利用依存關係之辭彙翻譯

Word Translation Disambiguation via Dependency

Meng-Chin Hsiao[1], Kun-Ju Yang[2], and Jason S. Chang[2]
a871002@nthu.us;shanks22@gmail.com;Jason.jschang@gmail.com

[1] Institute of Information and Systems and Applications, National Tsing Hua University
[2] Department of Computer Science, National Tsing Hua University

摘要

本論文提出了一個利用依存關係解決詞彙翻譯的新方法。我們的方法包含了訓練階段及測試階段。在訓練階段，取得與實詞具依存關係的搭配字，並在這些依存關係的條件下，學習分辨翻譯歧義的決策表(decision list)。在測試階段，對於句子中每個實詞檢查跟其有依存關係的搭配字。在測試階段，比對決策表，給予這些字一個正確翻譯。我們實際撰寫了程式，並利用香港新聞及香港立法會議記錄作爲訓練資料。在實驗中我們用了五種不同的方法去處理測試資料並透過一個自動的擬似 BLEU 的評估方法去比較實驗結果。由實驗結果顯示，依存關係的確可以顯著的幫助詞彙翻譯，而實驗也證實某些依存關係是比其他的依存關係更具影響力的。

Abstract

We introduce a new method for automatically disambiguation of word translations by using dependency relationships. In our approach, we learn the relationships between translations and dependency relationships from a parallel corpus. The method consists of a training stage and a runtime stage. During the training stage, the system automatically learns a translation decision list based on source sentences and its dependency relationships. At runtime, for each content word in the given sentence, we give a most appropriate Chinese translation relevant to the context of the given sentence according to the decision list. We also describe the implementation of the proposed method using bilingual Hong Kong news and Hong Kong Hansard corpus. In the experiment, we use five different ways to translate content words in the test data and evaluate the results based an automatic BLEU-like evaluation methodology. Experimental results indicate that dependency relations can obviously help us to disambiguate word translations and some kinds of dependency are more effective than others.

關鍵詞 ： 翻譯選擇，統計式機器翻譯，平行語料庫，決策表，依存關係

Keyword: translation selection, statistical machine translation, parallel corpus, decision list, dependency.

1. Introduction

English is the major language in today’s world; for this reason, the latest knowledge and information is mostly written in English. People who want to get new information have to be good at reading English. Although non-native speakers of English can consult a dictionary to understand the meanings of a word, it is still difficult to find the suitable translation of m-context meanings of the words in the specific sentence. Hence, there are more and more machine translation systems on the web to help people overcome the language barrier. For example, BABEL FISH(http://babelfish.yahoo.com/translate_txt) and Google Translate (http://google.com/tra nslate_t) are two representative machine translation services on the web.

The traditional machine translation systems mostly translate by word or phrase. However, such a word (phrase)-based approach may lead to problems for not considering the structure of the sentence. Consider the word *"motion"* in the given sentence. When the sentence containing it was submitted to BABEL FISH (Figure 1) for translation, the incorrect answer *"行動"* is returned. To improve the kind of limitation seen in BABEL FISH, many researchers consider cross-language phrasal information in statistical machine translation (SMT). At present, some machine translation systems (e.g., Google Translate) have been developed based on the idea to improve performance. However, we submit whole sentence containing "*motion*" and "*passed*" to Google Translate (Figure 2), we still cannot get the suitable translation like "*通過*" of the word "*passed*", especially when "*motion*" and "*passed*" are for apart. Sometimes, words are not translated at all. To obtain the proper translation of the words in a sentence, a promising approach is to consider the syntactic information of the sentence, and to use them for improving the performance of word translation disambiguation (WTD).

Figure 1. Submitting text containing "motion" and "passed"
to BABEL FISH for translation in Chinese

We present a new method that automatically determines the translation of given words in the sentence by considering dependency relationships between the words in a sentence. Dependency information includes structure information and dependency can be established between two words that are far apart in the sentence. For example, consider the following sentence "I move that the motion on "*Education on media literacy" as set out on the Agenda be passed.*", "*motion*" and "*passed*" has a dependency of subject-complement. Intuitively, by conditioning probability of translations of "*motion*" and "*passed*" on the dependency pair, nsubjpass (passed-20, motion-5), we can find the correct translation of the words for the context.

The rest of the paper is organized as follows. We review the related work in the next section. Then we present our method in details for automatically training a word translation disambiguation system (Section 3).

Afterward we compare the quality of results between the proposed model and other models (Section 4). Finally, we discuss the results, make conclusion, and close with future work.

Figure 2. Submitting text containing *"motion"* and *"passed"* to Google Translate for translation in Chinese

2. Related Work

Word translation disambiguation has been an important problem in natural language processing. This problem is related to the WSD tasks and is one of the difficult issues in machine translation. In our work, we focus on finding the translations of each content word in the given sentence. The contexts would be English and the target words will be their translations in a second language (e.g., consider the word "*motion*" can be translated as "*行動*" or "*會議*" depending on the sentential content).

Dagan, Itai, and Ulrike (1994) presented an approach for resolving lexical ambiguities in one language using a statistical data on lexical relationship in another language. Yarowsky (1994) showed that decision list (Rivest, 1987) is a good way to model the relation between the words and their translations. We also use the decision list in our approach for estimating translation probability of the word. Yarowsky (1995) exploited two powerful properties that one sense per collocation and one sense per discourse for WSD. He also presented a bootstrapping approach for word sense disambiguation. We also exploit one sense per dependency relationship in our approach.

Pedersen (2000) presented a corpus-based approach to word sense disambiguation that builds an ensemble of Naive Bayesian classifiers, each of which is based on lexical features that represent co-occurring words in varying sized windows of context. Koehn and Knight (2000) present a novel approach to the WTD problem that can be trained using only unrelated monolingual corpora and a lexicon to estimate word translation probabilities using the EM algorithm. Zhou, Ding, and Huang (2001) also proposed an approach to training the translation model by using unrelated monolingual corpora. They parsed a Chinese corpus and an English corpus with dependency parsers, and two dependency triple databases are generated. Then, the similarity between a Chinese word and an English word can be estimated using the two monolingual dependency triple databases with the help of a simple Chinese-English dictionary. Their translation model overcomes the long distance dependence problem to some extent. Their model can be used translate Chinese collocations into English. In our approach, we only parse the English sentences in a parallel corpus with a dependency parser and try to translate English in to Chinese.

Li and Li (2002, 2004) considered bilingual bootstrapping as an extension of Yarowsky's approach. When the task is word translation disambiguation between two languages, they used the asymmetric relationship between the ambiguous words in the two languages to significantly increase the performance of bootstrapping. They have developed a method for implementing this bootstrapping approach that combines the use of naive Bayes and the EM algorithm. Ng, Wang, and Chan (2003) considered WSD when manually sense-tagged data is not available for supervised learning. They evaluated an approach to automatically acquire sense-tagged training data from English-Chinese parallel corpora. Pham, Ng, and Lee (2005) have investigated the use of unlabeled training data for WSD, in the framework of semi-supervised learning. Empirical results show that unlabeled data can bring significant improvement in WSD accuracy. We used a bilingual corpus but we do not require sense annotation of the data, because we rely on word alignment tool to annotate translation information of the words in the source sentences.

In a study more closely related to our work, Carpuat and Wu (2005) purposed a state-of-the-art Chinese word sense disambiguation model to choose translation candidates for a typical IBM statistical MT system. However, they did not obtain significantly better translation quality than using statistical machine translation system alone. But Cabezas and Resnik (2005) purposed using target language vocabulary directly as "sense," leading to small improvement in translation performance over a state of the art phrase-based statistical MT system. In previous work, human judgment is required for evaluation of sample word tasks of WSD or WTD. In our research, our goal is to study all-word task of WTD and we propose an automatic evaluation methodology.

3. Word Translation Disambiguation Via Dependencies

Finding the appropriate translation of content words in a given sentence is important for machine translation as well as computer assisted language learning. State of the art phrase-based statistical MT systems do a good job if the word providing the "hint" is nearby. Unfortunately, a phrase-based MT system may fail to use of the word that is in a distant from the word we want to translate. To translate the words of the sentence, a promising alternative approach is to find the likely translation of each word through statistical analysis of its dependencies.

3.1 Problem Statement

We focus on a subtask of MT system; that is we focus on finding the appropriate translation of content words via dependencies. These dependencies provide recursive syntax structure information of the words in the sentence. We collect these dependencies and the relevant translations in a parallel corpus and find out the relationship between them. The goal is to find the proper translation of content words in the given sentence. Formal statement of the problem is as follows.

Problem Statement: We are given an English sentence *S* (e.g., "*A very big apple on the table was eaten by him.*") that we want to translate. Our goal is to give each content word, $w_1, w_2, \ldots, w_m$, in *S* a most appropriate Chinese translation relevant to the context of *S*. For this, we derive dependencies (e.g., advmod (big-3, very-2), amod(apple-4, big-3), nsubjpass(eaten-9, apple-4), etc.), $d_1, \ldots, d_p$, in *S*, then use the dependencies of the word *w* (i.e., dependency relationship (*w*, *w'*) or dependency relationship(*w'*, *w*)) to find the most appropriate translation for *w*.

In the rest of this section, we describe our solution to this problem. First, we define a dependency-based translation model for word translation disambiguation (Section 3.2). This training strategy relies on a set of dependency relationships derived from a dependency relationships collection. In this section, we also describe the other two strategies that we use when no dependency information is available. Finally, we show how our method handles a given sentence at run time by using a decision list (Section 3.3).

3.2 Training the Dependency-Based Translation Model

We take advantage of a word-aligned parallel corpus as training data to establish a decision list for word translations based on dependency relationships. For each word in a sentence, we obtain the translation and dependency relationships using word alignment tool (e.g., Giza++) and a general purpose parser (e.g., Stanford parser). With that information, we compute the word translation probability for all dependency relationships based on logarithmic likelihood ratio (LogL):

$$LogL = Log(\frac{P(t \mid w_t, d, w_d)}{P(\bar{t} \mid w_{t,} d, w_d)})$$

$$= Log(\frac{\frac{count(t, w_t, d, w_d))}{count(w_t, d, w_d)}}{\frac{count(\bar{t}, w_t, d, w_d)}{count(w_t, d, w_d)}}) = Log(\frac{count(t, w_t, d, w_d)}{count(\bar{t}, w_t, d, w_d)})$$

(1) Parse the source language using a dependency parser (Section 3.2.1)
(2) Use an alignment tool to align words in a parallel corpus (Section 3.2.2)
(3) Compute the decision list for translation and dependency (Section 3.2.3)
(4) Compute the probability of a translation for each word **(**Section 3.2.4)

Figure 3. Outline of the process used to train in our method

3.2.1 Parse the source language using a dependency parser

In the first stage of the training process (Step (1) in Figure 3), we use the English part of an English-Chinese parallel corpus as the input data. First, we utilize a tagger to tokenize the sentences, give each word in the source sentences a part of speech (POS) tag, and obtain dependency relationships from the source sentences via a dependency parser. We use an English sentence as an example to show the process. (Figure 4).

3.2.2 Use an alignment tool to align words in a parallel corpus

In the second stage of the training process (Step (2) in Figure 3), we use a word alignment tool to align words in a parallel corpus. First, we lemmatize the tokens obtained from the first stage. Words that are tagged proper noun are not lemmatized. Then, target language sentences are segmented using a word segmentation tool. Finally, each pair of source and target sentence is word-aligned using an existing word alignment model to produce word alignment information. Figure 5 shows an example of the process.

E: I move that the motion under the Interpretation and General Clauses Ordinance as set out in the Agenda be passed.

tokenizing, POS tagging, dependency parsing

E: I/PRP move/VBP that/IN the/DT motion/NN under/IN the/DT Interpretation/NNP and/CC General/NNP Clauses/NNP Ordinance/NNP as/IN set/VBN out/RP in/IN the/DT Agenda/NNP be/VB passed/VBN ./.

nsubj(move-2, I-1)	complm(passed-20, that-3)
det(motion-5, the-4)	nsubjpass(passed-20, motion-5)
prep(motion-5, under-6)	det(Interpretation-8, the-7)
pobj(under-6, Interpretation-8)	cc(Interpretation-8, and-9)
nn(Ordinance-12, General-10)	nn(Ordinance-12, Clauses-11)
conj(Interpretation-8, Ordinance-12)	mark(set-14, as-13)
dep(motion-5, set-14)	prt(set-14, out-15)

Figure 4. An example to show the result of the tagger and dependency parser

3.2.3 Compute the decision list for each translation and dependency

After deriving the translations and dependencies, we are in a position to train a classifier for WTD. We use the dependency relationship to condition the translation probability, and then we compute a score for each translation conditioned on one of the relevant dependency relationships. There are many different approaches to do this for various pattern recognition problems. We choose the decision list for simplicity and efficiency considerations. The algorithm we used is similar to the approach proposed in Yarowsky (1994) for WSD. For each possible translation of a given word, we compute the logarithmic likelihood ratio (LogL)

$$LogL = Log\left(\frac{count(t, w_t, d, w_d)}{count(\bar{t}, w_t, d, w_d)}\right)$$

where t is the translation of the word w_t with dependency d with another word w_d and *count* (t, w_t, d, w_d) is the number of instance of word w_t aligned with the translation t under of dependency relationship d (w_t, w_d) d, and count ($\bar{t}$, w_t, d, w_d) is the number of instance of word w_t aligned with the other translations $\bar{t}$ under the same relationship.[1]

Sample output is shown in Table 1. The LogL in Table 1 are computed by count (t, w_t, d, w_d) and count ($\bar{t}$,

[1] Here d (w_t, w_d) and d (w_d, w_t) are treated as different dependency relationships.

w_t, d, w_d). In the experiment described in Chapter4, we smooth count (t, w_t, d, w_d) and count ($\bar{t}$, w_t, d, w_d) by held out data.

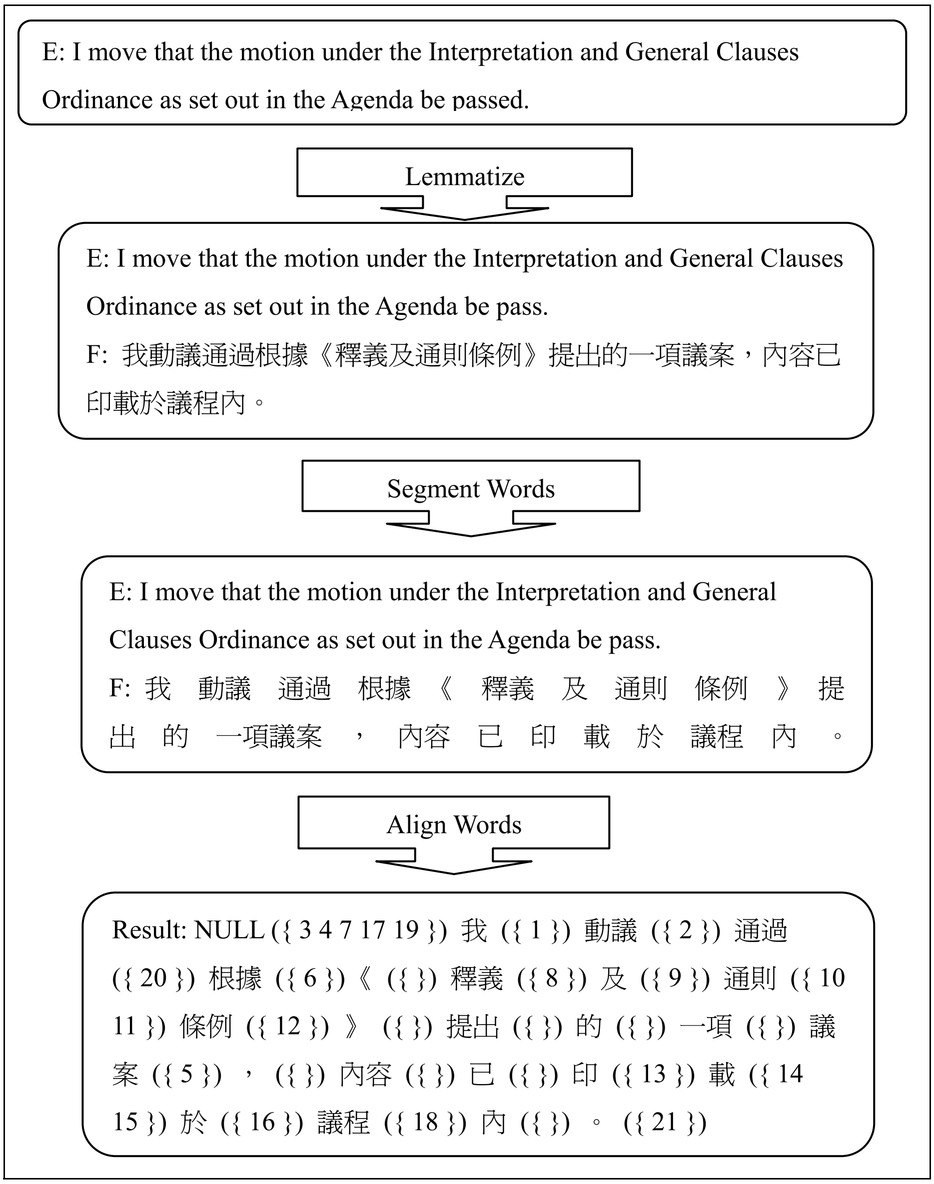


Figure 5. An example to show the data handling after word alignment

Table 1. Calculating LOGL with N = count (t, w_t, d, w_d), N'= count ($\bar{t}$, w_t, d, w_d), $\bar{t} \neq t$

dep	w_d	w_t	t	LogL	N	N'
nsubjpass	pass	motion	議案	1.608	294	58.8798
nsubjpass	pass	motion	動議	-1.975	43	309.8798
nsubjpass	pass	motion	…	…	…	…

3.2.4 Parse the source language using a dependency parser

In the last stage of the training process (Step (4) in Figure 3), we compute two types of word translation probability: P (t | w) and P (t | w, p). For unseen dependency relationships, we use P (t | w, p) to predict translation and for unseen word/POS combination, we use P (t | w) to predict translation for all POS's.

The word translation probability is calculated based on sentences where the source and target words are aligned.

We compute the word translation probability for all word aligned with a translation by the ratio of two counts:

$$P(t \mid w) = \frac{count(w,t)}{count(w)}$$

where *count* (*w*, *t*) is the number of instance of word *w* aligned with some translation, and *count* (*w*) is the number of w instances.

Table 2. An example to calculate P (t | w) for "plant"

w	t	Count
plant	核電廠	342
plant	植物	258
plant	種植	167
plant	…	…
P(核電廠|plant)=342/2172, P(植物|plant)=258/2172		

Table 3 Examples of P (t | w, p) for the noun "plant"

w	t	Count
plant	核電廠	341
plant	植物	245
plant	發電廠	132
plant	…	…
P(核電廠|plant, Noun)=341/1852		

We can then condition the translation probability using the POS information obtained from the first stage. Table 3 shows an example for the word "*plant*" that is tagged noun.

$$P(t \mid w, p) = \frac{count(w,t,p)}{count(w,p)}$$

where *count* (*w*, *t*, *p*) is the number of instances of word w with the POS *p* aligned with some translation, and *count* (*w*, *p*) is the number of *w* with the POS *p* instances.

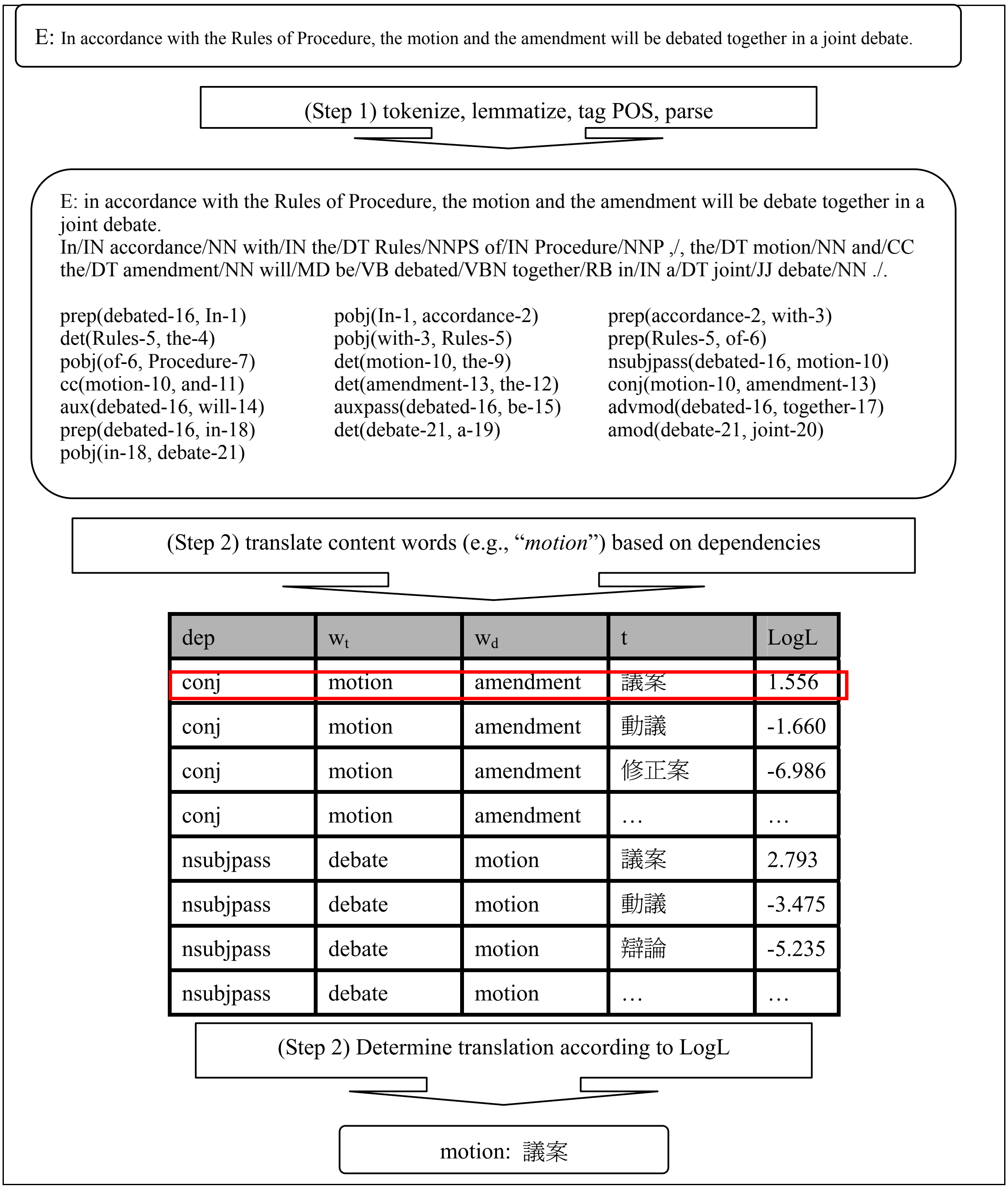
3.3 Word Translation Disambiguation at Runtime

After the decision list and context-independent translation probabilities are obtained in the training process, we can then use them to disambiguate translations for the words in a sentence containing the words. The process of word translation disambiguation at runtime is shown in Figure 6.

Step 1: Parse the input sentence by a dependency parser
Step 2: Select the highest score translation by using dependencies
Step 3: Use P (*t* | *w, p*) to predict translation for unseen dependency relationship
Step 4: Use P (*t* | *w*) to predict translation for unseen word/pos combination

Figure 6. Outline of the process at run time

In Step 1 we exploit a parser to obtain word tokens, POS tags, and dependency relationships of given sentence, and then we lemmatize all tokens except for words that are tagged as "NNP". In Step 2 we determine the translation of words by using the most reliable piece of evidence. Figure 6 shows the process at runtime by using the sentence *"In accordance with the Rules of Procedure, the motion and the amendment will be debated together in a joint debate."* as an example. In some situation, there is no dependency relationships information available to help us translate the word. In Step 3 we use POS information to find the translation of the word for unseen dependency relationship. In Step 4 we take the highest frequency translation to be the translation of the word. If the word is not in our training data, we cannot translate the word.



dep	w_t	w_d	t	LogL
conj	motion	amendment	議案	1.556
conj	motion	amendment	動議	-1.660
conj	motion	amendment	修正案	-6.986
conj	motion	amendment	…	…
nsubjpass	debate	motion	議案	2.793
nsubjpass	debate	motion	動議	-3.475
nsubjpass	debate	motion	辯論	-5.235
nsubjpass	debate	motion	…	…

Figure 7 An example of Step 1 and Step 2

4 Experiments and Evaluation

This approach was designed to disambiguate the translation of the words in the given sentence, by using the statistical properties of the dependency relationships with word to translation. In this section, we first describe the details of the experiments for the evaluation (Section 4.1). Then, we introduce the test data and automatic evaluation methodology and results. (Section 4.2)

4.1 Experimental Setting

In this section we will describe the implementation and experiments of the method described in section 3. For training the proposed model, we used a collection of approximately 740,000 sentence pairs from Hong Kong News English-Chinese Corpus (HKNC1997~2003) and approximately 1,375,000 sentence pairs obtained from Hong Kong Hansard English-Chinese Corpus (HKLC1985~2003).

First, to preprocess the training data, we used Stanford Parser (Version 1.5.1) to implement tokenizing, POS tagging, and dependency parsing. We filtered out the English sentences with word length longer than forty or have some unusual letters. After filtering, we were left with approximately 630,000 sentence pairs from HKNC and approximately 1060,000 sentence pairs from HKLC. Then we use an in-house word lemmatization tool to lemmatize each word in the English sentences. We also segment each Chinese sentence using a word segmentation tool developed by CKIP in Academia Sinica. Finally, we use the Giza++v2 toolkit made available at (www.fjoch.com/GIZA++.html) to obtain word alignment information for the training data. In our experiment, we only use the direction of Chinese to English for word alignment information part. After filtering some errors that occurred in the word alignment process, we were left with about 556,000 sentence pairs from HKNC and about 983,000 sentence pairs from HKLC for training, and we reserved 3,500 sentence pairs obtained from HKNC and 9,500 sentence pairs obtained from HKLC for testing.

Second, we grouped POS's used in the Stanford Parser into nine groups. Table 4 shows the grouping of parts of speech. The grouping was done to reduce sparseness.

Table 4. The nine POS groups

Pos Group	Original tags	Notes
Light Verb	have, do, know, think, get, go, say, see, come, make, take, look, give, find, use	15 high-frequency verbs (Svartvil and Ekedahl 1995)
V	VB, VBD, VBG, VBN, VBP, VBZ, ask	verb but not light verb
N	FW, NN, NNS, PDT	noun
NNP	NNP, NNPS	proper noun
C	CD	quantifier
$	$	$, no., rule, section, …
J	JJ, JJR, JJS, a	adjective
R	RB, RBR, RBS, RP	adverb
F	the other tag	function word

Third, after calculating count (t, w_t, d, w_d) and count ($\bar{t}$, w_t, d, w_d) described in section 3.2.3, we smoothed the counts for the unseen translations for w_t and w_d that have dependency relationship d using held out estimator that is purposed by Jelinek and Mercer(1985). We split training data into two parts that have equal number of sentence pairs. One was used as training data and the other was used as held out data, and then we changed the role of two parts and did held out estimation again. Table 5 shows the final modified number N. Table 6 shows the results of smoothing. Every count ($\bar{t}$, w_t, d, w_d) had to add the counts for unseen translations.

Table 5. The average of two held out estimators

Count C	Obs. counts Set 1	Obs. counts Set 2	Smoothing counts
0	0.87995	0.87965	0.87980
1	0.25552	0.25562	0.25557
2	1.08762	1.08368	1.08565
…	…	…	…
8	7.14678	7.13740	7.14209
9	8.20037	8.15285	8.17661

Table 6. An example of smoothing

dep	wd	wt	t	LogL	N	smoothing N	N'	smoothing N'
nsubjpass	pass	motion	議案	1.608	294	294	58	58.8798
nsubjpass	pass	motion	動議	-1.975	43	43	309	309.8798
nsubjpass	pass	motion	議題	-5.100	3	2.1331	349	349.8798
nsubjpass	pass	motion	決議案	-5.778	2	1.08565	350	350.8798
nsubjpass	pass	motion	表決	-7.228	1	0.25557	351	351. 8798
nsubjpass	pass	motion	…	…	…	…	…	…

Table 7. The properties of test data

property	HKNC	HKLC
sentences	1,500	3,800
all words	31,569	84,290
content words	16,980 (53.79%)	42,343 (50.23%)
LV	585 (1.85%)	2,142 (2.54%)
be	858 (2.72%)	2,798 (3.32%)
F(Function words)	13,146 (41.64%)	37,007 (43.90%)

4.2 Evaluation and Discussion

In this section, we describe our test data and evaluation methodology (4.2.1). We then show the evaluation result of our experiment and give some discussions (4.2.2).

4.2.1 Test Data and Evaluation Methodology

We randomly choose 1,500 sentences out of 3,500 sentence pairs from HKNC and 3,800 sentences out of 9,500 sentence pairs from HKLC for testing. Then we translate the content words in the given sentences of test data. We did not consider the translation of the words that POS tagged in the group F and LV, also not did we consider the translation of the verb “*be*”. Table 7 shows the properties of our test data.

The traditional WSD evaluation methodology relies on human judgment. In our experiment, we do not focus on the sense of the words, but rather the translation of the content words in the given sentences. Since it is infeasible for human to evaluate such a large set of data, we developed a BLEU-like automatic evaluation methodology. We evaluate one sentence at a time. First, we combine all translations of content words that in the given sentence. Identical an overlapping translations of two neighboring word are combined and redundancy are removed. For example, see Figure 8 for more details.

EPD/環保署 and Transport/運輸署 Department/署 will also/亦 step/加強 up/加強 their programmes/計劃 to educate/教育 vehicle/車主 owners/車主 and mechanics/技師 to exercise/行使 their responsibility/責任 to maintain/維修 vehicles/車輛 properly/妥善 .

Combining translations

環保署 → 環保署運輸署 →環保署運輸署 → 環保署運輸署亦 → 環保署運輸署亦加強 → 環保署運輸署亦加強 → 環保署運輸署亦加強計劃 → 環保署運輸署亦加強計劃教育 → 環保署運輸署亦加強計劃教育車主 → 環保署運輸署亦加強計劃教育車主 → 環保署運輸署亦加強計劃教育車主技師 → … → 環保署運輸署亦加強計劃教育車主技師行使責任維修車輛妥善

Figure 8. An example to show how to combine the translations of the sentence

Second, we calculated unigram precision rate based on the aligned Chinese sentence as the reference translation. Figure 9 shows an example of the process. Third, we filtered the highest ten percentage and lowest ten percentage sentences for data balance, and then we average the score of middle eighty percentage sentence to be the result.

Result after combine:環保署運輸署亦加強計劃教育車主技師行使責任維修車輛妥善

Answer: 環境保護署和運輸署亦將加強宣傳，要求車主和技師妥善地保養和維修車輛。

$$\text{Score} = \frac{\text{the number of the Chinese characters matching the reference}}{\text{number of the Chinese characters produced}} = \frac{19}{27}$$

Figure 9. An example to show how to calculate the score of the sentence

4.2.2 The Results of WTD of Different Methods

We used five different methods to disambiguate word translations. Table 8 shows the results of WTD. Baseline is the result of using only P (t | w) to estimate the translation of the content words, while baseline with POS is the result of using both P (t | w, p) and P (t | w), dependency method (all) is the result of using the process described in 3.3, window size 1 is the result of using a window based co-occurrence (the word to the right or left of the word in question) instead of dependency, and dependency method (some) is the result of using the process described in 3.3 leaving out five kinds of dependency relationships, including *determiner, negative, possessive, coordinating conjunction, preposition*.

Table 8. Results of WTD in different methods

Method	HKNC	HKLC	HKNC+HKLC
Baseline	0.582	0.564	0.569
baseline + POS	0.589	0.569	0.575
window size 1	0.698	0.643	0.659
dependency method (all)	0.714	0.686	0.694
dependency method (some)	0.716	0.685	0.694

The results in Figure 10 indicate that the dependency method obviously outperforms baseline with POS and also outperforms window based co-occurrence approach. We also found that using POS can only improve slightly and ignoring some kinds of dependency relationships does not affect the results too much.

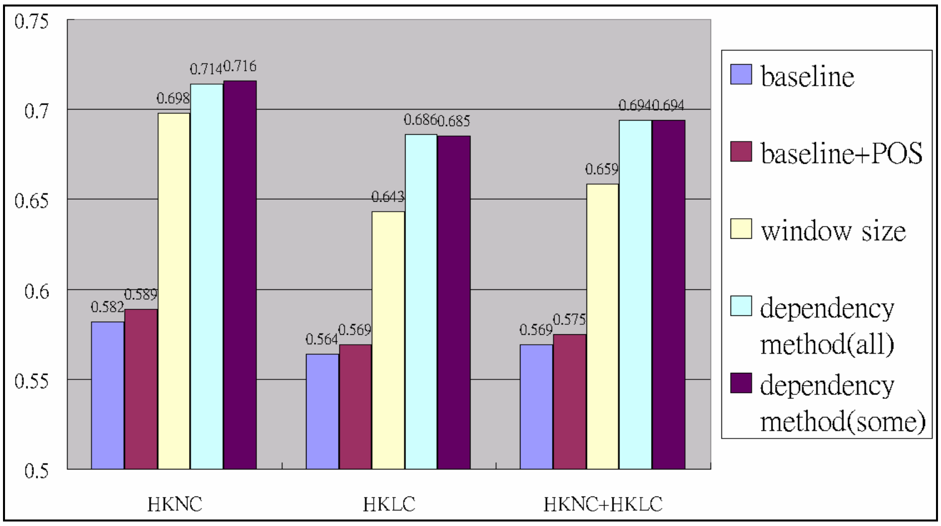


Figure 10. Results of WTD in different methods

However, we did not evaluate the performance of translations based on dependency. As shown in Table 9 that over ninety percent of the cases, words are translated via dependency.

Table 9. The percentage of the word translations when we used dependency method (some)

type	HKNC	HKLC
all content words	16,980	42,343
dependency method	15,698 (92.45%)	38,587 (91.13%)
baseline +POS	1,199 (7.06%)	3,610 (8.53%)
baseline	12 (0.67%)	34 (0.08%)
no answer	71 (0.42%)	112 (0.26%)

As shown in Figure 11, in different number of sentences, the results of HKNC are better than the results of HKLC. We believe this is a result of the different character of the corpora and not the different number of sentences in the two corpora.

Because of data sparseness, we may not calculate a suitable score for translations of the words with dependency relationships. If we use lager training set, we may improve the performance. Some word translation errors may be caused by word alignment errors. In addition, there also have some problems caused by incorrect segmentation. For example, “*吸煙者*” is segmented into “*吸煙*” and “*者*”, but in our module we only consider the one to one case, therefore the word "*smoker*" will be translate to "*吸煙*" and not "*吸煙者*".

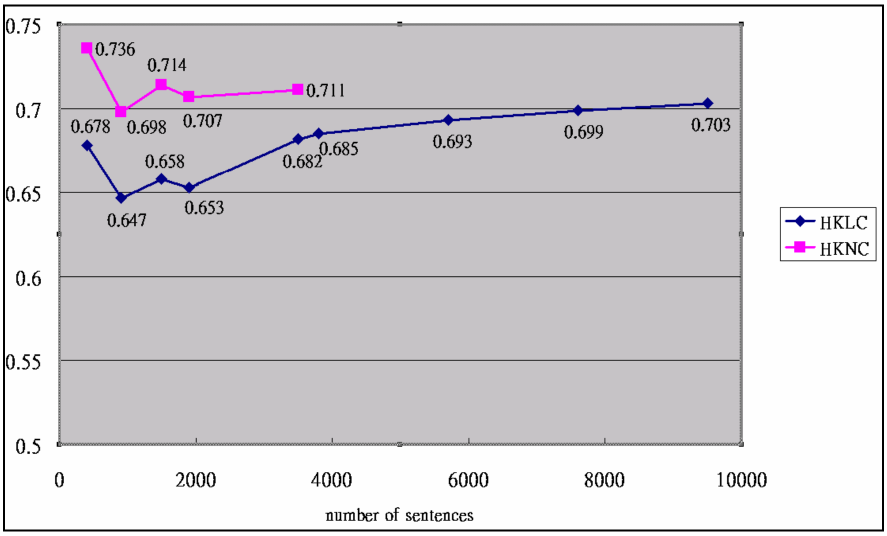


Figure 11. Results of WTD in different corpora

Table 10 an example to explain majority voting methodology

Dep	wt	wd	t	LogL
advmod	plant	at	植物	2.502
Advmod	plant	at	廠	-3.131
Dep	wt	wd	t	LogL
Amod	plant	chemical	處理廠	1.290
Amod	plant	chemical	一個堆	-3.568
Amod	plant	chemical	廠	-3.568

5 Future Work and Conclusion

In summary, we have introduced a method for word translation disambiguation, which improves the ability to disambiguate the translations of the content words in the given sentence using a dependency-based translation model trained as a parallel corpus. We have implemented and evaluated the method using a bilingual English-Chinese corpus. We have shown that the method outperforms the baseline. In addition, we also found some kinds of dependency are more effective than others. Moreover, we have purposed an automatic BLEU-like evaluation methodology for WTD. The results of word translation disambiguation can assist user in reading English, and also can be used as additional input information for an MT system to improve the performance.

Many future directions present themselves. First, it would be interesting to extend the method to translate all words in the sentence including function words. Second, we can give different weight to different type of dependency since we believe different type of dependency relationships have different level of effectiveness. Third, we are currently using the dependency relationships with the highest score, but we can also consider all dependency relationships of the word in the given sentence. Table 10 shows an example that "*plant*" that has two dependency relationships with "*at*" and "*chemical*". In the way we described in our approach, we will choose "*植物*" as the answer. If we combine scores of two dependency relationships to calculate a new score for each translation, we may choose “廠” as our answer which seems to be more suitable.

References

1. Bengt Altenberg and Sylviane Grange. “The grammatical and lexical patterning of make in native and non-native student writing”. Applied Linguisics, 22(2), 173-194, 2001.
2. Clara Cabezas and Philip Resnik. “Using WSD Techniques for Lexical Selection in Statistical Machine Translation”. http://handle.dtic.mil/100.2/ADA453538 , July 2005.
3. Marine Carpuat and Dekai Wu. “Word Sense Disambiguation vs. Statistical Machine Translation”. In 43th Annual Meeting of the Association for Computation Linguistics (ACL 2005), 2005.
4. Dagan, Ido, Alon Itai, and Ulrike Schwall. "Two Languages are More Informative than One". In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL91). Berkeley, 1991.
5. Philipp Koehn, and Kevin Knight. “Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm”. In Proceedings of the 17th National Conference on Artificial Intelligence, pages 711–715, Austin, TX, 2000.
6. Cong Li and Hang Li. “Word translation disam-biguation using bilingual bootstrapping”. In Proceed-ings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 343-351, 2002.
7. Yajuan Lü, Ming Zhou, Sheng Li, Changning Huang, Tiejun Zhao (2001b). “Automatic translation template acquisition based on bilingual structure alignment”. International Journal of Computational Linguistics and Chinese Language Processing. 6(1), pp. 1-26, 2001
8. Hwee Tou Ng, BinWang, and Yee Seng Chan. “Exploiting parallel texts for word sense disambiguation: An empirical study”. In Proceedings of ACL-03, Sapporo, Japan, pages 455–462, 2003.
9. K. Papineni, S. Roukos, T. Ward, and W. Zhu. “Bleu: a method for automatic evaluation of machine translation”. In Proceedings of 40th Annual Meeting of the ACL, Philadelphia, 2002.
10. Ted Pedersen. “A simple approach to building ensembles of naive Bayesian classifiers for word sense disambiguation”. In Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics, Seattle, 2000.
11. Thanh Phong Pham, Hwee Tou Ng, and Wee Sun Lee. “Word sense disambiguation with semi-supervised learning” AAAI-05, The Twentieth National Conference on Artificial Intelligence, 2005.
12. Dan Klein and Christopher D. Manning. “Fast exact inference with a factored model for natural language parsing”. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, Advances in Neural Information Processing Systems 15, Cambridge, MA.MIT Press, 2003.
13. D. Yarowsky. “Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French”. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM, 1994.
14. D. Yarowsky. “Unsupervised word sense disambiguation rivaling supervised methods”. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196, 1995.
15. Ming Zhou, Yuan Ding, and Changning Huang. “Improving translation selection with a new translation model trained by independent monolingual corpora”. Computational linguistics and Chinese Language Processing. Vol. 6, No. 1, pp 1-26, 2001.