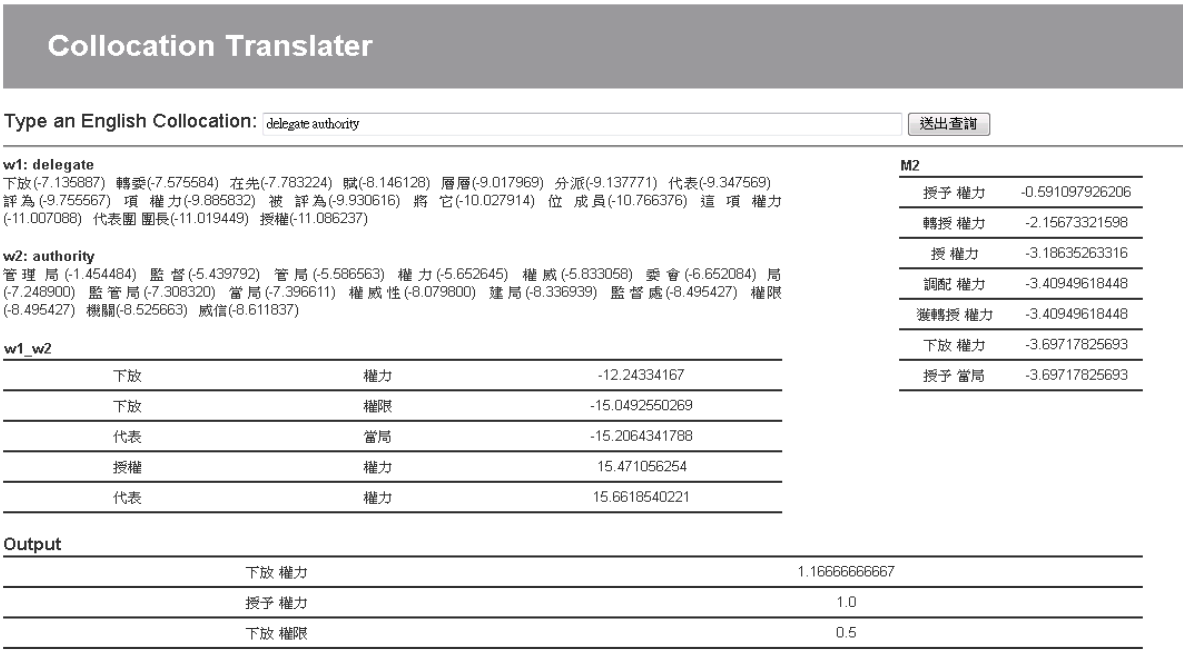
Much previous work has been done on collocation translation by extracting bilingual collocations pairs from parallel corpora. Recently, researchers have also proposed methods for retrieve collocations and their translation based on parsers and bilingual dictionaries. However, previous works using parallel corpora are mostly heuristic and methods based on bilingual dictionaries may be limited by the availability of broad-coverage dictionaries.

More recently, the mainstream Statistical Machine Translation (SMT) system like Moses has been widely used in many translation tasks such as translating texts and sentences. Unfortunately, the traditional SMT system does not take into consideration of the structure of collocations including variable word forms and non-contiguous phrases. Little work has been done on improving the SMT system for finding flexible collocations translation as a tool to assist ESL learners or to help the task of compiling bilingual collocation dictionaries.

Consider the elastic collocation "delegate ~ authority" and its translations. The translations of "authority" can be "機關", "權力" and "管理局" which are found in parallel corpus. The traditional SMT system can find "delegate some authority" as "下放 一些 權力", but usually there is no continuous "delegate authority" phrase translation in the parallel corpus. The SMT system might translate the collocation word by word, resulting in a incorrect translation, such as "委託 機關" (Figure 2). Intuitively, a English collocation translation should be also a Chinese collocation, and using an appropriate Chinese collocation set might filter out the incorrect translations, and leads to better translations such as "下放 權力". As shown in Figure 1, Google Translate surely has a good translation in this example.

Figure 1. Submitting a English Collocation "delegate authority" to Google Translate

In this paper, we propose a method that automatically translates the given collocation, by a combination word-based translation model and a bidirectional alignment translation model relying on aligned parallel corpora. A sample process of translating the collocation "delegate authority" is shown in Figure 2. The output translation candidates are generated by these two models.



Collocation Translater

Type an English Collocation: delegate authority 送出查詢

w1: delegate

下放(-7.135887) 轉委(-7.575584) 在先(-7.783224) 賦(-8.146128) 層層(-9.017969) 分派(-9.137771) 代表(-9.347569) 評為(-9.755567) 項 權力(-9.885832) 被 評為(-9.930616) 將 它(-10.027914) 位 成員(-10.766376) 這 項 權力(-11.007088) 代表團 團長(-11.019449) 授權(-11.086237)

w2: authority

管 理 局 (-1.454484) 監 督(-5.439792) 管 局 (-5.586563) 權 力 (-5.652645) 權 威 (-5.833058) 委 會 (-6.652084) 局 (-7.248900) 監 管 局 (-7.308320) 當 局 (-7.396611) 權 威 性 (-8.079800) 建 局 (-8.336939) 監 督 處 (-8.495427) 權限 (-8.495427) 機關(-8.525663) 威信(-8.611837)

w1_w2

下放	權力	-12.24334167
下放	權限	-15.0492550269
代表	當局	-15.2064341788
授權	權力	15.471056254
代表	權力	15.6618540221

M2

授予 權力	-0.591097926206
轉授 權力	-2.15673321598
授 權力	-3.18635263316
調配 權力	-3.40949618448
撥轉授 權力	-3.40949618448
下放 權力	-3.69717825693
授予 當局	-3.69717825693

Output

下放 權力	1.16666666667
授予 權力	1.0
下放 權限	0.5

Figure 2. An example of translating “delegate authority”

At runtime, the given collocation is first decomposed into two parts as base words and collocates, in order to obtain a set of possible word translations. The combined translations of two words are then generated. The additional translations are also generated if available from the bidirectional alignment translation model. Finally, the top 3 Chinese translation candidates of these two models are combined, ranked and returned.

The rest of the paper is organized as follows. Chapter 2 reviews related works. Chapter 3 gives a formal statement of the problem that we attempt to resolve, and then present our method to extract translations from parallel corpus, involving generating translations by word alignment and filtering translation candidates using a dependency relation model. Chapter 4 describes the experimental settings and the data sets we utilize. In Chapter 5, we describe the evaluation results and present a further discussion. Finally, Chapter 6 gives the conclusion of this paper and points the future research direction.

2 RELATED WORK

Machine Translation (MT) has been an area of active research since 1950’s.. In the early years, rule-based approach is the state of the art for Machine Translation. Brown et al. (1993) propose a series of statistical models for improving MT performance and create a new approach called Statistical Machine Translation (SMT). Recently, much previous work have been done on phrase-based SMT (Marcu and Wong, 2002; Koehn et al. 2003; Koehn et al. 2004). While the traditional phrase-based SMT system which translates a paragraph of texts or a complete sentence, there are much previous work that consider translation of phrases, such as technical term translation (Dagan and Church, 1994), noun phrase translation (Cao and Li, 2002; Koehn and Knight, 2003), or bilingual collocation translation (Smadja et al. 1996). These sub sentential translation tasks are helpful for assisting human translators or machine translation. In our work, we focus on retrieving bilingual collocations, similarly to what has been done by Smadja and McKeown 1996.

Acquisition of bilingual collocation translation has been an active research topic recently. However, most previous work address translation of rigid collocation, such as technical terms and noun phrases (Kupiec, 1993; Ohmori and Higashida, 1999; Dagan and Church, 1994; Fung and Mckeown, 1997). The traditional SMT and previous works also focus on translating continuous words in a sentence. Translating non-continuous words, such as elastic collocations, might result in an unseen phrase in training corpus and generate improper translations. In contrast, we focus on translating elastic collocations, which have intervening words between the base word and the collocate, such as verb-noun collocations.

Many previous researchers have used bilingual dictionaries to generate collocation translations. Lü and Zhou (2004) utilize bilingual dictionaries to generate collocation translation candidates and build a collocation translation triple model based on dependency parser using the EM algorithm. However, using bilingual dictionaries as the translation source might be limited by the coverage of dictionaries. In contrast, our method uses parallel corpora as source to generate collocation translations, in an attempt to avoid the problem of limited coverage of bilingual dictionaries.

Recently, retrieving collocation translation from sentence-aligned parallel corpora is a popular approach. Smadja et al. (1996) propose a statistical method based on DICE coefficient to measure the correlation of a collocation and its translations from sentence-aligned parallel corpus. However, using only statistical information, such as DICE, to translate collocations may generate translations which are not collocations in the target language. Intuitively, the translation of collocation is also a collocation in target language. For instance, the verb-noun collocation should have a translation which is also a verb-noun collocation in the target language. Zhou et al. (2001) found that about 70% of the Chinese translations have the same relation type with the source English collocations. Seretan and Wehrli (2007) introduce a similar method to identify verb-object collocation translation in sentence-aligned parallel corpus, using a parser to ensure that the both syntactical relations of the source collocation and the target translation are the same. Finally, an optional semantic filter using a bilingual dictionary can be used to validate the semantic head of collocations. Our approach, utilize a dependency parser, similar to Seretan and Wehrli' s (2007) method but with different experiment settings, to ensure that the target language translation has the same relation type as the source collocation using an additional monolingual corpus of the target language. The main difference between our work and previous works is that we extract word translations from a parallel corpus based on the word alignment information. More specifically, our method is based on statistical machine translation model, not statistical association measures such as DICE.

In contrast to previous works, we present a model that generating collocation to assist ESL learners or bilingual dictionaries editors. The process of extracting word translation extraction is based on word alignment from parallel corpus. The translation candidates are filtered and ranked based on dependency relations, generated from a monolingual corpus using a target language dependency parser.

3 Method

Submitting a collocation to the SMT system directly might not receive a correct or fluent translation. The traditional SMT system typically translates continuous phrases. Unfortunately, elastic collocations, such as verb-noun collocations, which contain intervening words, may be unseen phrases in the training corpus of an SMT system. The SMT system might translate unseen phrases word by word, and generates inappropriate translations. To generate a proper translation for elastic collocation, an effective approach is to consider the

structure of collocations and various word forms.

3.1 Problem Statement

We focus on finding translation equivalents of verb-noun collocation in a parallel corpus. These translations then are ranked and returned as output. The returned translations can be examined by a human user directly, or passed to an SMT system to improve translation quality. Therefore, our goal is to return a set of ranked collocation translations. We now formally state the problem we are addressing.

Problem Statement: We are given a verb-noun collocation (V_c, N_c) and a word-aligned parallel corpus *PC*, and a phrase table *PT* from a SMT system (e.g., *Moses*). Our goal is to retrieve a set of combined translations of the base word and the collocate $CT_{combine} = \{(V_{t\_comb}, N_{t\_comb})_1, (V_{t\_comb}, N_{t\_comb})_2, \ldots, (V_{t\_comb}, N_{t\_comb})_m\}$ from *PT*, and another set of aligned collocation translations $CT_{align} = \{(V_{t\_align}, N_{t\_align})_1, (V_{t\_align}, N_{t\_align})_2, \ldots, (V_{t\_align}, N_{t\_align})_n\}$ from *PC*. These translations are finally ranked and returned as the system output.

In the rest of the paper, we describe the method for solving this problem in detail. First, we show the steps of extracting collocation translation from *PC* and building translation models (Section 3.2). Finally, we present how to generate collocation translations by these two models and ranks translation candidates at run time (Section 3.3).

3.2 Extracting Collocation Translation from Parallel Corpus

We attempt to find translations of verb-noun collocations from a parallel corpus, and filter translation candidates using a monolingual corpus. Our training process is showed in Figure 3.

(1) Generate word alignment from parallel corpus *PC*. (Section 3.2.1)
(2) Build the combination translation model. (Section 3.2.2)
(3) Build the alignment translation model from word alignment. (Section 3.2.3)

Figure 3. Outline of the training process

3.2.1 Generate word alignment from parallel corpus

In the first stage of the training process (Step (1) in Figure 3.), we generate word alignment data for each sentence pair in a parallel corpus using a alignment tool.

The input for this stage of training is a parallel corpus, as we will describe in Section 4.1. For each sentence pair in the parallel corpus, we use a word alignment tool to align a source word to the corresponding target words. The same procedure is performed in the inverse direction, from the target language to the source language. The output of this stage is the alignment information in both directions.

The alignment information in both directions is used to generate the phrase table (Section 3.2.2) and bidirectional alignment translation model (Section 3.2.3).

3.2.2 Build the combination translation model

In the second stage of the training process, we build a combination translation model based on a word translation model using a parallel corpus and a model based on a separate target language corpus.

The word translation model is used to generate translations of the base word and the collocate of a given collocation. To build this model, we need a phrase table *PT*, which is generated using an SMT tools, as our training data. A typical phrase table in the SMT system is usually contains the corresponding translation equivalents with direct and inverse translation probabilities for almost all the phrases up to a certain length in the training corpus. Figure 4 shows a sample part of the phrase table:

ePhrase	cPhrase	e_given_c	e_given_c_lexicol	c_given_e	c_given_e_lexicol	phrasePenalty
authority	管理局	5.78E-01	0.552336	0.404034	7.37E-04	2.72E+00
authority	事務 監督	0.755776	0.120515	3.12E-02	3.60E-05	2.72E+00
authority	管理局 共同	1	0.552336	6.54E-03	1.64E-04	2.72E+00
authority	監督	0.113352	0.221398	3.83E-02	3.87E-02	2.72E+00
authority	管局	0.329389	0.158094	1.14E-02	7.53E-03	2.72E+00
authority	權力	0.0603637	0.0872868	5.81E-02	3.72E-02	2.72E+00
authority	管理局 局長	0.515152	0.276991	5.79E-03	9.68E-04	2.72E+00
authority	權威	0.24152	0.191795	1.21E-02	6.73E-03	2.72E+00

Figure 4. An example of phrase table from English to Chinese

Take the phrase table in Figure 4 as an example, for each translation pair t_i in *PT*, the bidirectional translation probability $P_{i\_bidirect}$ is calculated:

$$P_{i\_bidirect} = \log(P_{i\_inverse}) + \log(P_{i\_direct})$$

where $P_{i\_inverse}$ (*e_given_c* in Figure 4.) is the inverse translation probability and P_{i_direct} (*c_given_e* in Figure 4.) is the direct translation probability. Then we build the word translation model, which consists of translation pairs and corresponding bidirectional translation probabilities. Table 1. shows an example of the word translation model.

Table 1. An example of the word translation model

Word	Translation	Bidirectional Probability
authority	管理局	-1.454484
	事務 監督	-3.747178
	管理局 共同	-5.029688
	監督	-5.439792
	管局	-5.586563
	權力	-5.652645
	管理局 局長	-5.814680
	權威	-5.833058

The target language model is also required to filter out inappropriate collocation translations. We build this model based on a target language monolingual corpus.

In the first step of the procedure, we parse a monolingual corpus of the target language using a dependency parser to generate *RelationPairs*. For each relation pair in *RelationPairs*, we only count the frequency of the verb-noun relation pairs <w1, w2, VN>, since we aim at

translating verb-noun collocation. Next, we generate the target language model *VNPairsFrequency*, consisting of the frequency of each verb-noun relation pair. The combination translation model is then generated by combining the word translation model and the target language model. We will describe the run time process of combination translation in Section 3.3.

3.2.3 Build the bidirectional alignment translation model

In the third and final stage of training, we address building a bidirectional alignment translation model from word alignment for translating collocations. The input to this stage is the alignment information of both directions *Align_StoT* and *Align_TtoS*, generated in the previous section (Section 3.2.1). The algorithm is shown in Figure 5.

procedure BuildBidirectionalModel(*PC, Align_StoT, Align_TtoS*)
(1) *LemmatizePC* = Lemmatize(*PC*)
for each $src\_sentence_i$, $tgt\_sentence_i$ in *fulfill their functions*
for each src_word_j in $src\_sentece_i$
(2) *SrcTrans* [*j*] = FindIntersection(src_word_j, *Align_StoT, Align_TtoS*)
for each src_word_j in $src\_sentece_i$
(3a) SkipBigramList = GenerateSkipBgram1toN (src_word_j, src_word_{j+N})
(3b) TransList = TranslateSkipBigrams(SkipBigramList, SrcTrans)
(4) BidirectionalTransFreq = CountFreq (TransList)
(5) Return BidirectionalTransFreq

Figure 5. The algorithm of building bidirectional alignment translation model

In Step (1) of the algorithm, we first lemmatize all source sentences to generate the lemmatized parallel corpus *LemmatizePC*.

In Step (2) of the algorithm, we extract translations for each source word in each sentence pair. We first find target words aligned to the source word by the source to target alignment information. For each aligned target word, the target to source alignment information is then used to determine whether the source word is also aligned to this target word. We choose the target word as the translation of the source word if the source word is also aligned to it. The translations of each source word *SrcTrans* are generated.

In Step (3a), source skip bigrams are generated for each source sentence. A skip bigram is combined by the head word and the tail word of a phrase. In order to limit the amount of the data processed, we only consider phrases with the distance 1 to 4 words in generating skip bigram. Then, in Step (3b), we retrieve the corresponding translations for each skip bigram.

In Step (4), we count the frequency of each skip bigram translation pair. Since we focus on translating verb-noun collocation, we only deal with verb-noun bigram and translation pairs to reduce processing time. Table 1 shows examples related to the skip bigram "*play role*"

Finally, in Step (5), the frequency of skip bigram and translation pairs, *BidirectionalTransFreq*, is returned. Table 2 shows an example output of this stage.

Table 2. An example of bidirectional alignment translation model

Collocation	Translation	Frequency
play role	發揮 作用	1193
	扮演 角色	612
	擔當 角色	475
	擔任 角色	46
	發揮 功能	37

3.3 The Run Time Process

Once all collocation translation models are obtained, these models are combined and used to translate collocations. For a given collocation, we generate and evaluate translations using the procedure shown in Figure 6. In the following, we first present the translating process of the combination translation model, and then the bidirectional alignment model. Finally, we describe the ranking algorithm to output the collocation translations.

```
procedure    TranslatingCollocation ( C, CombTM, BiTM )
(1a) Base, Collocate = DecomposeCollocation(C)
(1b) BaseTransList = GenerateCombBaseTranslation(Base)
(1c) CollocateTransList = GenerateCombCollocateTranslation(Collocate)
(1d) CombTransList = ∅
       for each bTrans in BaseTransList,
            for each cTrans in CollocateTransList
(2a)             Score = CalculateCombTM_Score(cTrans, bTrans)
(2b)             CombTrans = (bTrans, cTrans)
(2c)             CombTransList += (CombTrans, Score)
(3)    Sort CombTransList in decreasing order
(4a) BiTransList = GenerateListOfBiTransWithScore( C )
(4b) Sort BiTransList in decreasing
(5)    RankedCandidates = Rank( CombTransList, BiTransList)
(6)    Return top N RankedCandidates
```

Figure 6. Generation and Ranking Procedure at run time

3.3.1 Combination Translation Model

In Step (1a), we first decompose the given collocation into the base word and the collocate. Consider the collocation "*delegate authority*" for example, "*authority*" is the base word and "*delegate*" is the collocate. A set of the base word translations is generated as *BaseTransList*, and translation list for the collocate *CollocateTransList* is also generated. We then generate possible collocation translations *CombTransList* using Cartesian product of each *bTrans* in *BaseTransList* and each *cTrans* in *CombTransList*. Each *CombTrans* (*bTrans*, cTrans) in *CombTransList* gets a word translation model score using the following formula:

$$Score_{WTM}(CombTrans) = Score_{WTM}(bTrans) + Score_{WTM}(cTrans)$$

and a target language model score as follows :

$$Score_{TLM}\ (CombTrans) = log\ (\lambda_s * P(CombTrans) + (1 - \lambda_s) * P(bTrans) * P(bTrans) * P(VN\_Collocation|AllCollocations))$$

where $\lambda_s = \frac{1}{1+Freq(CombTrans)}$ is the smoothing weight to cope with the data sparse problem, $0 \leq \lambda \leq 1$. We combine the $Score_{\mathrm{WTM}}$ and $Score_{TLM}$ by a weighting formula:

$$Score_{CombTM} = \lambda * Score_{WTM} + (1 - \lambda) * Score_{TLM}$$

where λ is the model weight and $0 \leq \lambda \leq 1$.

We retrieve the translations and rank translations in descending order of $Score_{CombTM}$. The N top-ranked translations of combination translation model are produced.

3.3.2 Bidirectional Alignment Model

In step (4a), we generate another set of translations using the bidirectional alignment model for the given collocation *C*. Translations of *C* are retrieved from the bidirectional alignment model, and each translation is scored as follows:

$$Score_{BiModel} = \frac{P(BiTrans)}{P(C)}$$

The generated translations are ranked in descending order of $Score_{BiModel}$, and N top-ranked translations of bidirectional alignment model are retrieved.

Once all translations of two models are generated, we merge the N top-ranked translations of two models and re-rank them. The ranking algorithm we use aims at retrieving the translations that two models have in common. The score of the top N translation of each model is re-calculated as the formula:

$$TransScore_N = \frac{1}{N}$$

where N means the output rank of a translation in a model. We then merge all translations, and if there is a translation that both in output of two models, we add two scores together. Finally, the merged translations are ranked with their merged score (Step 5), and the K top-ranked translations are returned as the final result produced by our method.

4 Experimental Setting and Results

We have proposed a new method to retrieve translations for a given collocation from parallel corpus that are likely to help ESL learners or bilingual collocation dictionary editors. As such, our method is trained and evaluated on top of word alignment information of parallel corpus and an additional monolingual corpus. Furthermore, since the goal of our model was to

retrieve a set of good translations to assist bilingual collocation dictionary editors, we evaluated our method on a group of English collocations, which are selected from an English collocation dictionary. Finally, since we do not have reference answers for such translation advising task, we will use human judges to evaluate the quality of our generated collocation translations.

In this chapter, we first present the details of training our system for the evaluation (Section 4.1). Then, Section 4.2 describes the alternative methods that we used in our comparison. Section 4.3 introduces the datasets used in our experiments and the evaluation metrics for evaluating the performance of our system, and Section 4.4 describe the tuning process of our system module. Section 4.5 reports the results of our experiment evaluations. Finally, in Section 4.6, we analyze the experimental results in detail.

4.1 Experimental Settings

In our bidirectional alignment translation model, we used the Hong Kong Parallel Text (HKPT; LDC2004T08) as the training data, which contains approximately 222,000 sentence pairs,. English sentences of HKPT were lower-cased and performed lemmatization using Nature Language Toolkit (NLTK), a suite of open source modules written in Python. Chinese sentences of HKPT were word-segmented by the CKIP Chinese word segmentation system (Ma and Chen, 2003). To obtain word alignment information of English and Chinese sentences, we used GIZA++ (Och and Ney, 2003) as the word alignment tool.

For word translation model of our combination translation model, the phrase table of HKPT was built by the state-of-the-art phrase-based SMT system, Moses (Koehn et al., 2007). Common settings are used to run Moses: GIZA++ was used for word alignment, grow-diagonal-final (Koehn et al., 2005) heuristics were used to combine bi-direction word alignment, and extract bilingual phrase (Koehn et al., 2005).

For the target language model of our combination translation model, we used Central News Agency (CNA) as the monolingual corpus, by using the CKIP Chinese Parser to produce dependency relations.

Our system uses some parameters during training. The parameters were tested with different values and finally the values were set as shown in Table 3. We did not test these values exhaustively and further tuning may improve the performance of our system.

Table 3. Parameter used in training

Parameter	**Value**	**Description**
minBidiretionProb	**-15.0**	**Minimum bidirectional translation probability of the base word and the collocate translation.**
numWordTrans	**100**	**Number of the base word and the collocate translations used to generate collocation translations.**

4.2 Methods Compared

Recall that our method starts with an English verb-noun collocation given by a user, and find

the Chinese translations of the collocation. The output of our system is a list of ranked translation candidates, which can either be shown to the user directly, or incorporated into the existing SMT systems.

In this paper, we have introduced a hybrid method for generating collocation translations from a parallel corpus and an additional target language monolingual corpus for a given collocation. Therefore, we compare the results of different translation retrieval methods from a parallel corpus.

We compared different methods for retrieving collocation translations from a parallel corpus, which are listed as follows:

— **MOSES**: The state-of-the-art SMT framework that are widely used recently. We build the Moses translating system using the same HKPT parallel corpus with default setting as our baseline system.
— **Combination Translation Model (CTM)**: The system based on translating the base word and the collocate separately and then combined them to generate candidates. The candidates are filtered by the target language model as output.
— **Bidirectional Alignment Translation Model (BTM)**: The system extracts translation based on the bidirectional alignment information of a word-aligned parallel corpus using GIZA++.
— **Hybrid Translation Model (HYBRID)**: Our system based on both CTM and BTM by combining the results of each model with the translation ranking scheme as described in Section 3.3.

4.3 Evaluation Data Sets and Metrics

The evaluation of the traditional SMT systems usually base on the quality of translated texts. Bilingual Evaluation Understudy (*BLEU*; Papineni et al, 2002) is a mainstream automatically scheme to evaluate quality of the MT translations. The translation of the input texts is compared the similarity with human-translated reference answers. However, since our system aims at assisting user to find appropriate translations for bilingual collocation dictionaries editors, the lack of reference translations results in a difficult situation of translation equivalents.

To evaluate our system, we randomly selected 55 English verb-noun collocations from the Oxford Collocations Dictionary (OCD; Oxford University Press, 2009), which collects about 25,000 common collocations. All nouns of collocations were chose from Academic Word List (*AWL*; Coxhead 2003). The testing data consisted of 80 collocations, which were selected in the same way.

We used two human judges to examine the generated translations for each collocation in the data sets for evaluation. The human judges were asked to examine retrieved collocation translation one at a time, and judge each translation candidate as “correct”, “partial acceptable”, or “unacceptable” for the collocations.

By using the judgments from two human judges, we evaluate the translations using the *Top-N accuracy*, and Mean Reciprocal Rank (MRR) metrics that describes in the next.

Definition 4.1. The *Top-N accuracy* of a translation model for *K* collocations in test data, in our definition, is the percentage of all collocations with translation results, where Top-N translations contain a correct translation.

Example 4.1. Consider top 3 translations returned by the system for 10 collocations in test data. If there are 3 collocations with correct translations at first place, 2 at second place, and 1 at third place, the Top-N accuracy of this system is (3+2+1)/10 = 60%.

We also compute Mean Reciprocal Rank (MRR), a measure of how much effort needed for a user to find a compatible translation in the returned order of collocation translations (Voorhees and Tice, 1999). The MRR value is a real number between 0 and 1, where 1 denotes the compatible translations always occur at first place. We report the MRR results to examine the effectiveness of our system being used to assist bilingual dictionaries editors.

Definition 4.2. The Reciprocal Rank for a system, for a input collocation *c* from the data set *D,* is defined as R_c^{-1}, where R_c is the first rank of a translation judged as a correct translation for *c*. The Mean Reciprocal Rank (MRR) of the system is the average of the Reciprocal Rank values over all evaluated collocations in *D*.

Example 4.2. Consider a collocation *c* and the system outputs 5 translations for *c*. If three translations are judged correct and ranked at 2, 3, 5. The *Reciprocal Rank* for *c* is 2^{-1} = 0.5.

We also calculate *Kappa statistics* (Cohen, 1960) to evaluate the agreement between two human judges. Cohen's Kappa coefficient κ is a statistical measure of the inter-judge agreement, which consider the agreement occurring by chance and the agreement of observed judgment result. If the judges are in complete agreement with each other for the classification totally, then $\kappa = 1$. If there is no agreement between the judges, then $\kappa \leq 0$.

Definition 4.3. The Cohen's Kappa coefficient κ is calculated as the equation:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$$

where *Pr*(*a*) is relative observed agreement between judges, and *Pr*(*e*) is the hypothetical probability of agreement by chance, which is calculated by using the observed judgments by each human judge.

4.4 Tuning Parameters

In this section, we describe the process of tuning the parameter λ (weight of word translation model in the combination translation model (*CTM*)) by using the development data. Recall that the score of *CTM* is calculated as the following:

$$Score_{CombTM} = \lambda * \text{Score}_{\text{WTM}} + (1 - \lambda) * \text{Score}_{TLM}$$

The different weights of λ determine whether the word translation model (*WTM*) or the target language model (*TLM*) has more influence on the collocation translations score $Score_{CombTM}$. A higher value of λ means that $Score_{CombTM}$ relies more on *WTM* than *TLM*. In contrast, *TLM* has more influence for a lower λ.

To select a suitable weight λ, we choose a set values in the division between 0 and 1 to

find the best *MRR* values by using development data. As the result. We make $\lambda = 0.4$ as our model weight.

4.5 Evaluation Results

In this section, we evaluate the performance of various systems in Section 4.2 using the testing data set and different metrics we described in Section 4.3.

For each compared system, we generated top 3 ranked translations for each collocation in the testing data. Samples of the system output for collocations in the testing data are listed in Appendix A. We first calculate the Kappa value to acquire the agreement between two judges. In order to calculate the Kappa value, we mixed all top 3 translations from various systems and generated a translation pool, which contains all generated translations from different systems for each collocation in test data. The human judges then evaluated on all 1451 translations in the translation pool, and we got the Kappa value $\kappa = 0.61$, which indicates that the human judges have substantial agreement while judging translation results

Table 4. Top-N precision of different systems

	Top-1	***Top-2***	***Top-3***	***Top-4***	***Top-5***
Moses (baseline)	.55	.73	.77	.78	.82
BTM	.49	.56	.59	.59	.60
CTM	**.67**	.75	.80	.82	.83
Hybrid (CTM+BTM)	.65	**.81**	**.85**	**.88**	**.89**

Table 5. MRR value for all translations for collocations in test data

by seeing “correct

System	***MRR***
Moses (baseline)	.72
BTM	.55
CTM	.76
Hybrid (CTM+BTM)	**.78**

We report the top-N accuracies from top-1 to top-5 in Table 4. The results indicate that, except the top-1 accuracy, our **Hybrid** method has significantly better accuracy improvement than other three methods from top-2 to top-5. Compared with the baseline, our system improves 7% ~ 10% more accuracies. **Hybrid**, combined **CTM** and **BTM**, improves about more 6% accuracy than only **CTM**. This result indicates that although top-N accuracies of **BTM** is the lowest since it suffers from low translation coverage, **BTM** still improves

Table 4 reports the *MRR* value for all compared methods. The reported *MRR* is an average value of the judgment by two judges. **Hybrid** has the best *MRR* 0.78 of all methods , which means that a correct answer can be found at the first 2 translations in ranked translation list by a human user. Also our **HYBRID** method, compared to the traditional SMT system **MOSES**, improves 0.06 *MRR* score.

5 Conclusion and Future Work

In this paper, we have introduced a new method for translating verb-noun collocations by using a parallel corpus and an additional monolingual corpus. The generated collocation translations can be used to assist ESL learners and bilingual collocation dictionaries editors with the choice of proper translations. Our method is based on a parallel corpus to extract collocation translations, and a monolingual corpus of the target language to filter out inappropriate translations. Evaluations of our experiments have shown that our method produce better translations for a given collocation than the traditional SMT system.

Many avenues exist for future research and improvement of our system. For example, we could extend the parallel corpus by using more general corpora to increase the quality of collocation translations. The ranking algorithm to combine and rank outputs of two models could be used a better existing algorithm. Also, dealing with different types of collocation, such as Adjective-Noun and Phrasal Verb-Noun, could be considered to translate more collocations in our system. Additionally, an interesting direction to explore is to use more semantic information to improve translations. If example sentences of a collocation are available, we could use the word sense disambiguation technique to help us choose a precise translation.

References

[1] Cao Y. and Li H. 2002. Base noun phrase translation using web data and the EM algorithm. In Proceedings of the 19th international conference on computational linguistics-volume 1, Association for Computational Linguistics. 1 p.

[2] Cohen J. 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1):37-46.

[3] Dagan I. and Church K. 1994. Termight: Identifying and translating technical terminology. In Proceedings of the fourth conference on applied natural language processing, Association for Computational Linguistics. 34 p.

[4] Fung P and McKeown K. 1997. A technical word-and term-translation aid using noisy parallel corpora across language groups. Machine Translation 12(1):53-87.

[5] Koehn P. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. Machine Translation: From Real Users to Research :115-24.

[6] Koehn P. and Knight K. 2003. Feature-rich statistical translation of noun phrases. Proceedings of the 41st annual meeting on association for computational linguistics-volume 1Association for Computational Linguistics. 311 p.

[7] Koehn P., Och F. J. and Marcu D. 2003. 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 1Association for Computational Linguistics. 48 p.

[8] Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C. and Zens R. 2007. Moses: Open source toolkit for statistical machine translation. Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessionsAssociation for Computational Linguistics. 177 p.

[9] Kupiec J. 1993. An algorithm for finding noun phrase correspondences in bilingual

corpora. Proceedings of the 31st annual meeting on association for computational linguisticsAssociation for Computational Linguistics. 17 p.

[10] Loper E. and Bird S. 2002. NLTK: The natural language toolkit. Proceedings of the ACL-02 workshop on effective tools and methodologies for teaching natural language processing and computational linguistics-volume 1Association for Computational Linguistics. 63 p.

[11] Lü Y. and Zhou M. 2004. Collocation translation acquisition using monolingual corpora. Proceedings of the 42nd annual meeting on association for computational linguisticsAssociation for Computational Linguistics. 167 p.

[12] Marcu D. and Wong W. 2002. A phrase-based, joint probability model for statistical machine translation. In Proceedings of the ACL-02 conference on empirical methods in natural language processing-volume 10. Association for Computational Linguistics. 133 p.

[13] Ohmori K. and Higashida M. 1999. Extracting bilingual collocations from non-aligned parallel corpora. In Proeeding. of the 8th international conference on theoretical and methodological issues in machine translation (TMI99). Citeseer. 88 p.

[14] Oxford University Press. 2009. Oxford collocations dictionary 2nd . USA: Oxford University Press.

[15] Papineni K., Roukos S., Ward T. and Zhu W. J. 2002. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics. 311 p.

[16] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. Biometrics, 33(1):159-174. International Biometric Society

[17] Seretan V. and Wehrli E. 2007. Collocation translation based on sentence alignment and parsing. Actes de la 14e conférence sur le traitement automatique des langues naturelles (TALN 2007). Citeseer. 401 p.

[18] Smadja F, McKeown KR, Hatzivassiloglou V. 1996. Translating collocations for bilingual lexicons: A statistical approach. Computational Linguistics 22(1):1-38.

[19] Zhou M, Ding Y, Huang C. 2001. Improving translation selection with a new translation model trained by independent monolingual corpora. Computational Linguistics and Chinese Language Processing 6(1):1-26.