Statistical Query Translation Models for Cross-Language Information Retrieval

JIANFENG GAO
Microsoft Research

JIAN-YUN NIE
University of Montreal

MING ZHOU
Microsoft Research Asia

Query translation is an important task in cross-language information retrieval (CLIR), which aims to determine the best translation words and weights for a query. This paper presents three statistical query translation models that focus on resolution of query translation ambiguities. All the models assume that the selection of the translation of a query term depends on the translations of other terms in the query. They differ in the way linguistic structures are detected and exploited. The co-occurrence model treats a query as a bag of words, and use all the other terms in the query as the context for translation disambiguation. The other two models exploit linguistic dependencies among terms. The noun phrase (NP) translation model detects NPs in a query, and translates each NP as a unit by assuming that the translation of a term only depends on other terms within the same NP. Similarly, the dependency translation model detects and translates dependency triples, such as verb-object, as units. The evaluations show that linguistic structures always lead to more precise translations. The experiments of CLIR on TREC Chinese collections show that all the three models have a positive impact on query translation, and lead to significant improvements of CLIR performance over the simple dictionary-based translation method. The best results are obtained by combining the three models.

Categories and Subject Descriptors: H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval Models*

General Terms: Algorithm, Languages, Theory

Additional Key Words and Phrases: Query translation, CLIR, Statistical models, Linguistic structures

---

1. INTRODUCTION

With the huge expansion of documents in many languages on web and the desire of non-native speakers of the language to be able to retrieve them, cross-language information retrieval (CLIR) systems have become increasingly important in recent years.

The goal of CLIR is to resolve the language mismatch between documents and queries. This can be achieved by translating either documents or queries. Since translating queries is more efficient and easier, research in the area of CLIR has focused mainly on methods for query translation. The goal of query translation is to determine the best translation words and weights for a given query. A common approach to query translation is based on dictionary. This approach is popular because of its simplicity and the increasing availability of machine readable bilingual dictionaries. However, besides the problem of incompleteness of the dictionary, we are also faced with the problem of ambiguity in translation, i.e., multiple translations are stored in dictionary for the same word. This paper focuses on this problem: We try to select the best set of translation words by solving ambiguities according to the coherence of translation words and syntactic information.

In general, information retrieval (IR) can be viewed as a reasoning process that tries to determine if there is a relationship between a document and a query, and how strong the relationship is [van Rijsbergen 1986]. This process may involve any type of reasoning, such as reasoning based on synonymy relations, which is often used in a query expansion process. Query translation can also be viewed as a reasoning process that tries to determine a related query, although in a different language. The relationship between query translation and inference in IR has been clearly shown in Nie [2003]: It turns out that CLIR is a particular case of more general inferential IR that exploits relationships between terms. In query translation, the term relationships used are translation relationships. In this paper, we will concentrate on several concrete approaches to implement such a process to determine (or infer) the best query translation according to both the translation relationships and the other translation words. Our approaches are defined within the statistical framework. Statistical reasoning is popular in IR due to the fact that most available knowledge in IR is of statistical nature. Query translation in CLIR is not an exception. Much of the translation and linguistic knowledge available is statistical: a word can be translated to another to a certain degree, and different words in a language are coherent (i.e. tend to co-occur) to some degree. The statistical framework allows us to integrate such different types of knowledge.

In this paper we present three statistical models for query translation for English-Chinese CLIR. These models differ in the information used to determine the translations. The first model is the *decaying co-occurrence model*. It assumes that the selection of the translation of a query term depends on both the translation probability and the translations of other terms within a query. The best set of translation terms contains those that are good translations of the original terms, and form a coherent set together. Following the previous studies [e.g., Adrian 2000; Ballesteros and Croft 1998; Gao et al. 2000; Gao et al. 2001b; Gao et al. 2002b], a mutual information value between translation terms is estimated according to their co-occurrence within a predefined window. The translation term that has the highest mutual information score with other translation terms is considered to be the most coherent and is selected. In addition, we also take into account the distance between terms, assuming that closer terms have stronger relationships [Gao et al. 2002b].

While in the co-occurrence model, a query is simply viewed as a sequence of words without any linguistic structure, the other two models take advantage of the syntactic structure among terms. In the *noun phrase (NP) translation model*, we first identify NPs in a query, and then translate them as units by assuming that the selection of a translation only depends on the other translations within the same NP [Gao et al. 2001b].

The third model is the *dependency translation model*. A dependency, represented as a triple, is a pair of words that have a syntactic dependency relation, such as verb-object. It is our observation that there is a strong correspondence in dependency relations in the translation between English and Chinese, despite the great differences between the two languages. In the dependency translation model, we first detect dependency triples in a query using a parser, and then translate them as units. Similar to the NP translation, we assume that the selection of a translation only depends on the translation of the other word in the same dependency triple [Gao et al. 2002b]. While NPs only capture dependence of adjacent terms in a query, dependency triples can captures syntactic dependences

between non-adjacent terms. Therefore, the dependency translation model can be viewed as a generalization of the NP translation model.

We evaluate the three models using TREC Chinese collections. Our results show that each of the methods achieves significant improvement over the simple dictionary-based approaches. We demonstrate that linguistic structures such as phrases and dependency triples are beneficial to query translation if they can be detected and used properly.

The remainder of this paper is organized as follows. Section 2 formulates the query translation problem, and discusses major research tasks. Sections 3 to 5 describe in detail each of the three query translation models, respectively. Evaluations are presented where appropriate. Section 6 presents experimental results of CLIR on TREC Chinese collections. The discussion and related work are presented in Section 7. Finally, the paper is concluded in Section 8.

2. QUERY TRANSLATION

We refer to the language of queries as source language (i.e., English in this paper), and the language of documents as target language (i.e., Chinese in this paper). In the rest of this paper, we use the following notation.

- Let an English query be denoted by $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$, where n is the number of distinct terms in $\mathbf{e}$. We also assume some way of detecting linguistic structures $\mathbf{s}$ (e.g., phrases or dependency triples) of $\mathbf{e}$.
- We assume there is an English-Chinese bilingual dictionary $\mathbf{D}$, which defines for each English query term e_i a set of m distinct Chinese translations: $\mathbf{D}(e_i) = \{c_{i,1}, c_{i,2}, \ldots, c_{i,m}\}$.
- We assume a one-to-one word translation between source and target languages[1]. Therefore, the translated query is represented by $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$, where each word c_i is selected from the translation set provided by the dictionary, i.e., $c_j \in \mathbf{D}(e_i)$. In particular, either e_i or c_j can be an empty word, as suggested by Brown et al. [1993].
- We assume some way of generating a set of candidate query translations given $\mathbf{e}$, denoted by $\mathbf{GEN}(\mathbf{e})$. In our experiments, $\mathbf{GEN}(\mathbf{e})$ is represented as a lattice where each node is a Chinese word $\mathbf{c}_{i,j}$, i.e., the j-th translation of the i-th query term in $\mathbf{e}$.

The task of a query translation model is to assign a score for each of the translate candidates in $\mathbf{GEN}(\mathbf{e})$, and select the one with the highest score:

$$\mathbf{c}^* = \arg\max_{\mathbf{c} \in \mathbf{GEN}(\mathbf{e})} Score(\mathbf{c}, \mathbf{e}, \mathbf{s}) \quad (1)$$

Notice that Equation (1) indicates that the translation model works as a ranking function. Therefore, the score can be either the conditional probability $P(\mathbf{c} \mid \mathbf{e}, \mathbf{s})$ or any order-preserving transformation of the probability. The probability is typically broken down into its component probabilities as

[1] This assumption is made to simplify the model. As it does not hold in reality, it may lead to some translation errors. For example, a compound English term could be translated into a single Chinese word. But using this assumption, we may obtain a multi-word Chinese translation. However, as our ultimate goal is query translation, it is not necessary to preserve grammaticality in the translation and we found in our experiments that generating translation words more than necessary does not hurt IR, as described in Section 6. Therefore, we believe that the assumption is reasonable for CLIR.

$$P(\mathbf{c} \mid \mathbf{e}, \mathbf{s}) = \prod_{i=1}^{n} P(c_i \mid \Phi(\mathbf{e}, \mathbf{s}, \mathbf{c}_i)) \quad (2)$$

where $\mathbf{c}_i$ denotes the set of translated query terms excluding c_i, and Φ is a function that maps ($\mathbf{e}$, $\mathbf{s}$, $\mathbf{c}_i$) into equivalence classes according to the independence assumptions used by the translation model.

Sections 3 to 5 will describe the three query translation models in turn. For each of them, we focus the discussion on the following three basic research tasks:

- **Modeling:** defining and detecting the linguistic structures **s**, and defining the mapping function Φ, i.e., the independence assumptions (e.g., the Markov assumption) that are used to break down the probabilities in Equation (2),
- **Training:** learning the free parameters of the statistical model using bilingual or monolingual training data, and
- **Decoding:** performing the argmax operation of Equation (1) in an efficient way.

3. CO-OCCURRENCE MODEL

3.1 Basic Principle

A correct translation is the one that fits well the context of the whole sentence (or query). The sentence in source language reflects well the context of the sentence, but it would be difficult to directly compare a translation with the source sentence unless there is a well-defined similarity measurement between words across languages. An alternative approach is to assume that all the translations selected for the other words of the source sentence form a specification of the context. Then a good translation is the one that has a high *cohesion* with the other translations. The advantage of the alternative approach is that there is no need to measure cross-language word similarities. Only relationships between words of the same language are used. They can be obtained through co-occurrence statistics in a monolingual text corpus. This is the principle of co-occurrence approach to translation selection, which will be described in this section. It can also be expressed as follows: *Correct translations of query terms tend to co-occur in the target language and incorrect translations do not*.

In what follows, we describe in turn (1) how to measure the term similarity via a so-called decaying co-occurrence model, and (2) how to select an optimal set of query term translations.

3.2 Decaying Co-occurrence Model

The definition of similarity between two terms, w_i and w_j, can take different forms of co-occurrence statistics. Mutual information is among the most commonly used ones [van Rijsbergen 1979]. It is defined as follows

$$MI(w_i, w_j) = P(w_i, w_j) * \log(\frac{P(w_i, w_j)}{P(w_i) * P(w_j)}) \quad (3)$$

where

$$P(w_i, w_j) = \frac{C(w_i, w_j)}{\sum_{w_i', w_j'} C(w_i', w_j')},$$

$$P(w) = \frac{C(w)}{\sum_{w'} C(w')}$$

Here $C(w_i,w_j)$ is the frequency of co-occurrences of terms w_i and w_j within a predefined window (e.g., a sentence) in the collection of the target language, $C(w)$ is the number of occurrences of term w in the collection.

We observe that in Equation (3) any co-occurrence within the windows is treated in the same way, no matter how far they are from each other. In reality, we find that closer words usually have stronger relationships, thus should be more similar. Therefore, we add a distance factor $D(w_i, w_j)$ in the mutual information calculation. This factor decreases exponentially when the distance between two terms w_i and w_j, increases, i.e.,

$$D(w_i, w_j) = \exp(-a * (Dis(w_i, w_j) - 1)) \quad (4)$$

where α is the decaying rate, which is determined empirically, and $Dis(w_i,w_j)$ is the average distance between w_i and w_j in the corpus.

Term similarity in the extended co-occurrence model consists of two components: (1) the mutual information $MI(w_i, w_j)$ as defined in Equation (3), and (2) the decaying factor $D(w_i, w_j)$:

$$sim(w_i, w_j) = MI(w_i, w_j) \times D(w_i, w_j) \quad (5)$$

3.3 Training

The decaying co-occurrence model parameters (i.e., $MI(.)$ and $D(.)$ in Equation (5)) are estimated on a Chinese newspaper corpus consisting of approximately 80 million characters. The text corpus was first word-segmented using a Chinese word segmentation system MSRSeg [Gao et al. 2005a]. Then all stop words were removed. We set the window size as a sentence when estimating $MI(.)$ and $D(.)$. To deal with the sparse data problem, we used Good-Turning smoothing when estimating $P(w_i, w_2)$ and $P(w)$ in Equation (3). That is, we assume that the number of unseen events (i.e., term w and term-pair (w_i, w_j) in our case) is the same as the number of the events occur once. As a result, the final estimate for $P(w)$, for example, is $P(w)=r^*/N$, where $r^*=(r+1)n_{r+1}/n_r$. Here r is the number of occurrences of w in the training corpus, N is the total number of word occurrences, and n_r is the number of words which occur r times in training data.

The decaying rate α in Equation (4) was optimized using tests with query expansion in Chinese monolingual IR. That is, for each term, we expand it by an additional synonym term that has the highest cohesion value with the other words of the original query. This expansion task is very similar to the translation selection in CLIR. Therefore, it gives a good indication on the possible impact on query translation.

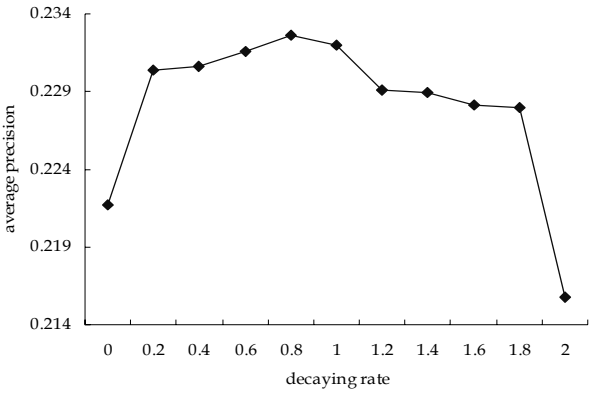


Figure 1. Impact of the decaying rate on query expansion

A Chinese synonym dictionary is generated for query expansion from the LDC bilingual dictionaries[2] as follows: For each Chinese term c and each of its English translation e, we consider all the Chinese translations c' of e as synonyms of c.

Our tests were carried out on several TREC Chinese collections. For example, Figure 1 shows the TREC-9 retrieval results with query expansion while the decaying rate varies. It can be seen that the decaying co-occurrence model performs generally better (when $\alpha < 2$) than the basic co-occurrence model (when $\alpha = 0$). With a decaying rate of 0.8, we obtain the best performance of the average precision 23.3%, which is 5% better than the basic model. These experiments show that the decaying factor allows us to better distinguish between strong and weak term relationships. As the problem of translation selection in CLIR is similar to this expansion task, we can expect a similar effect with the decaying factor. Although, in our CLIR experiments we found that the optimal value of the decay rate varies slightly from collection to collection, we will only report CLIR results in Section 6 by setting the decaying rate to 0.8.

3.4 Approximate Translation Selection Algorithm

Given the measurement of term similarity, ideally, we should select for each query term the translation that co-occurs the most often with (or the most similar to) the selected translations of other terms in the same query. However, finding such an optimal translations is computationally very expensive, as will be described below. Therefore, we use an approximate greedy algorithm as follows [e.g., Adriani 2000; Gao et al. 2001b]:

(a) Given a query $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$ in the source language, for each query term e, we define a set of m distinct translations according to a bilingual dictionary $\mathbf{D}$: $\mathbf{D}(e_i) = \{c_{i,1}, c_{i,2}, \ldots, c_{i,m}\}$.

(b) For each $\mathbf{D}(e_i)$

[2] http://morph.ldc.upenn.edu/Projects/Chinese/.

a) For each translation $c_{i,j} \in \mathbf{D}(e_i)$, define the similarity score between the translation $w_{i,j}$ and a set $\mathbf{D}(e_k)$ ($k \neq i$) set as the sum of the similarities between $c_{i,j}$ and each translations in the set $\mathbf{D}(e_k)$ according to Equation (5), i.e.,

$$sim(c_{i,j}, \mathbf{D}(e_k)) = \sum_{c_{k,l} \in \mathbf{D}(e_k)} sim(c_{i,j}, c_{k,l}) \quad (6)$$

b) Compute the cohesion score for $c_{i,j}$ as

$$cohesion(c_{i,j} \mid \mathbf{e}, \mathbf{D}) = \log[\sum_{\mathbf{D}(e_k)} sim(c_{i,j}, \mathbf{D}(e_k))] \quad (7)$$

c) Select the translation $c \in \mathbf{D}(e_i)$ with the highest cohesion score

$$c = \arg\max_{c_{i,j} \in \mathbf{D}(e_i)} cohesion(c_{i,j} \mid \mathbf{e}, \mathbf{D}) \quad (8)$$

Apparently, the above algorithm is sub-optimal. As pointed out in Liu et al. [2005], the cohesion score for a translation as in Equation (7) is computed with regard to all possible translations of other query terms. It does not differentiate correct translations from incorrect ones. As a result, the translation of different query terms is determined independently. In spite of the deficiency, the greedy search algorithm has been widely used since an exact algorithm is prohibitively expensive. In the next subsection, we will formulate the translation selection problem under the framework of graphic model (GM) [e.g., Jordan et al. 1999], and discuss the underlying assumptions of the greedy algorithm.

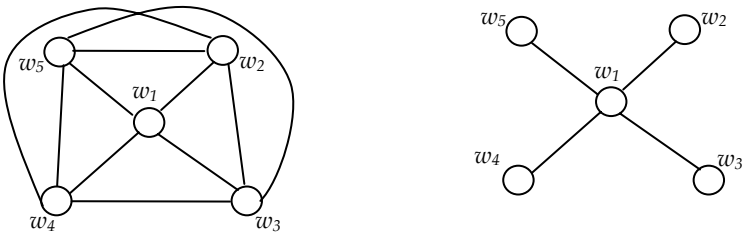
3.5 A GM View

A query translation model can be viewed as an undirected GM. For example, Figure 2 shows a query translation model of a 5-term query. Each node represents a distribution of a translation set of a query term. The edges of the graph represent a set of independency assumptions among query term translations. The task of query translation is to find a set of translations that maximize the joint probability $P(w_1, w_2, w_3, w_4, w_5)$.

The GM view imposes three research tasks of query translation. The first is how to generate translation candidates for each term, and how to model the distribution of the candidates. Traditionally, a bilingual dictionary is used and all translations of a query term are assumed to be uniformly distributed. We may also induce a distribution using a statistical translation model learned from parallel bilingual corpora.

The second is how to determine the graph topology, i.e., what independence assumptions we may use. The third is how to compute the joint probability. These two problems are closely related. The efficiency of the joint probability computing largely depends on the graph topology.

In the co-occurrence model as described above, we assume that the selection of each translation is consistent with the selected translations for other query terms. Therefore, we assume that the five nodes form a clique as shown in Figure 2 (a). Suppose that we wish to compute the marginal probability $P(w_1)$. We obtain this marginal by summing over the other variables as:



(a) (b)

Figure 2. GM of query translation

$$P(w_1) = \frac{1}{Z}\sum_{w_2}\sum_{w_3}\sum_{w_4}\sum_{w_5} h(w_1, w_2, w_3, w_4, w_5)$$

where $h(.)$ is a feature function, and Z is a normalization factor.

We see that the computational complexity of $P(w_1)$ scales as d^6 (assuming that each query term has d possible translations). This is prohibitively expensive even for a very short query. We therefore resort to an approximated word selection algorithm as described in Section 3.4 by introducing a translation independence assumption. The corresponding GM is shown in Figure 2 (b). Now, $P(w_1)$ can then be factored as:

$$P(w_1) = \frac{1}{Z}\sum_{w_2} h(w_1, w_2)\sum_{w_3} h(w_1, w_3)\sum_{w_4} h(w_1, w_4)\sum_{w_5} h(w_1, w_5)$$

Note that no more than two variables appear together in any summand, and thus the computational complexity is reduced to d^2. As discussed, the reduction of complexity comes with the sacrifice of accuracy.

In general, the computation complexity depends on the largest size of the clique in the graph. The NP and dependency translation models described in Sections 4 and 5 are used to implement the idea that the linguistic structure of a sentence can be utilized to identify cliques. Linguistic units, such as NPs or dependency triples, can be translated as units and the translation can be done accurately using only internal information of the unit. As a consequence, the graph would be divided into a few smaller sub-graphs. The probability of each sub-graph can be inferred independently, with an optimal order that leads to a lower computation complexity.

Using the three translation models that we propose in this paper, our query translation process can be cast in a sequential manner as follows.

- Identify NPs and dependency triples of a query.
- Translate words in NPs using the NP translation model described in Section 4.
- Translate words in dependencies using the dependency translation model described in Section 5.
- Translate remaining words using the co-occurrence model.

We call the above approach to combining the three translation models the *sequential combining approach*. Using such an approach, for each query term, we only keep one translation. We can also combine these translation models using a *parallel combining approach*, where for a given query, we first obtain three sets of translation terms obtained using the three translation models respectively, and then combine the three sets. As a result, there are multiple translations for each query term. In our experiments, we only use the sequential combining approach when we evaluate directly the translation accuracy, and use the parallel combining approach when we evaluate the CLIR performance. This is due to the fact that multiple translations often lead to better CLIR performance [Xu and Weischedel 2000; Gao et al. 2000]. Readers can refer to Section 6 for a detailed description and analysis.

4. NP TRANSLATION MODEL

Although the translation of multi-word phrases is usually more precise than a word-by-word translation, many significant NPs are not stored in the dictionary. For instance, in TREC-9 queries, more than 50% of noun phrases, which can be detected by our method described in this section, are not in our dictionary.

In the previous IR research, NPs have been identified using a set of syntactic patterns [e.g., Ballestero and Croft 1997; Fagan 1988]: Sequences of nouns and adjective-noun pairs were taken as phrases. However, this simple method has not produced consistent improvement. Fagan [1988] reported a decrease in performance, while Ballestero and Croft [1997] did not obtain a significant improvement over single words. One of the problems is that this simple approach often over-generates NPs: non-NPs may be identified as NPs. This may negatively affect the monolingual IR performance because of a deformed distribution of occurrences of these items. In addition, the identified phrases are still translated word-by-word in Ballestero and Croft [1997].

In our approach, we use a more sophisticated NP identification process. It is carried out in a bottom-up manner: we first identify base NPs, and then complex NPs. The reason to separate the process into two steps lies in the fact that base NPs can be identified with high accuracy, while the complex NPs cannot be. Therefore, we only use a small set of syntactic patterns in the second step in order to select sufficiently reliable complex NPs that may affect significantly the performance of retrieval. Though we focus on NPs in this section, the methods can be extended to other consecutive phrases, such as the chunks defined in Tjong and Buchholz [2000].

4.1 Base NP Identification

4.1.1 Principle

A base NP is a simple noun phrase that does not contain other noun phrases recursively. For example, the elements within [...] in the example shown in Figure 3 are base NPs. The part-of-speech (POS) tags NNS (plural noun), IN (preposition), and VBG (verb-ing) etc. are those defined in Marcus et al. [1993].

The identification of base NPs usually involves two steps: (1) POS tagging, and (2) base NP chunking.

[Measures/NNS] of/IN [manufacturing/VBG activity/NN] fell/VBD more/RBR than/IN [the/DT overall/JJ measures/NNS] ./.

Figure 3. An example sentence with base NP brackets

In classical statistical approaches [Church 1988; Ramshaw and Marcus 1998], these two steps have been separated. POS tagging often serves as a precursor, and NP chunking uses POS patterns (e.g., DT-JJ-NNS) that are learnt from a tagged corpus.

By separating the two steps, the solution of the first step is used in the second step as if it is certain. The uncertainty involved in the first step is no longer taken into account in the second step. In fact, the correct solution of the first step may be ranked second, third, etc. This is particularly the case when the probabilities of these solutions are close to that of the first solution. Therefore, a too early selection in the first step may be an important source of error.

In our approach, we try to integrate the two steps and their uncertainties together, and use a unified statistical model to choose the globally optimal solution [Xun et al. 2000]. We keep the n-best ($n > 1$) ranked POS assignments in the first step. Then, in the second step, we determine the best base NPs by considering both the probability of POS tagging and that of base NP pattern. The value of n is chosen empirically to obtain the optimum balance between efficiency and accuracy.

4.1.2 Base NP Tagging Model

This section formulates the above two steps in mathematical terms. A more detailed description can be found in Xun et al. [2000]. The task of base NP tagging can be stated as follows. Given an English sentence $\mathbf{e} = \{e_1,\ldots,e_n\}$, we search the most probable base NP sequence $\mathbf{b}^* = \{b_1,\ldots,b_m\}$ ($m \le n$) that maximizes the conditional probability $P(\mathbf{b}\,|\,\mathbf{e})$. The POS tag sequence $\mathbf{t} = \{t_1,\ldots,t_n\}$ is introduced as a hidden variable:

$$P(\mathbf{b}\,|\,\mathbf{e}) = \sum_{\mathbf{t}} P(\mathbf{b},\mathbf{t}\,|\,\mathbf{e})$$

The base NP tagging is performed using the so-called maximum approximation:

$$\begin{aligned}\mathbf{b}^* &= \arg\max_{\mathbf{b}} P(\mathbf{b}\,|\,\mathbf{e}) \\ &\approx \arg\max_{\mathbf{b}}\{\max_{\mathbf{t}} P(\mathbf{t}\,|\,\mathbf{e})P(\mathbf{b}\,|\,\mathbf{t},\mathbf{e})\}\end{aligned} \tag{9}$$

In this formula, $P(\mathbf{t}\,|\,\mathbf{e})$ aims to determine the best POS tags for a sentence $\mathbf{e}$, and $P(\mathbf{b}\,|\,\mathbf{t},\ \mathbf{e})$ aims to determine the best base NP tag sequences from them. Hence the search space consists of the set of all possible POS tag sequences and all possible base NP sequences. In practice, in order to reduce the search space, only n-best POS tagging of $\mathbf{e}$ are retained in the first step. In our implementation, we used the A* algorithm to search for the n-best $\mathbf{t}$ according to the probability $P(\mathbf{t}\,|\,\mathbf{e})$. According to Bayes' rule, we have

$$P(\mathbf{t}\,|\,\mathbf{e}) \propto P(\mathbf{e}\,|\,\mathbf{t})P(\mathbf{t}). \tag{10}$$

We also assume independence among the relationships between tags and English words; and we use a tag trigram model to approximate $P(\mathbf{t})$.

$$P(\mathbf{e} \mid \mathbf{t}) = \prod_{i=1...n} P(e_i \mid t_i)$$
$$P(\mathbf{t}) = P(t_1)P(t_2 \mid t_1) \prod_{i=3...n} P(t_i \mid t_{i-2} t_{i-1})$$

In the second step, we determine the best base NP sequence, given the *n*-best POS sequences. A similar approach to the first step is used. According to Bayes' rule, we have

$$P(\mathbf{b} \mid \mathbf{t}, \mathbf{e}) = \frac{P(\mathbf{e} \mid \mathbf{b}, \mathbf{t}) P(\mathbf{t} \mid \mathbf{b}) P(\mathbf{b})}{P(\mathbf{e} \mid \mathbf{t}) P(\mathbf{t})}.$$

For a give **e** and its **t**, the denominator is a constant and can be dropped. Let $b_{i,j}$ denote a base NP that spans from t_i to t_j. We have

$$P(\mathbf{t} \mid \mathbf{b}) = \prod_{b_{i,j} \in \mathbf{b}} P(t_i ... t_j \mid b_{i,j}) = 1.$$

Therefore, we get Equation (11)

$$P(\mathbf{b} \mid \mathbf{t}, \mathbf{e}) \propto P(\mathbf{e} \mid \mathbf{b}, \mathbf{t}) P(\mathbf{b}). \tag{11}$$

Here, the two terms on the right hand side can be decomposed as follows

$$P(\mathbf{e} \mid \mathbf{b}, \mathbf{t}) = \prod_{i=1...n} P(e_i \mid t_i, b_k)$$
$$P(\mathbf{b}) = P(b_1)P(b_2 \mid b_1) \prod_{i=3...m} P(b_i \mid b_{i-2} b_{i-1})$$

where b_k is the base NP that contains the tag t_i.

4.1.3 Training

The tagging models are trained on Penn Treebank [Marcus et al. 1993]. First of all, all possible base NP patterns (i.e., POS tag sequences) are extracted from the annotated corpus. There are more than 6000 patterns in the Penn Treebank. After being filtered by a set of linguistic rules, 1169 patterns are kept. Then, all parameters in our statistical model are estimated on training data using maximum-likelihood estimation (MLE) with a particular smoothing method, called Modified Absolute Discounting, a variant of the modified Kneser-Ney smoothing method [Chen and Goodman 1998], as described in Gao et al. [2001]. These parameters, as shown in Equations (10) and (11), are (1) $P(t_i \mid t_{i-2}\ t_{i-1})$, (2) $P(e_i \mid t_i)$, (3) $P(b_i \mid b_{i-2}\ b_{i-1})$, and (4) $P(e_i \mid t_i,\ b_k)$.

It is worth noting that we used a specific method of estimating $P(e \mid t)$ when e is unseen in training data, referred to as unknown words u afterwards. We rewrite the probability as

$$P(u \mid t) = \frac{P(t \mid u) P(u)}{P(t)}, \tag{12}$$

where $P(u)$ is a constant for all unknown words, and $P(t)$ can be estimated from training data. We now describe the way $P(t \mid u)$ is estimated. We split the training data into two folds. We construct a lexicon using the first fold, and treat all words

that occur in the second fold but are not stored in the lexicon as unknown words. A set of probabilities $P(t \mid u)$, each for a POS tag, is estimated on the second fold. We then used the second fold to construct the lexicon and estimated $P(t \mid u)$ on the first fold. The final probability is the average of the two estimates.

4.1.4 Decoding

We take two steps to identify the base NP sequence of a given English sentence **e**. In the first step, an A* search algorithm is applied for POS tagging. The top n **t** are retained, each is assigned a POS tagging probability P_t according to Equation (10). In the second step, for each **t**, a Viterbi algorithm is applied to search for the best base NP sequence. Every resulting **b** is assigned a base NP tagging probability P_b according to Equation (11). The final score of a base NP sequence is computed as $P_t^{\alpha} P_b$, where α is a POS tagging model weight (α = 2.4 in our experiments, which is tuned on Penn Treebank – see next subsection).

4.1.5 Evaluations

To test the performance of our approach, we used the section 20 of Penn Treebank as test data, sections 1-19 as training data to estimate the four model parameters in Equations (10) and (11), and sections 21-24 as held-out data to estimate unknown word probabilities in Equation (12) and tune other parameters such as the POS tagging model weight α and n in n-best **t** retained in the first step.

We retained 10-best POS tag sequences for each sentence in the first step because it achieves the best tradeoff of efficiency and accuracy in our experiments. We achieve 92.5% in precision and 93.8% in recall. The results are slightly better than most state-of-the-art results reported in Xun et al. [2000].

4.2 Complex NP Identification

Unlike base NP, there is not a widely accepted definition of complex NP. It is even more debatable in Chinese. In addition, a lot of complex NPs in English cannot be translated into Chinese as a unit. Therefore, with the help of a linguist, we selected 14 frequently used English NP patterns, which can be translated into Chinese as a unit. We define a NP pattern as a sequence of word classes. Each word class can be a terminal label (or word) such as *in*, *of* and *and*, or a non-terminal label such as base NP tag or POS tags defined in Marcus et al. [1993]. The top three most-frequently-used NP patterns are shown in Figure 4. Any sequence of words or base NPs corresponding to one of the patterns is identified as a complex NP.

4.3 NP Translation

We notice that many NPs that we identified are not stored in the bilingual dictionary. This section describes how we translate the NPs better than a word-by-word method.

Our NP translation model is motivated by two observations. First of all, we observe that most English NPs are translated to Chinese as NPs. For example, on a 60K-sentence-pair word-aligned English-Chinese bilingual corpus, we found that more than 80% of English NPs can be aligned to their translated Chinese NPs. Secondly, as pointed out in Koehn [2003], word selection can almost always be resolved depending solely upon the internal context of the NP.

Complex NP patterns	**Examples** (extracted from TREC-9 CLIR queries)
[Base NP *of* Base NP]	[the sales] of [Chinese ships]
[Base NP *in* Base NP]	[human rights violations] in [China]
[Base NP *and* Base NP]	[China 's Panda bear population] and [research organizations]

Figure 4. Examples of complex NP patterns

4.3.1 Principle

This section describes the concept of *translation template* which is the fundamental to the NP translation model.

We observed that there are some translation templates between English NP patterns and Chinese NP patterns. For example, a [NN-1 NN-2] English phrase is usually translated into a [NN-1 NN-2] sequence in Chinese, and a [NN-1 *of* NN-2] phrase is usually translated into a [NN-2 NN-1] sequence in Chinese. So for an English NP corresponding to such a pattern, even if its translation is not stored in the dictionary, we can still generate its possible translation according to corresponding translation template. For instance, we can derive the translation of the multi-word phrase "drug sale" as 毒品(drug)/买卖(sale), and the translation of "security committee of UN" as 联合国(UN)/安理会(security committee).

The concept of translation templates is very similar to alignment templates described in Och and Ney [2004]. Formally, a NP translation template, denoted by **z**, is a triple (*E*, *C*, *A*), which describes the alignment *A* between an English NP pattern *E* and a Chinese NP pattern *C*. The alignment *A* is represented as a set of pairs (*i*, *j*), indicating that the *i*-th English word class in *E* is connected to the *j*-th Chinese word class in *C*. Either *i* or *j* can be empty, denoted by ε, indicating that an English (or Chinese) word class is connected to no Chinese (or English) word class.

In our experiments, translation templates are extracted from a word-aligned bilingual corpus. We first used the NP identification method described above to tag POS, base NP, and complex NP for English sentences. Then, for each English NP pattern *E*, we extracted its translated Chinese NP patterns *C* and the alignment *A*. An example is shown in Figure 5, where (a) is an English sentence with each word marked by its POS tag and position and elements within […] are base NPs, or complex NPs; (b) is the aligned Chinese sentence that has been segmented into a sequence of words; (c) shows the word alignment between the English and Chinese sentences; and (d) shows three translation templates extracted respectively for two base NPs and for the whole phrase. Notice that the word positions in the alignments shown in (d) are those in *E* and *C* of each **z**. Also notice that translation templates can be recursively defined.

As mentioned earlier, the obtained NP translations do not always correspond to document indexes. If they do not, the segmentation process will break them down into several words. Even in this case, we can still benefit from the word selection in the process that solves partially the translation ambiguity problem.

(a)	[[The/DT/1 natural/JJ/2 language/NN/3 computing/NNP/4 group/NNP/5] at/IN/6 [Microsoft/NNP/7 Research/NNP/8 Aisa/NNP/9]] …
(b)	微软/1 亚洲/2 研究院/3 自然/4 语言/5 计算/6 组/7 …
(c)	(1, ε) (2, 4) (3, 5) (4, 6) (5, 7) (6, ε) (7, 1) (8, 3) (9, 2) …
(d)	$\mathbf{z}_1$ = (*E* = [*DT JJ NN NNP-1 NNP-2*], *C* = [*JJ NN NNP-1 NNP-2*], *A* = {(1, ε), (2,1), (3,2), (4,3)}) $\mathbf{z}_2$ = (*E* = [*NNP-1 NNP-2 NNP-3*], *C* = [*NNP-1 NNP-3 NNP-2*], *A* = {(1, 1), (2, 2), (3, 3)}) $\mathbf{z}_3$ = (*E* = [*Base-NP-1 at Base-NP-2*], *C* = [*Base-NP-2 Base-NP-1*], *A* = {(1, 2), (2, ε), (3, 1)})

Figure 5. NP translation templates patterns

4.3.2 NP Translation Model

We first describe the translation model under the framework of source-channel models, and then generalize it under the framework of linear models for parameter estimation.

Given an English NP $\mathbf{e} = \{e_1,\ldots,e_l\}$, we search among all possible translations the most probable Chinese translation $\mathbf{c}^* = \{c_1,\ldots,c_J\}$:

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} P(\mathbf{c}\,|\,\mathbf{e}) = \arg\max_{\mathbf{c}} P(\mathbf{c})P(\mathbf{e}\,|\,\mathbf{c}) \tag{13}$$

Here, $P(\mathbf{c})$ is the Chinese language model probability estimated via a trigram model as

$$P(\mathbf{c}) = P(c_1)P(c_2\,|\,c_1)\prod_{j=3\ldots J} P(c_j\,|\,c_{j-2}c_{j-1}) \tag{14}$$

$P(\mathbf{e}\,|\,\mathbf{c})$ is the translation probability. Formally, the NP translation template $\mathbf{z}$ is introduced as a hidden variable as

$$P(\mathbf{e}\,|\,\mathbf{c}) = \sum_{\mathbf{z}} P(\mathbf{z}\,|\,\mathbf{c})P(\mathbf{e}\,|\,\mathbf{z},\mathbf{c})\,. \tag{15}$$

Hence, there are two probabilities to be estimated. The probability $P(\mathbf{z}\,|\,\mathbf{c})$ to apply a translation template and the probability $P(\mathbf{e}\,|\,\mathbf{z}, \mathbf{c})$ to use a translation template for word selection.

First, we describe the way $P(\mathbf{z}\,|\,\mathbf{c})$ is estimated. Recall that $\mathbf{z} = (E, C, A)$, we call $\mathbf{z}$ *applicable* to $\mathbf{c}$ if $\mathbf{c}$ matches the NP pattern *C*. Let $C(\mathbf{c}, \mathbf{z})$ be the number of occurrences of $\mathbf{c}$ to which $\mathbf{z}$ is applicable and $C(\mathbf{c})$ be the number of occurrences of c in training data. $P(\mathbf{z}\,|\,\mathbf{c})$ is estimated as

$$P(\mathbf{z}\,|\,\mathbf{c}) = \frac{C(\mathbf{c},\mathbf{z})}{C(\mathbf{c})}\,. \tag{16}$$

Second, we describe the way $P(\mathbf{e}\,|\,\mathbf{z}, \mathbf{c})$ is estimated. We assume that the English words are translated independently. We then decompose the probability as

$$P(\mathbf{e} \mid \mathbf{z}, \mathbf{c}) = P(\mathbf{e} \mid (E, C, A), \mathbf{c}) = \prod_{(i,j) \in A} P(e_i \mid c_j) \cdot \tag{17}$$

Here, $P(e \mid c)$ is a translation probability estimated by relative frequencies:

$$P(e \mid c) = \frac{C(c, e)}{C(c)} \cdot \tag{18}$$

where $C(c, e)$ is the frequency that the word c is aligned to the word e, and $C(c)$ is the frequency of word c in training data.

Notice that the model of Equation (17) is a deficient model since the constraint $\Sigma_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{z}, \mathbf{c}) = 1$ does not hold, as discussed in Och and Ney [2004]. However it is not necessary to normalize it since we use the model as a feature function in translation, as will be described below. We also notice that it is possible to define an alignment in A at the level of base NP such as $\mathbf{z}_3$ in Figure 5 (d). As shown in Figure 5 (d), we assume that all alignments in A are pairs of word positions. Therefore, when we apply A in NP translation, we recursively map each alignment pair of base NP position to a set of pairs of word positions. For example, the pair (1, 2) in $\mathbf{z}_3$ in Figure 5 (d), which is an alignment between the positions of two base NP, can be mapped into a set of word position pairs using the alignment of $\mathbf{z}_2$.

Substituting Equation (15) into Equation (13), we have

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} P(\mathbf{c}) \sum_{\mathbf{z}} P(\mathbf{z} \mid \mathbf{c}) P(\mathbf{e} \mid \mathbf{z}, \mathbf{c}) \tag{19}$$

We see that the NP translation model consists of three component models: (1) the Chinese language model $P(\mathbf{c})$ of Equation (14), (2) the translation template selection model $P(\mathbf{z} \mid \mathbf{c})$ of Equation (16), and (3) the word selection model $P(\mathbf{e} \mid \mathbf{z}, \mathbf{c})$ of Equation (17). It should be noted that different component models are trained on different corpora. The dynamic value ranges of different component model probabilities can be so different (e.g., $P(\mathbf{e} \mid \mathbf{z}, \mathbf{c})$ of Equation (17) is not a *probability* but a *score*) that it is inappropriate to combine all these models through simple multiplication as in Equations (13) and (15). One way to balance these score quantities is to introduce for each component model a model weight λ to adjust the model score $P(.)$ to $P(.)^{\lambda}$. In our experiments, these weights are optimized so as to minimize the NP translation errors on training data under the framework of linear models[3].

It is worth noticing that the source-channel models are the rationale framework behind the NP translation model. Linear models are just another representation based on which we describe the optimization algorithm of model weights.

Now, let us reformulate the NP translation model under the framework of linear model [Duda et al. 2001]. It includes (1) a set of D feature functions that map the given English NP, the Chinese NP, and the translation templates into a real value, i.e., $f_d(\mathbf{e}, \mathbf{c}, \mathbf{z})$, for $d = 1 \ldots D$; and (2) a set of parameters, each for one feature, λ_i for $i = 1 \ldots D$. Then the decision rule of Equation (19) can be rewritten as

[3] The use of λ in this way to balance to balance incompatible likelihoods is commonly used in statistical speech recognition systems to balance acoustic and language model scores [Huang, Acero and Hon 2001].

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} \sum_{d=1}^{D} \lambda_d f_d(\mathbf{e}, \mathbf{c}, \mathbf{z}). \quad (20)$$

We can see that using the linear models, NP translation is viewed as a reranking problem. In our experiments, we used three feature functions. They are derived from the above three component models, respectively.

- **Chinese language model feature.** It is defined as the logarithm of the trigram model of Equation (14), i.e., $h_{LM}(\mathbf{c}) = \log P(\mathbf{c}) = \log P(c_1)P(c_2)\Pi_{i=3\ldots J}P(c_j \mid c_{j-2}\ c_{j-1})$.
- **Translation template selection model feature.** It is defined as the logarithm of $P(\mathbf{z} \mid \mathbf{c})$, i.e., $h_{TS}(\mathbf{z}, \mathbf{c}) = \log P(\mathbf{z} \mid \mathbf{c})$.
- **Word selection model feature.** It is defined as the logarithm of $P(\mathrm{e} \mid \mathrm{z}, \mathrm{c})$ of Equation (17), i.e., $h_{WS}(\mathbf{e}, \mathbf{z}, \mathbf{c}) = \log P(\mathbf{e} \mid (E, C, A), \mathbf{c}) = \log \Pi_{(i,j)\in A} P(e_i \mid c_j)$.

4.3.3 Training

For the three different component models, we used different training approaches. The Chinese language model probabilities are trained on a word-segmented Chinese text corpus consisting of approximately 1.6 billion Chinese characters. It contains documents of different domain, style, and time [Gao et al. 2002a]. The trigram probabilities are computed using MLE with Modified Absolute Discounting smoothing described in Gao et al. [2001a].

The translation template selection model is trained on a word-aligned bilingual corpus containing approximately 60K English-Chinese sentence pairs. Translation templates are first extracted automatically from the corpus, and then filtered by a linguist. The probability $P(\mathbf{z} \mid \mathbf{c})$ is then estimated according to Equation (11). For each Chinese NP pattern, there are 4.21 translation templates on average.

The word selection model probabilities are computed according to Equation (18) using the same word-aligned bilingual corpus containing 60K English-Chinese sentence pairs. To deal with the data sparseness problem, the probabilities $P_E(e \mid c)$ estimated via Equation (18) are linearly interpolated with the probabilities $P_D(e \mid c)$ derived from a bilingual dictionary by assuming a uniform distribution. That is, if c has n translations in the dictionary, each of them is assigned the same probability $P_D(e \mid c) = 1/n$.

The model weights λ, as shown in Equation (20), are estimated using an iterative procedure that is used for multi-dimensional function optimization [Press et al. 1992]. Assume that we can minimize NP translation errors with respect to one parameter λ using *line search*. The procedure works as follows: Take $\lambda_0, \lambda_1, \ldots, \lambda_N$ as a set of directions. Using line search, move along the first direction so that the number of NP translation errors on training data is minimized; then move from there along the second direction to the minimal error rate, and so on. Cycling through the whole set of directions as many times as necessary, until the error number stops decreasing. In our experiments, we found that the procedure can converge on different minima given different starting points. We thus perform the procedure multiple times, each from a different, random starting point, and pick the parameter setting that achieves the minimal errors. Note that this optimization approach is limited to a very small number of model parameters. Efficient algorithms for tuning a larger number of model parameters can be found in Och [2003] and Gao et al. [2005b].

4.3.4 Decoding

Given an English NP **e**, we take the following steps to search for the best Chinese translation.

1. **Template matching.** We find all translation templates that are applicable to the given English NP.
2. **Candidate generating.** For each of the translation templates, we determine a set of Chinese words for each English word position. The set of Chinese words are all possible translations of the English word, stored in a bilingual dictionary. We then form a lattice for each **e**.
3. **Searching.** For each lattice, we use an A* decoder to find top n translation candidates according to Equation (19) where only two features, h_{LM} and h_{WS}, are used.
4. **Fusion and reranking.** We fuse all retained translation candidates, and rerank them according to Equation (19), where all features are applied.

Notice that Equation (20) does not take into account the sum on **z** in Equation (19), because considering the sum in decoding directly is computationally expensive. We therefore approximate the sum during decoding in two steps: First, for each **z**, we find the best translation, as shown in steps 2 and 3 in the above algorithm. Second, we select the translation among all retained best translations according to Equation (20), as described in step 4.

5. DEPENDENCY TRANSLATION MODEL

The NP translation model is limited to NPs contiguous in both source and target languages. The dependency translation model to be described aims at eliminating the limitation to some degree by taking into account syntactic dependencies in translation selection. We can therefore translate more precisely discontinuous phrases, including dependency triples, such as verb-object and adjective-noun, regardless of the number of intervening words.

Similar to that of the NP translation model, the dependency translation model is also built on the basis of two hypotheses. First, dependencies have the best cohesion properties across languages [Fox 2002]. That is, dependency representation usually remains in the translations, and an ideal query translation should contain the same syntactic dependences as in the original query. Second, word selection can mostly be resolved via the internal context of the dependency. Thus, syntactic dependencies also provide an additional criterion to the earlier cohesion measure used in the co-occurrence model described in Section 3.4: A good translation word should not only co-occur with other translation words, but also have the required syntactic dependency relations with them.

5.1 Principle

A dependency, denoted by a triple (w_1, r, w_2), called a *dependency triple* afterwards, represents syntactic dependency relation r between two words w_1 and w_2, such as verb-object and subject-verb. Figure 6 shows an English sentence and the dependency triples that are extracted from it. For example, "dog" is the subject of the verb "barked". It is our observation that there is a strong correspondence in

(a)	The brown dog on the hill barked last night.
(b)	(dog, subject-verb, barked) (the, det-noun, dog) (brown, adjective-noun, dog)…

Figure 6. Examples of an English sentence (a) and its dependency triples (b).

dependency relations in the translation between English and Chinese, despite the great differences between the two languages. For example, a subject-verb relation in English, e.g. (dog, subject-verb, barking), is usually translated into the same subject-verb relation in Chinese, e.g. (狗, subject-verb, 吠). This suggests that similar to NP translation, there also exist a translation template between English dependency triples and Chinese ones.

Unlike NP translation templates, there is only one translation template: an English dependency triple $\mathbf{e}_t = (e_1, r_e, e_2)$ is most likely to be translated to a Chinese dependency triple $\mathbf{c}_t = (c_1, r_c, c_2)$, where c_1 and c_2 are the Chinese translations of the English terms e_1 and e_2, respectively, and r_c is the Chinese counterpart of r_e.

Among all the dependency relations, we only consider the following four types that can be detected precisely using our parser and cannot be handled by the NP translation model[4]: (1) subject-verb, (2) verb-object, (3) adjective-noun, and (4) adverb-verb. That is, $r \in$ {subject-verb, verb-object, adjective-noun, adverb-verb}.

To prove the validity of the above translation template, we perform the following test. We used the abovementioned word-aligned bilingual corpus containing 60K English-Chinese sentence pairs. The corpus was first parsed using an English and Chinese parser NLPWIN, a broad-coverage rule-based parser developed at Microsoft Research able to produce syntactic analysis at varying levels of depth [Heidorn 2000]. For the purposes of our experiments, we used a dependency tree output with unstemmed surface words. The four types of dependency triples were then extracted from the trees. We analyzed the correspondence on dependency relations between Chinese and English. The results are shown in Table I. As we can see, more than 80% of dependency relations of subject-verb, adjective-noun, and adverb-verb have one-to-one mappings between English and Chinese, while the mapping rate of verb-object is approximately 65%.

Further analyses showed that the mapping errors of verb-object occur in the following situations: (1) a single English verb maps a Chinese dependency triple (e.g. read → 读[read] verb | 书[book] object), or (2) an English verb-prep-object sequence maps a Chinese verb-object sequence (e.g. change-to-currency → 用[use] verb | 货币[currency] object). As the first case is not a dependency translation, and will not affect the dependency model, it is ignored. The second problem is quite common. In fact, the combination of verb-preposition in English is very often translated into a single verb in Chinese. If we consider such a combination in English as "verb", the mapping rate of verb-object is increased to more than 80% (see Table I – (*) case). This is the way we will use to map verb-object dependencies between Chinese and English.

[4] We might obtain better performance using more dependency relations. We leave it to future work

Table I: Dependency relation correspondence between Chinese and English

Dependency	subject-verb	adjective-noun	adverb-verb	verb-object	verb-object (*)
Mapping Rate	81.2%	81.0%	80.9%	64.8%	80.7%

5.2 Dependency Translation Model

Given an English dependency triple $\mathbf{e}_t = (e_1, r_e, e_2)$, and a set of its candidates of Chinese dependency triple translation, the best Chinese dependency triple $\mathbf{c}_t = (c_1, r_c, c_2)$ is the one that maximizes the following equation

$$\mathbf{c}_t^* = \arg\max_{\mathbf{c}} P(\mathbf{c}_t \mid \mathbf{e}_t) = \arg\max_{\mathbf{c}} P(\mathbf{c}_t) P(\mathbf{e}_t \mid \mathbf{c}_t) \cdot \quad (21)$$

Here, $P(\mathbf{c}_t)$ is the *a priori* probability of words of the translated Chinese dependency triple. It can be estimated using MLE as

$$P(\mathbf{c}_t) = \frac{C(\mathbf{c}_t)}{N}, \quad (22)$$

where $C(\mathbf{c}_t)$ is the number of occurrences of $\mathbf{c}_t$ in the collection, and N is the number of all dependency triples.

$P(\mathbf{e}_t \mid \mathbf{c}_t)$ is the translation probability. We assume that (1) $\mathbf{e}_t$ and $\mathbf{c}_t$ can be translated with each other only if they have the same type of dependency relation, i.e., $r_e = r_c$; (2) words in a dependency triple are translated independently. We therefore decompose the probability $P(\mathbf{e}_t \mid \mathbf{c}_t)$ as

$$P(\mathbf{e}_t \mid \mathbf{c}_t) = P(e_1 \mid c_1) P(e_2 \mid c_2) \delta(r_e, r_c), \quad (23)$$

where $\delta(r_e, r_c) = 1$ if $r_e = r_c$ and 0 otherwise.

$P(e \mid c)$ is a word translation probability, which could be estimated on word-aligned bilingual corpus using Equation (18). However, we observe that within a dependency triple (w_1, r, w_2), the translation selection of a word (e.g., w_1) largely depends on the other word w_2 and the relation r. For example, the word "bear" in a dependency triple (bear, verb-object, child) is translated to 怀, while it is most likely to be translated to 忍受 as an individual word (if the translation probability is trained directly on a word-aligned corpus or the translation is obtained via dictionary look up). This suggests that translation probabilities in Equation (23) are better trained on a set of aligned bilingual dependency triple pairs. Unfortunately, it is difficult to obtain such a corpus in large quantity. Therefore, in our model, instead of using a translation probability we assume that the likelihood of c to be translated to e can be measured by their semantic similarity, denoted by $sim(e, c)$ (see section 5.2.1 for its calculation). Notice that e and c are not necessary to be a translation pair stored in a dictionary but just a pair of *cross-lingual synonyms* derived via their semantic similarity e.g., 怀 is not a translation of "bear" defined in a dictionary, but a *synonym*. Since our goal is to obtain good IR results, such cross-lingual synonyms may solve the term mismatch problem and boost the CLIR performance, playing a similar role of synonym-based query expansion in monolingual IR, as described in Section 5.5.

Now, we see from Equations (21) and (23) that the likelihood of $\mathbf{e}_t$ to be translated to $\mathbf{c}_t$, assuming that $r_e = r_c$, can be scored via two factors: (1) $P(\mathbf{c}_t)$ of Equation (22), and (2) $sim(e, c)$. Similar to the NP translation model described in Section 4.3, we define a feature function for each type of factors, and combine them under the framework of linear models as shown in Equation (20). In the dependency translation model, we used two type of features.

- **Chinese language model feature.** It is defined as the logarithm of the model of Equation (21), i.e., $h_{LM}(\mathbf{c}_t) = \log P(\mathbf{c}_t)$.
- **Cross-lingual word similarity feature.** It is defined as the similarity between two words, i.e., i.e., $h_{WS}(\mathbf{e}_t, \mathbf{c}_t) = sim(e, c)$. Since there are 4 dependency relations, each with 2 words, there are in total 8 types of word pair. We define 8 feature functions, each for one type of word pair, such as the similarity between a verb pair in a verb-object dependency.

Similar to the NP translation model, the linear model parameters λs, which are used to combine the above two features, are estimated using the iterative procedure for multi-dimensional function optimization (Section 4.3.3).

5.2.1 Cross-lingual Word Similarity

This section describes the way $sim(e, c)$ is computed. Lin [1997; 1998] presents a method of computing the similarity between two words in the same language on the basis of *dependency context*. Zhou et al. [2001] extended the word similarity measurement of Lin's to the cross-lingual case. The description below is adapted from Lin [1998] and Zhou et al. [2001].

Let us first discuss the way of computing $sim(w_1, w_2)$ for w_1 and w_2 in the same language. The basic idea is to consider all dependencies including a word w as its *dependency context*, denoted by $T(w)$. For example, in Figure 5, the dependency context of the word "dog" consists of three dependency triples, i.e., $T(\text{"dog"})$ = { (*, subject-verb, barked), (the, det-noun, *), (brown, adjective-noun, *) }, where the wildcard symbol * denotes any word. It is assumed that two words are likely to have similar semantic meanings if their dependency contexts are identical. For simplicity, in what follows, we use (r,w) to denote either $(*, r, w)$ or $(w, r, *)$.

Using an information-theoretic definition, $sim(w_1,w_2)$ is measured by the ratio between the amount of information needed to describe the commonality of w_1 and w_2, denoted by $I(common(w_1,w_2))$, and the information needed to fully describe what w_1 and w_2 are, denoted by $I(describe(w_1,w_2))$:

$$sim(w_1, w_2) = \frac{I(common(w_1, w_2))}{I(describe(w_1, w_2))}, \tag{24}$$

We assume that a dependency triple (w_1, r, w_2) is generated via three steps:

- A: a randomly selected word is w_1;
- B: a randomly selected dependency type is r;
- C: a randomly selected word is w_2.

We assume that A and C are conditionally independent given B. Let $I(w_1, r, w_2)$ be the amount of information needed to describe a dependency triple (w_1, r, w_2). Its value can be computed as

$$I(w_1, r, w_2) = \log \frac{P(A, B, C)}{P(B)P(A \mid B)P(C \mid B)}. \quad (25)$$

All the probabilities in the right hand side of Equation (25) can be computed using MLE as shown in Equations (26) to (29), where $C(x)$ is the occurrence of x in training data, and the wildcard symbol * denotes any word or dependency type.

$$P(B) = \frac{C(*, r, *)}{C(*, *, *)}, \quad (26)$$

$$P(A \mid B) = \frac{C(w_1, r, *)}{C(*, r, *)}, \quad (27)$$

$$P(C \mid B) = \frac{C(*, r, w_2)}{C(*, r, *)}, \quad (28)$$

$$P(A, B, C) = \frac{C(w_1, r, w_2)}{C(*, *, *)}. \quad (29)$$

Then, substituting Equations (26) to (29) into (25), we have

$$I(w_1, r, w_2) = \log \frac{C(w_1, r, w_2) \times C(*, r, *)}{C(w_1, r, *) \times C(*, r, w_2)}. \quad (30)$$

Next, we assume that information can be additive, then $I(common(w_1, w_2))$ is calculated as the sum of the information contained in common dependency triples belonging to both dependency context sets $T(w_1)$ and $T(w_2)$:

$$I(common(w_1, w_2)) = \sum_{(r,w) \in T(w_1) \cap T(w_2)} (I(w_1, r, w) + I(w_2, r, w)). \quad (31)$$

Similarly, $I(describe(w_1, w_2))$ is calculated as the sum of the information contained in dependency triples belonging to either dependency context sets $T(w_1)$ or $T(w_2)$:

$$I(describe(w_1, w_2)) = \sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w). \quad (32)$$

Now, let us discuss how the similarity measurement of Equation (24) is extended to measure the cross-lingual word similarity $sim(e, c)$. Simply, for an English dependency context (r_e, e), a Chinese dependency context (r_c, c) is similar if $r_c = r_e$, and c and e form a translation pair in a bilingual dictionary. In this case, we call (r_c, c) is a possible translation of (r_e, e), vice versa.

The probability of such a translation may be estimated as $P(c \mid e)\delta(r_c, r_e)$, where $\delta(r_c, r_e) = 1$ if $r_c = r_e$; 0 otherwise. $P(c \mid e)$ is the translation probability which can be estimated from bilingual corpus as Equation (18).

So, the cross-lingual commonality $I(common(e, c))$ is adapted from Equation (31) as

$$I(common(e, c)) = \sum_{(r_e, e') \in T(e)} I(e, r, e') P(e' \mid c') \delta(r_c, r_e) + \sum_{(r_e, c') \in T(c)} I(c, r, c') P(c' \mid e') \delta(r_c, r_e) \quad (33)$$

Table II: Statistics of the extracted dependency triples

Language	subject-verb	adjective-noun	adverb-verb	verb-object
Chinese	26,773,214	14,707,246	10,191,300	22,259,701
English	6,475,461	2,026,177	741,719	6,558,566

Similarly, the descriptions of e and c are adapted from Equation (32) as

$$I(describe(e,c)) = \sum_{(r,e')\in T(e)} I(e,r,e') + \sum_{(r,c')\in T(c)} I(c,r,c') \quad (34)$$

5.3 Training

There are two types of parameters in the dependency translation model, $sim(e, c)$ and $P(\mathbf{c}_t)$.

One advantage of our model is that $sim(e, c)$ can be trained on unrelated English and Chinese corpora. In our experiments, we used an English text containing articles published in the Wall Street Journal from 1987 to 1992, which amount to 750MB, and a Chinese text corpus containing articles published in the People's Daily from 1980 to 1998, which amount to 1200MB. We estimated $sim(e, c)$ in the following steps. First, NLPWIN is used to extract triples in both corpora. Table II shows the number of different types of dependency triples extracted. Then, for each e or c, we construct its dependency context. Finally, in the runtime, we computed $sim(e, c)$ using the method described in Section 5.2.1. Notice that there is a risk that unrelated dependency triples in Chinese and English can be connected since e and c might not be translation pair but synonyms. However, as we will show in Section 5.4, the gain outweighs the loss significantly in our translation evaluation experiments.

$P(\mathbf{c}_t)$ is trained via MLE, as shown in Equation (18), on the Chinese dependency triple corpus, extracted from the Chinese text corpus by NLPWIN as described above. In addition, we used Good-Turning smoothing, as described in Section 3.3, to deal with the sparse data problem.

5.4 Decoding

Given an English sentence, we translate all its dependency triples to Chinese using the dependency translation model in the following.

1. **Dependency triple detection.** We use NLPWIN to detect all dependency triples of the four types in a given English sentence.
2. **Candidate generation.** For each of the dependency triple $\mathbf{e}_t = (e_1, r_e, e_2)$ we generate all Chinese translation candidates $\mathbf{c}_t = (c_1, r_c, c_2)$, where $r_e = r_c$, $sim(e_1, c_1) > \theta_1$, and $sim(e_2, c_2) > \theta_2$. θ_1 are θ_2 thresholds whose values are determined empirically.
3. **Candidate ranking.** All translation candidates are ranked using the linear models of Equation (20), where two types of feature functions are used. As described in Section 5.2, they are (1) $h_{LM}(\mathbf{c}_t)$, and (2) a set of $h_{WS}(\mathbf{e}_t, \mathbf{c}_t)$.

Table III: Verb-object dependency triple translation results

	T1	T2	T3	T4
Method A	42.4%	44.0%	54.1%	54.1%
Method B	55.0%	67.3%	62.3%	67.6%
Method C	74.2%	80.9%	73.3%	81.0%

5.5 Evaluating Dependency Translation

This section discusses evaluation results of dependency translation. We first present verb-object dependency triple translation results. Then, we show that our method of computing $sim(e, c)$, instead of word-to-word translation probability, takes into account semantic similarities between words and can translate dependency triples which contain words whose translations can be neither found in a dictionary nor learnt on word-aligned bilingual corpora.

5.5.1 Verb-Object Dependency Triple Translation Results

Verb-object dependency triples represent the most serious translation ambiguities among the four dependency types. Therefore, we focus on them in our translation evaluations. We performed the evaluation on the Chinese and English verb-object dependency triple sets shown in Table II. We used 80% of the data as training data. For the remaining 20% dependency triples, we constructed the following four test sets, where a correct translation for each dependency triple is generated manually.

- **T1**: The set contains 1000 dependency triples with high frequent verbs.
- **T2**: The set contains 275 dependency triples with low frequent verbs.
- **T3**: The set contains top 1000 high frequent dependency triples.
- **T4**: The set contains 700 low frequent dependency triples.

The translation performance is measured by accuracy, defined as

$$\frac{\#\text{ of correct translated dependency triples}}{\#\text{ of dependency triples in a test set}}.$$

The following three translation methods are compared:

- **Method A**: Each English word in a dependency triple is translated to the most frequent translation word among those stored in the dictionary.
- **Method B**: For each English word in a dependency triple, translation selection is achieved by the decaying co-occurrence model, described in Section 3.
- **Method C**: The dependency translation model is used.

The results are shown in Table III. We can see that Method C, which uses the dependency translation model, achieves the most precise translations in all four test sets. The results also confirm that the risk of relating unrelated triples by using non-parallel corpora for training is small. We also notice that the improvements achieved by the dependency translation model accuracy are consistent among different test sets, demonstrating that the model is robust against the frequency of dependency triples and verbs.

5.5.2 Impact of Using Cross-Lingual Semantic Similarity

This section discusses the impact of introducing the cross-lingual semantic word similarity, i.e., $sim(e, c)$ described in Section 5.2.1, on dependency translation.

We can view the dependency translation model as a method of expanding the candidate translation set of an English word e by adding all semantically similar Chinese words according to $sim(e, c)$, which are not stored in a bilingual dictionary. In comparison, we also used an English synonym dictionary, which is constructed from the LDC bilingual dictionaries as follows: For each English term e and each of its Chinese translation c, we consider all the English translations e' of c as synonyms of e.

We compared three methods of translating an English dependency triple $\mathbf{e}_t$. For all the three methods, we used the translation procedure as described in 5.4. The differences among the three methods are (1) the way the translation candidates are generated or expanded and (2) how $sim(e, c)$ is computed. Specifically,

- **Method A**. For each English word e in a dependency triple, we consider all its possible translations stored in the LDC bilingual dictionary. The resulting candidate set is denoted by $TD(e)$. We assume that every $c \in TD(e)$ have the same value, i.e., $sim(e, c) = 1/|TD(e)|$ for all $c \in TD(e)$.
- **Method B**: For each English word e in a dependency triple, we consider all its possible translations in $TD(e)$ and the candidate set $TS(e)$ that contains all possible translations of all the synonyms of e according to the synonym dictionary mentioned above. For every c that only belongs to $TD(e)$, we have $sim(e, c) = \alpha/|TD(e)|$. For every c that only belongs to $TS(e)$, we have $sim(e, c) = \beta/|TS(e)|$. For every c that only belongs to both $TD(e)$ and $TS(e)$, we have $sim(e, c) = \alpha/|TD(e)| + \beta/|TS(e)|$. α =0.6 and β = 0.4 in our experiments.
- **Method C**: we use the dependency translation model to generate and rank translation candidates, as described in Section 5.4.

We performed the comparison experiments on T1. The results are shown in Table IV. We see that the use of synonym dictionary does not bring any benefit because it introduces a lot of noise along with some correct expansion of the set of translation candidates. The dependency translation model achieves the best results. The use of cross-lingual, semantically similar words in dependency translation eliminates to some degree the limited coverage of bilingual dictionaries. As an example, for the word "bear" in a dependency triple (bear, verb-object, child), the correct translation 怀 is not stored in the dictionary. The use of the dependency translation model (i.e., Method C) leads to a correct translation. We found that the dependency triple (怀, verb-object, 孩子) and (bear, verb-object, child) are very frequent in the corpora, so both I(怀, verb-object, 孩子) and I(bear, verb-object, child) are strong. So globally, the commonality between 怀 and "bear" is high, and as a result, sim(怀|bear) is also very high.

Unfortunately, there are also some negative examples of using the dependency translation model, as shown in Figure 7. We found that the wrong translations usually occur when the combinations of wrongly expanded translation candidates are frequent ones in the corpus, thus cannot be filtered out using the translation ranking model. The Chinese dependency triples in the two negative examples of Figure 7 are very frequent ones. In these cases, even if the component words separately are strongly related to the original English words, their combination

Table IV: Results of dependency translation with different method of translation candidate generation

Methods	Accuracy
A: Using dictionary only	74.2%
B: Using synonyms	69.8%
C: Using dependency translation model	80.3%

	English dependencies	Method A/B	Method C
Positive examples	bear | child	忍受|孩子 (suffer from child)	怀|孩子 (bear child)
	break | silence	打碎| 沉默 (smash silence)	打破|沉默 (break silence)
Negative examples	build | road	修建|公路 (build road)	制定|办法 (set up a method)
	make | sound	发出|声音 (make sound)	发表|讲话 (give a speech)

Figure 7: Translation examples using dependency translation model (Methods A and B produce the same result on this example)

corresponds to a meaning different from the original one. Our model is unable to deal with this problem. We leave it to future work.

6. CLIR EXPERIMENTS

6.1 Settings

We evaluate the three proposed query translation models on CLIR experiments on TREC Chinese collections. The TREC-9 collection contains articles published in Hong Kong Commercial Daily, Hong Kong Daily News, and Takungpao. They amount to 260MB. A set of 25 English queries (with translated Chinese queries) has been set up and evaluated by people at NIST (National Institute of Standards and Technology). The TREC-5&6 corpus contains articles published in the People's Daily from 1991 to 1993, and a part of the news released by the Xinhua News Agency in 1994 and 1995. A set of 54 English queries (with translated Chinese queries) has been set up and evaluated by people at NIST.

All Chinese texts, articles and translated queries, are word-segmented using the Chinese word segmentation system MSRSeg [Gao et al. 2005a]. The system also identifies named entities of various types. Then, stop words are removed. Each of the TREC queries has three fields: title, description, and narratives. In our experiments, we used two versions of queries, *short queries* that contain titles only and *long queries* that contain all the three fields.

The bilingual dictionary we used is a combination of three human compiled bilingual lexicons, including the LDC English-Chinese dictionary and a bilingual lexicon generated from a parallel bilingual corpus automatically. The combined dictionary contains 401,477 English entries, including 109,841 words, and 291,636 phrases. The use of the combined dictionary is motivated by previous studies [e.g., Gao et al. 2000; Xu and Weieschedel 2000], which showed that larger lexicon resource improves CLIR performance significantly.

The Okapi system with BM2500 weighting [Robertson and Walker 2000] is used as the basic retrieval system. The main evaluation metric is interpolated 11-point average precision. Statistical t-test and query-by-query analysis are also employed. To decide whether the improvement by method X over method Y is significant, the t-test calculates a p-value based on the performance data of X and Y. The smaller the p-value, the more significant is the improvement. In our experiments reported below we conclude that the improvement is statistically significant if p-value is less than 0.05.

6.2 Main Results

The main results are shown in Tables V to VII (i.e., average precisions) and Figure 8 (i.e., precision-recall curves). To investigate the effectiveness of our models for query translation, the following three baseline methods are compared.

ML (Monolingual). We retrieve documents using the manually translated Chinese queries provided with the TREC collections. Its performance has been considered as an upper-bound of CLIR because the translation process always introduces translation errors. However, recent studies show that CLIR results can be better than monolingual retrieval results, which is also observed in our experiments.

ST (Simple Translation) We retrieve documents using query translation obtained from the bilingual dictionary. Phrase entries in the dictionary are first used for phrase matching and translation, and then the remaining words are translated by their translations stored in the dictionary. For each phrase/word with multiple translations stored in the dictionary, we only take the first translation, which is supposed to be the most frequently used translation. We could take more translations for each phrase/words, but our pilot experiments show that it hurts the performance in most cases.

BST (Best-Sense Translation) We retrieve documents using translation words selected manually from the dictionary, one translation per word, by a native Chinese speaker. If none of the translations stored in the dictionary is correct, the first one is chosen. This method reflects the upper bound performance using the dictionary.

COTM (co-occurrence translation model), **NPTM** (NP translation model) and **DPTM** (dependency translation model) are the three query translation models described in Sections 3 to 5, respectively. Notice that since NLPWIN, which is used to detect dependency triples, is a rule-based parser and performs well only when the input word sequence is a grammatical sentence, we only tested DPTM on long queries. The NP detector as described in Sections 4.1 and 4.2 is statistical in nature, and can handle arbitrary word sequences, so we tested NPTM using both long and short queries.

6.3 Discussion

The experimental results shown in Tables V to VII and Figure 8 give rise to the following observations.

6.3.1 Impact of COTM

We see that COTM brings statistically significant improvements over ST for long queries, as shown in Table VI (Row 4) and Table VII (Row 4). However, its improvement over ST for short queries is marginal. This is expected because

Table V: 11-point average precision for short queries on TREC-9 dataset
(* indicates that the improvement is statistically significant.)

	Translation Model	Average Precision	% of ML	Impr. over ST
1	**ML**	0.2956		
2	**ST**	0.1398	44.28%	
3	**BST**	0.1833	62.01%	40.03%*
4	**COTM**	0.1399	47.33%	6.88%
5	**NPTM**	0.2345	79.33%	79.14%*
6	**COTM + NPTM**	0.2708	91.61%	106.88%*

Table VI: 11-point average precision for long queries on TREC-9 dataset
(* indicates that the improvement is statistically significant.)

	Translation Model	Average Precision	% of ML	Impr. over ST
1	**ML**	0.3179		
2	**ST**	0.2003	62.99%	
3	**BST**	0.2924	91.96%	46.00%*
4	**COTM**	0.2657	83.58%	32.69%*
5	**NPTM**	0.2562	80.58%	27.93%*
6	**DPTM**	0.2160	67.94%	7.86%
7	**NPTM + NPTM**	0.3093	97.28%	54.44%*
8	**COTM + DPTM**	0.2705	85.09%	35.09%*
9	**COTM + NPTM + DPTM**	0.3303	103.88%	64.92%*

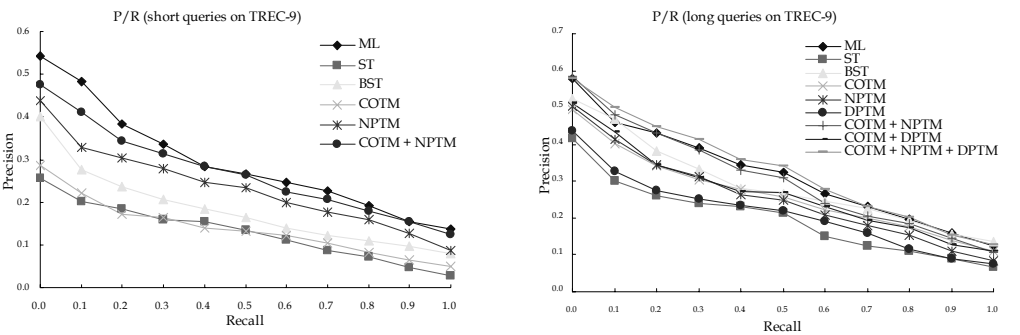


Figure 8: Precision-Recall curves for short and long queries on TREC-9 dataset.

Table VII: 11-point average precision for long queries on TREC-5&6 dataset
(* indicates that the improvement is statistically significant.)

	Translation Model	Average Precision	% of ML	Impr. over ST
1	**ML**	0.5184		
2	**ST**	0.2811	54.22%	
3	**BST**	0.3906	75.35%	38.95%*
4	**COTM**	0.3391	65.41%	20.63%*
5	**COTM + NPTM**	0.3894	75.12%	38.53%*
6	**COTM + NPTM + DPTM**	0.4541	87.60%	61.54%*

COTM resolves translation ambiguities with resort to context terms. Long queries contain much richer contextual information than short queries.

6.3.2 Impact of NPTM

We observe that unlike COTM, NPTM achieves substantial improvements over ST for both long and short queries, as shown in Tables V to VII. We notice that NPTM even outperforms BST for short queries, as shown in Rows 3 and 5 in Table V. It is thus interesting to compare the phrase translation results using NPTM and with that using dictionary look-up (Rows 2 and 3 in Table V). A further analysis shows that by using NP identification and translation, we obtained better translations. For example, in TREC-9 short query retrieval, only 11 multi-word phrases out of 25 queries are stored in the dictionary, and translated as a phrase, whilst using NPTM, 26 NPs are identified and translated. It thus leads to a significant improvement over BST. However, NPTM described in Section 4 also has some limitations as described below. We divide NP into two types: compositional NP and non-compositional NP.

A compositional NP refers to a phrase whose translation can be assembled by translations of words within the phrase, such as "computer hacker" (电脑黑客), "public key" (公共密钥), and "environmental protection laws" (环境保护法), etc. Generally speaking, NPTM is good for compositional NP translation. However, it failed to translate correctly some domain-specific NPs. For example, "stealth technology" (隐秘技术) and "stealth countermeasure" (反隐秘技术) in #59, and "synthetic aperture radar" (合成孔径雷达) in #66 correspond to special terminology in Chinese and they are not translated correctly by NPTM.

A non-compositional NP is a phrase whose translation cannot be assembled by translations of its component words. Our method NPTM is unable to deal with the translation of non-compositional NPs. Examples include "three-links" (三通) in #65, "vehicle fatalities" (车祸) in #68, "most-favored nation" (最惠国), and "World Conference on Women" (世妇会), etc. A large portion of non-compositional NPs in queries are political abbreviations. If these NPs are not stored in the dictionary, they are most likely to be translated incorrectly. This indicates that the coverage of the dictionary is still an important problem to be solved to improve the performance of CLIR.

6.3.3 Impact of DPTM.

We find that the use of DPTM leads to an effectiveness well below that with the other two models, COTM and NPTM. For example, as shown in Table VI (Rows 2 and 6), the improvement of DPTM over ST is not statistically significant. This is however expectable because dependency triples have a much lower coverage than the other models. Consider TREC-9 long query retrieval, only a few triples from 11 queries out of 25 have been translated by DPTM, while all the other words are translated by the first translation word in the dictionary. So this “counter-performance” is not surprising. Figure 9 shows a closer view on the 11 queries. From the 11 queries, NLPWIN extracted 52 dependency triples which appear at least 5 times in the corpus. The minimal occurrences of 5 is set due to the fact that many low frequency dependency triples are in fact noise. The 52 triples include 12 verb-object dependency triples, 8 sub-verb triples, 32 adjective-noun triples and no

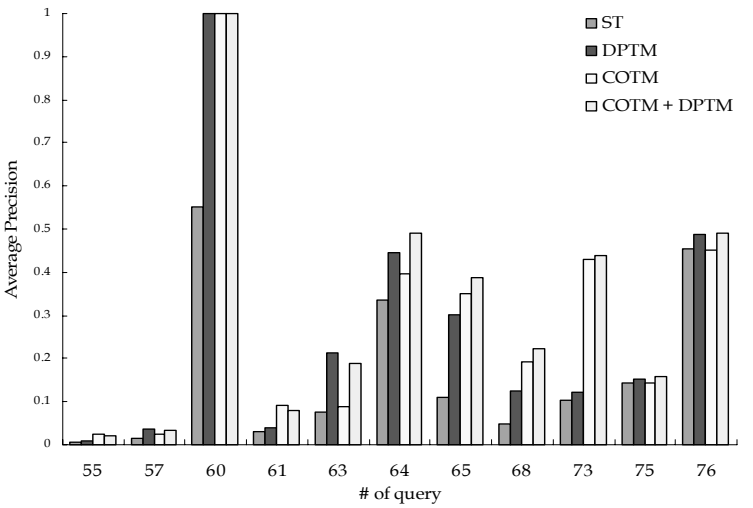


Figure 9: Average precision for 11 long queries on TREC-9 dataset

adv-verb triple. As shown in Figure 9, for these queries, the dependency triple translation has positive impact on the methods of ST and COTM for almost all the 11 queries, except for #61. In the case of query #61, only one translation word differs: "(coal) consumption" is translated to 消耗 by COTM, and to 消费 by DPTM. Both translations are correct, but the first translation word is often used for industrial consumption (which is the case for this query) whereas the second is often used for consumption of particular consumers. For all the 11 queries, globally, the triple method makes a statistically significant improvement of 56% over ST, and 10% over COTM.

A further analysis shows that DPTM is able to assemble translation words correctly in a dependency triple because the triples can capture the syntactic dependency between not only sequential words (i.e. phrases) but also non-sequential words in a sentence as the (computer-virus, subject-verb, originate) triple in query #64 shown in Figure 10. All the translations in Figure 10 are correct.

We can compare some of examples of Figure 10 with those translated by COTM. For Queries #63 and #64, COTM gives some different but correct translations for the following words: "develop" is translated to 发展, and "originate" to 发源 (v.s. to 开发 and 起源 by DPTM). However, the word "hacker" in Query #65 is wrongly translated to 恶作剧者 (joker), and "charge" to 保护 (protect) by COTM. For these cases, DPTM generates the correct translations. These examples show that the translations with dependency triples can successfully correct some of the incorrect selections of translation words.

6.3.4 Impact of Combining Translation Models

Previous work [e.g., Xu and Weischedel 2000] showed that if multiple translations of a query term were accepted in query translation, it is possible to obtain better performance of cross-language retrieval because of the query expansion effect. Therefore, we combine models using the parallel combining approach, as

Q#	Sentence, the extracted triples and translations
63	What new or renewable energy sources are being developed in China?
	(energy, adjective-noun, renewable) → 可再生 | 能源 (energy, adjective-noun, new) → 新 | 能源 (develop, verb-object, energy) → 开发 | 能源
64	Have any computer viruses been discovered to have originated in Asia?
	(discover, verb-object, originate) → 发现 | 起源 (computer-virus, subject-verb, originate) → 计算机病毒 | 起源
65	Have any computer hackers been charged with crimes in Asia?
	(hacker, adjective-noun, computer) → 计算机 | 黑客 (charge, verb-object, hacker) → 控告 | 黑客

Figure 10: Examples of dependency triples and their translations

described in Section 3.5. That is, whenever we combine these query translation models, we combine their translation results directly. That is, we combine linearly the three (or two) sets of translation queries obtained by the three (or any two of the three) translation models, respectively.

As expected, the combined models always perform better than each component model to be combined. Interestingly, for some queries, their CLIR results are even better than their monolingual retrieval results. Some examples we observed in TREC-9 long query experiments are: "public key" in query #68 is translated to 公共密钥 as well as 公共密码, "Olympics" in query #71 to 奥林匹克 (Olympic) and 奥运会 (Olympic games), and "panda bear" in query #76 to 大熊猫 and 大猫熊, etc. All these terms are commonly used translations. Thus, as a result, by combining all the three query translation models, we get the best CLIR results, which are very close to, or even better than, the monolingual retrieval results, as shown in Tables V to VII.

7. RELATED WORK AND DISCUSSION

Bilingual dictionaries have been used in several CLIR experiments. However, previous work showed that English-Chinese CLIR using simple dictionary translation yields a performance lower than 60% of the monolingual performance, as described in Kowk [2000]. The main problems observed are: (1) the dictionary may have a poor coverage; and (2) it is difficult to select the correct translation of a word among all the translations provided by the dictionary.

For the first problem, much effort has been spent on collecting larger lexical resources either manually or automatically [e.g., Kowk 2000; Nie et al. 1999; Xu and Weischedel 2000; Zhang and Vines 2004]. The coverage of the dictionary can be increased to some extent.

The second problem is also called *translation selection or disambiguation* problem. This is the focus of this paper. For each of the three query translation models described above, there has been large amount of related work. In what follows, we present a brief review. The study reported in this paper is an extension of our previous research reported in Gao et al. [2000; 2001b; 2002b], which will not be reviewed in this section.

7.1 Remarks on the Co-occurrence Model

Co-occurrence information has been utilized by several recent studies [e.g., Ballesteros and Croft 1998; Bian and Chen 1998; Fung et al. 1999; Jang et al. 1999; Peters and Picchi 1996; Mandala et al. 1999; Liu et al. 2005] to deal with the selection of the correct translation terms from a bilingual dictionary. A term similarity is determined by the mutual information (or its variants) between terms. Then the most similar translation term among those in the dictionary is selected. Our co-occurrence model is an extension of the previous methods in that we incorporate a decaying factor that decreases the mutual information when the distance between the terms increases. A similar idea has been applied successfully to statistical language modeling [Clarkson and Robinson 1997], showing improved performance of the cache language model. Our experiments show similar improvements on CLIR.

One potential problem of most proposed co-occurrence models is the use of the approximate word selection algorithm. As described in Section 3.4, each query term translation is actually determined independently. To remedy the problem, Liu et al. [2005] presented a so-called maximum coherent model that is able to estimate translations of multiple query terms simultaneously. In this paper, we remedy the problem simply by combining it with other two translation models. The basic idea is that the translations of a set of query terms need to be joined only when they are really correlated tightly such as query terms within a NP or a dependency. In this sense, our query translation methods are both stochastically and linguistically motivated: stochastically because we use statistics from corpus, linguistically because the structures (NPs and dependencies) we defined are informed by syntactic analysis.

7.2 Remarks on the NP Translation Model

A technique often used to deal with the problem of translation ambiguity is to identify phrases in the query and translate them as a whole using a phrase dictionary. It has been shown that this technique can improve IR performance. Hull and Grefenstette [1996] showed that the performance achieved by manually translating phrases in queries is significantly better than that of a word-by-word translation using a dictionary. Davis and Ogden [1997] showed that by using a phrase dictionary extracted from parallel sentences in French and English, the performance of CLIR is improved. Ballesteros and Croft [1998] performed phrase translation using information on phrase and word usage contained in the Collins machine readable dictionary. They demonstrated that translations of multi-word concepts as phrases are more precise. However, a critical problem remains: if a phrase is not stored in a lexicon, how can one identify it in a query and translate it correctly? It is unrealistic to expect a "complete" phrase dictionary. New phrases are constantly created. Therefore, we will always face the problem of identification and translation of unknown phrases, no matter how complete a phrase dictionary may be. This problem is one of the foci of this paper, as described in Sections 4.1 and 4.2.

Now, let us compare in more detail our NP translation model to the classical methods proposed in Ballesteros and Croft [1997] and Brown et al. [1993]. Ballesteros and Croft used a word-by-word strategy for phrase translation. It is based on two assumptions that are sub-optimal. First, they assume that there is a

one-one mapping between words in English NP and words in Chinese NP. However, in our experiments, we found that only 56% NP translation patterns have such one-one mappings. Second, they assume that the translation words in a phrase will remain in the same order as in the source language phrase. In our experiments, we found that 35% of translation patterns change word order. On the other hand, The IBM statistical models incorporate very little linguistic knowledge [Brown et al. 1993]. It is hard to capture non-local dependencies of the language with "local" models such as n-gram models. So even if the translation model generates the correct set of words, the language model will not assemble them correctly. In our method, we incorporate the language model with translation patterns. While the language model captures the "local" dependency, the translation patterns provide information on global dependency within a phrase. Although the method is not powerful enough for sentence-level translation, it performs well for NP translation.

Recently, phrase-based statistical machine translation [Och and Ney 2004; Chiang 2005] advanced the state-of-the-art. As mentioned earlier, our NP translation model is very similar to the template-based translation model described in Och and Ney [2004]. The use of hierarchical structure in our NP translation templates (e.g., $\mathbf{z}_3$ in Figure 5) can be viewed as a special case of the hierarchical phrase-based model in Chiang [2005]. There are however two major differences between our work and that of Och and Ney [2004] and Chiang [2005]. First, the NPs that we deal with are syntactically well-defined constitutes. Och and Ney [2004] and Chiang [2005] extract phrases from bilingual corpus. These phrases are just a sequence of consecutive words, and could be completely meaningless syntactically. Second, our translation templates use POS tag as word class while in Och and Ney [2004], the templates use word classes that are automatically learnt from bilingual corpus. In a word, our model is more syntactically-motivated, and would potentially more accurate and efficient. Moreover, in our study we view NP translation as a subtask of machine translation. We believe that focusing on such a narrower problem would allows more dedicated modeling. Koehn [2003] presents a pretty comprehensive piece of work along this line. The rich feature set used for NP translation, presented in Koehn [2003], might also improve the accuracy of our method.

7.3 Remarks on the Dependency Translation Model

The dependency translation model aims at incorporating syntax information to resolve translation ambiguities. The same goal has also motivated the research of syntax-based MT, which is closely related to our work.

Similar to our method, Ding and Palmer [2005] also use parsers to identify linguistic structures of both Chinese and English languages. Then, they identify those sub-structures from both languages that can be mapped. The identified mappings form the so-called *transduction grammar*.

Due to the structural difference between source and target language, some people use a parser in one language, and map the extracted linguistic structure to the other language [Yamada and Knight 2001; Quirk et al. 2005], assuming that there exist a large set of word-aligned bilingual sentence pairs.

There are also some methods that can learn a transduction grammar without parsing monolingual sentences. For example, Wu [1997] views translation as a

process of bilingual parsing via a synchronized grammar. Each rule of the grammar is a transduction that generates two output strings simultaneously, one for each language. However, for efficiency, the rules have to be unrealistically simple, such as using a single non-terminal symbol in a rule. This limitation is resolved in Chiang [2005], where the grammar is learned on word-aligned bilingual corpus using a more sophisticated probabilistic model.

All the work mentioned above requires a large amount of word-aligned bilingual corpus, which is not always available. Our model can be learned from unrelated bilingual corpus. This benefit results from the fact that we define dependency translation as a subtask of MT, like the case of NP translation model. We also argue that while most existing methods rely on constituency analysis, we believe that dependency analysis bring semantically related words together, and is more effective for resolving translation ambiguities as we showed in Section 5.5.

8. CONCLUSION

This paper presents three statistical query translation models for dealing with the problem of query translation ambiguity. The models differ in their use of linguistic information. The co-occurrence model does not take into account any linguistic structure explicitly, and simply views a query as a bag of words. The other two models, the NP translation model and the dependency translation model, exploit linguistic dependency constraints between terms in NPs or in higher level dependencies. Our evaluations of translation accuracy show that linguistic structures always lead to more precise translations. Our experiments of CLIR on TREC Chinese collections show that all the three models have a positive impact on query translation, and lead to significant improvements of CLIR performance over the simple dictionary-based translation method. The best results are obtained by combining the three models. This is consistent with the observations on general reasoning: when more information is available and is used in reasoning, we usually obtain better results. The integration of different types of knowledge in query translation is the most apparent in the second and third models, where different information is combined as feature functions. This combination method is very effective and it is also a flexible one for integrating more types of information or knowledge when it is available.

There are many areas for future research. One area is to improve the robustness of the parsers that we used to detect phrases and dependency triples. A large portion of translation errors in our experiments are due to the incorrect detection of those linguistic structures. We notice that we only need to extract some specific grammatical relations of a query for translation, and it is unnecessary to get the full analysis result, such as the full parse tree, of a given sentence. Therefore, a partial parser or a grammatical relation detector may be more suitable. This alternative will be investigated in the future. Since all the three models are statistical in nature, the lack of large amount of proper training data would be a disaster. For example, the lack of large enough aligned bilingual corpus would prevent us from extracting more reliable translation templates. The lack of domain-specific datasets leads to the failure of translating domain-specific terms. Recently, people have tried to automatically collect bilingual corpora from web [Nie et al. 1999; Resnik and Smith 2003; Zhang et al. 2005]. Since the web provides a potentially unlimited data source, it turns out to be a very promising research area.

ACKNOWLEDGMENTS

Thanks to Hongzhao He, Weijun Chen, Jian Zhang and Yi Su for their help with our experiments, Endong Xun and Cong Li for the implementation of the NP identification system, and to Chang-Ning Huang for useful discussions.

REFERENCES

Adriani, M. 2000. Using statistical term similarity for sense disambiguation in cross-language information retrieval. *Information Retrieval*. 2, 69-80.

Ballesteros, L. and Croft, W. B. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. In: *Proceedings of the 20th International Conference on Research and Development in Information Retrieval*. 84-91.

Ballesteros, L. and Croft, W. B. 1998. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st International Conference on Research and Development in Information Retrieval*. Melbourne, Australia.

Bian, G. W. and Chen, H. H. 1998. Integrating query translation and document translation in a cross-language information retrieval system. *Machine Translation and Information Soup*, Lecture Notes in Computer Science, #1529, Spring-Verlag, pp. 250-265.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R.L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2): 263-311.

Chen, A., Jiang, H. and Gey, F. 2000. Combining multiple sources for short query translation in Chinese-English cross-language retrieval. In: *Proceedings of the 5th International Workshop Information Retrieval with Asian Languages*, pp. 17-23.

Chen, S. F., and Goodman, J. 1998. An empirical study of smoothing techniques for language modeling. Technical Report, TR-10-98, Harvard University.

Chiang. D. 2005. A hierarchical phrase-based model for statistical machine translation. In: *ACL 2005*.

Church, K., 1988. A stochastic parts program and noun phrase parser for unrestricted text. In: *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136-143. Association of Computational Linguistics.

Clarkson, P. and Robinson, A. 1997. Language model adaptation using mixtures and an exponentially decaying cache. In: *Proceedings of ICASSP-97*, pp. 799--802.

Davis, M. W., and Ogden, W. C. 1997. Free resources and advanced alignment for cross-language text retrieval. In: *The Sixth Text Retrieval Conference (TREC-6)*. NIST, Gaithersbury, MD.

Ding, Y. and Palmer, M. Machine translation using probabilistic synchronous dependency insertion grammars. In: *ACL 2005*, pp. 541-548.

Duda, Richard O, Hart, Peter E. and Stork, David G. 2001. *Pattern classification*. John Wiley & Sons, Inc.

Fagan, J. 1988. *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods.* Computer Science, Cornell University.

Fox. H. J. 2002. Phrasal cohesion and statistical machine translation. In: *EMNLP 2002*.

Fung, P., Liu, X., and Cheung, C. S. 1999. Mixed language query disambiguation. In *ACL-99*. The 37th Annual Meeting of the Association for Computational Linguistics, College Park, Maryland, USA, pp. 333-340.

Gao, J., Goodman, J. and Miao, J. 2001a. The use of clustering techniques for language modeling – application to Asian languages. *Computational Linguistics and Chinese Language Processing*, Vol. 6, No. 1, pp 27-60.

Gao, J., Goodman, J., Li, M. and Lee, K. F. 2002a. Toward a unified approach to statistical language modeling for Chinese. *ACM Transactions on Asian Language Information Processing,* Vol. 1, No. 1, pp 3-33.

Gao, J., Li, M., Wu, A. and Huang, C. N. 2005a. Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics*, 31 (4).

Gao, J., Nie, J. Y., He, H., Chen, W. and Zhou, M. 2002b. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In: *SIGIR 2002*, pp. 183-190.

Gao, J., Nie, J. Y., Zhang, J., Xun, E., Su, Y., Zhou, M., and Huang, C. 2000. TREC-9 CLIR experiments at MSRCN. In: *TREC-9*, pp. 343-353.

Gao, J., Nie, J. Y., Zhang, J., Xun, E., Zhou, M., and Huang, C. 2001b. Improving query translation for CLIR using statistical Models. In: *ACM SIGIR'01,* New Orleans, Louisiana, pp. 96-104.

Gao, J., Yu, H., Yuan, W. and Xu, P. 2005b. Minimum sample risk methods for Language modeling. In: *HLT/EMNLP 2005*.

Heidorn, G. 2000. Intelligent writing assistance. In Dale et al. editor *Handbook of Natural Language Processing*, Marcel Dekker.

Hiemstra, D., and Jong, F. 1999. Disambiguation strategies for cross-language information retrieval. In: *Proceedings of the third European Conference on Research and Advanced Technology for Digital Libraries*. 274-293.

Huang, X., Acero, A., and Hon, H-W. 2001 *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice Hall PTR.

Hull, D. A. 1997. Using structured queries for disambiguation in cross-language information retrieval. In: *AAAI Symposium on Cross-Language Text and Speech Retrieval*.

Hull, D. A. and Grefenstette, G. 1996. Querying across languages: a dictionary-based approach to multilingual information retrieval. In: *Research and Development in Information Retrieval, ACM-SIGIR,* pp46-57.

Jang, M.G., Myaeng, S. H., and Park S. Y. 1999. Using mutual information to resolve query translation ambiguities and query term weighting. In: *ACL-99*. College Park, Maryland, pp. 223-229.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. 1999. An introduction to variational methods for graphical models. In Jordan, M. I. editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

Koehn, P. 2003. *Noun Phrase Translation*. Ph.D. thesis, University of Southern California.

Kowk, K. L. 2000. Exploiting a Chinese-English bilingual wordlist for English-Chinese cross language information retrieval. In: *Fifth International Workshop on Information Retrieval with Asian Languages, IRAL-2000.* Hong Kong, September 30 to October 1, 2000.

Lin, D. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In: *Proceedings of ACL/EACL-97*, Madrid, pp. 64-71.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In: *COLING-ACL98,* Montreal, Canada, August, pp. 768-774.

Liu, Y., Jin, R. and Chai, J. Y. 2005. A maximum coherence model for dictionary-based cross-language information retrieval. In: *SIGIR 2005*, pp. 536-543.

Mandala, R., Tokunaga, T., and Tanaka, H. 1999. Combining multiple evidence from different types of thesaurus for query expansion. In: *ACM SIGIR'99*. pp 191-197.

Marcus, M., Marcinkiewicx, M., and Santorini, B. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2): 313-330.

Nie, J. Y., Simard, M., Isabelle, P., and Durand, R. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In: *Conference on Research and Development in Information Retrieval, ACM-SIGIR*, pp. 74-81

Nie, J.Y. 2003. Query expansion and query translation as logical inference. *Journal of the American Society for Information Science and Technology*, 54(4): 335-346.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In: *ACL 2003*.

Och, F. J. and Ney, H. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30, pp. 417-449.

Peters, C. and Picchi, E. 1996. Cross language information retrieval: A system for comparable corpus querying. In: *SIGIR'96 Workshop on Cross-linguistic Information Retrieval*, pp. 24--33.

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. 1992. *Numerical Recipes In C: The Art of Scientific Computing*. New York: Cambridge Univ. Press.

Quirk, C. Menezes, A., and Cherry, C. 2005. Dependency treelet translation: syntactically informed phrasal SMT. In: *ACL 2005*, pp. 271-279.

Ramshaw, L. A. and Marcus, M. 1998. *Text chunking using transformation-based learning*. In *Natural Language Processing Using Very large Corpora*. Kluwer. Originally appeared in The second workshop on very large corpora WVLC'95, pp.82-94.

Resnik, P. and Smith, N. A. 2003. The web as a parallel corpus. *Computational Linguistics*. 29 (2003) pp.349-380

Robertson, S. E., and Walker, S. 2000. Microsoft Cambridge at TREC-9: Filtering track. In: *TREC-9*, pp. 361-368.

Tjong Kim Sang E. and Buchholz, B. S. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, Lisbon, Portugal.

van Rijsbergen, C. J. 1979. *Information retrieval*, 2nd ed. Butterworths, London.

van Rijsbergen, C. J. 1986. A Non-Classical Logic for Information Retrieval. *The Computer Journal*, 29(6): 481-485.

Voorhees, E., Harman, D. 2001. Overview of the ninth text retrieval conference (TREC-9). In: *TREC-9* pp. 1-14.

Wu, D. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23 (3), pp. 377-403.

Xu, J., and Weischedel, R. 2000a. Cross-lingual information retrieval using Hidden Markov models. In: *Proceeding of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong, October 7-8, 2000.

Xu, J., and Weischedel, R. 2000b. TREC-9 cross-language retrieval at BBN. In: *The Ninth Text Retrieval Conference (TREC-9)*. NIST, Gaithersbury, MD.

Xun, E., Zhou, M., and Huang, C. 2000. A unified statistical model for the identification of English base NP. In: *ACL 2000, The 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong 3-6 October.

Yamada, K. and Knight, K. 2001. A syntax based statistical translation model. In: *ACL 2001*.

Zhang, Y. and Vines, P. 2004. Using the web for automated translation extraction in cross-language information retrieval. In: *SIGIR2004*, pp. 162-169.

Zhang, Y., Wu, K., Gao, J. and Vines, P. 2006. Automatic acquisition of Chinese-English parallel corpus from the web. In: *ECIR2006*.

Zhou, M., Ding, Y., and Huang, C. 2001. 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.

Authors' addresses: Jianfeng Gao, Microsoft Research, One Microsoft Way, Redmond. WA. 98052, U.S.A. Email: jfgao@microsoft.com. Jian-Yun Nie, University de Montreal, Email: nie@iro.umontreal.ca. Ming Zhou, Microsoft Research Asia, 5F Sigma Center, Beijing, China, 100080.