A Study of Statistical Models for Query Translation: Finding a Good Unit of Translation

Jianfeng Gao
Microsoft Research
jfgao@microsoft.com

Jian-Yun Nie
Université de Montréal
nie@iro.umontreal.ca

ABSTRACT

This paper presents a study of three statistical query translation models that use different units of translation. We begin with a review of a word-based translation model that uses co-occurrence statistics for resolving translation ambiguities. The translation selection problem is then formulated under the framework of graphic model resorting to which the modeling assumptions and limitations of the co-occurrence model are discussed, and the research of finding better translation units is motivated. Then, two other models that use larger, linguistically motivated translation units (i.e., noun phrase and dependency triple) are presented. For each model, the modeling and training methods are described in detail. All query translation models are evaluated using TREC collections. Results show that larger translation units lead to more specific models that usually achieve better translation and cross-language information retrieval results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: *Retrieval models*

General Terms

Design, Algorithms, Theory, Experimentation

Keywords

Query Translation, Cross-Language Information Retrieval, Statistical Models, Linguistic Structures

1. Introduction

Query translation is a long standing research topic in the community of cross-language information retrieval (CLIR). Assume that a query is translated using a bilingual dictionary, there are two fundamental research tasks: (1) how to improve the coverage of the bilingual dictionary; and (2) how to select the correct translation of the query among all the translations provided by the dictionary. The second task is also called the problem of *translation selection*, and is the focus of this paper.

We limit our discussion in this paper to statistic models for translation selection. To make a statistical model trainable, we always decompose the translation of a query into a sequence of smaller *translation units*. Given that we will use statistical models to model translation, one important question is: *what unit of translation should a statistical model represent*?

SIGIR'06, August 6-10, 2006, Seattle, Washington, USA.

A natural unit is words since they are the smallest meaningful linguistic unit. However, a word-based model, though simple to train, is always over-general and may lead to too many translation ambiguities for resolving. In theory, larger units such as phrases are more specific and lead to less translation ambiguities. However they pose bigger challenges in model structuring and training.

This paper studies three statistical query translation models that use different translation units. We will begin with a review of a classical word-based translation model that uses co-occurrence statistics to resolve translation ambiguities. Then, we will formulate the translation selection problem under the framework of graphic model (GM). We will discuss the modeling assumptions and limitations of the co-occurrence model, and motive our research of finding better translation units.

We will then present two other models that use larger, linguistic-motivated translations units. They are noun phrases (NPs) and dependencies. A dependency, represented as a triple, is a pair of words that have a syntactic dependency relation, such as verb-objective. In both models, we assume that the selection of a translation only depends upon other selected translations in the same unit. While NPs capture dependence of adjacent words in a query, dependency triples can capture syntactic dependences between non-adjacent words. Though similar models have been proposed earlier, we will refine those using recent advances in the research community of statistical machine translation (SMT). We will show that (1) NP and dependency translation can be performed using a reranking approach based on a linear model; (2) the linear model provides a flexible statistical framework to incorporate various kinds of information, defined as feature functions, for resolving translation ambiguities; (3) the parameters of the linear model can be learned discriminatively so as to optimize the translation quality directly, and (4) most effective feature functions used in the linear model can be derived from generative models that are traditionally used in SMT, thus the ranking approach provides an appropriate framework to combine the strengths of both generative models and discriminative training methods.

We evaluate our query translation models using TREC collections. To our knowledge, this is the first systematic comparison of those models on the task of English to Chinese CLIR on gold test sets. We shall demonstrate that linguistic units such as NP and dependency triples are beneficial to query translation if they can be detected and used properly.

2. Co-occurrence Model

A co-occurrence model uses words as the unit of translation. The basic principle of the model is that correct translations of query words tend to co-occur in the target language and incorrect translations do not. Therefore, for a given query word

in source language (i.e., English in this study), the likelihood of its translation (i.e., Chinese) is measured via the similarity between a translation candidate (e.g., provided by a bilingual dictionary) and the other selected translations in the query. The definition of similarity between words can take different forms of co-occurrence statistics. Mutual information is among the most commonly used ones [22].

The advantage of the co-occurrence model is that it is easy to train. There is no need to measure cross-language word similarities (e.g., translation probabilities). Only relationships between words of the same language are used. They can be obtained through co-occurrence statistics in a monolingual text corpus. The disadvantage of the model is that it is difficult to find an efficient algorithm that optimizes exactly the translation of a whole query according to the model. We now describe it in detail.

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 translation is computationally very expensive, as will be described below. Therefore, one has to use an approximate greedy algorithm as follows [1, 10, 11]:

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

(2) For each set $\mathbf{D}(e_i)$

(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 Eq. (1)

$$sim(c_{i,j}, \mathbf{D}(e_k)) = \sum_{c_{k,l} \in \mathbf{D}(e_k)} sim(c_{i,j}, c_{k,l}) \quad (1)$$

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

$$cohesion(c_{i,j} \mid \mathbf{e}, \mathbf{D}) = \log[\sum_{\mathbf{D}(e_k)} sim(c_{i,j}, \mathbf{D}(e_k))] \quad (2)$$

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

$$c = \arg\max_{c_{i,j} \in \mathbf{D}(e_i)} cohesion(c_{i,j} \mid \mathbf{e}, \mathbf{D}) \quad (3)$$

Apparently, the above algorithm is sub-optimal. As pointed out in [15], the cohesion score for a translation as in Eq. (2) 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 section, we will formulate the translation selection problem under the framework of GM [e.g., 13], and discuss the underlying assumptions of the greedy algorithm.

3. GM View

A query translation model can be viewed as an undirected GM. For example, Figure 1 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

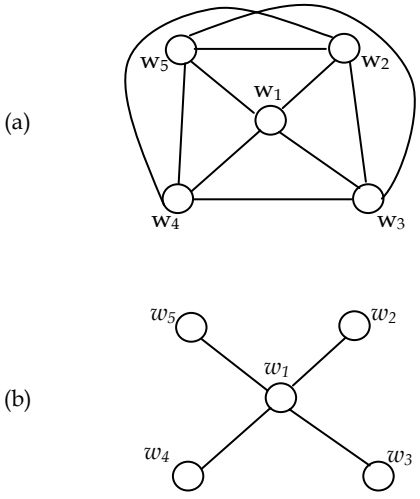

Figure 1. GMs the co-occurrence query translation model (a) and its approximation (b).

translations that maximize the joint probability $P(w_1,w_2,w_3,w_4,w_5)$.

The GM view illustrates 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 1 (a). Suppose that we wish to compute the marginal probability $P(w_1)$. We obtain this marginal by summing over the other variables as:

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

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 2 by introducing a translation independence assumption. The corresponding GM is shown in Figure 1 (b). Now, $P(w_1)$ can then be factored as:

$$P(w_1) = \frac{1}{Z} \sum_{w_2} h(w_1, w_2) \sum_{w_3} h(w_1, w_3) \sum_{w_4} h(w_1, w_4) \sum_{w_5} h(w_1, w_5) \quad (4)$$

Notice that if we define $h(.)$ in Equation (4) as the similarity between two words, the idea behind Equation (4) is similar to that of Equation (2) (Z can be removed when the probability

is used to rank translation candidates), where no more than two variables appear together in any summand, and thus the computational complexity is reduced to d^2. However, as discussed earlier, the reduction of complexity may come with the sacrifice of accuracy due to the independence assumption used.

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 5 and 6 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 unit 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 5.
- Translate words in dependencies using the dependency translation model described in Section 6.
- Translate remaining words using the co-occurrence model.

4. Reranking Approach

This section describes the reranking approach which is the fundamental modeling framework for both NP and dependency translation models.

Given an n-term English query $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$, we assume some way of detecting linguistic structures $\mathbf{s}$ of $\mathbf{e}$. We also assume some way of generating a set of candidate Chinese translations $\mathbf{c}$, denoted by $\mathbf{GEN}(\mathbf{e})$. The task of a query translation model is to assign a score for each of the translation candidates in $\mathbf{GEN}(\mathbf{e})$ and select the one with the highest score:

$$\mathbf{c}^* = \arg\max_{\mathbf{c} \in \mathbf{GEN}(\mathbf{e})} Score(\mathbf{c}, \mathbf{e}, \mathbf{s}) \quad (5)$$

In this study, we assume that the score is assigned via a linear model, which consists of (1) a set of D feature functions that map ($\mathbf{c}$, $\mathbf{e}$, $\mathbf{s}$) to a real value, i.e., $f_d(\mathbf{c}, \mathbf{e}, \mathbf{s})$, 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 (5) can be rewritten as

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} \sum_{d=1}^{D} \lambda_d f_d(\mathbf{c}, \mathbf{e}, \mathbf{s}) \quad (6)$$

Notice that the linear model of Equation (6) is a very general framework [6]. For example, the source-channel models for SMT [16] can be viewed as a special case of the linear model if we define both source model and channel model as feature functions. We shall show that most feature functions can be derived from generative models which are traditionally used in the framework of source-channel models for SMT. The values of those feature functions are (log) probabilities that are learned from large monolingual or bilingual corpora via MLE. Therefore, those features are more informative than binary features that are traditionally used in linear models for classification problems [6].

The model weights λ, as shown in Equation (6), are estimated using an iterative procedure that is used for multi-dimensional function optimization [18]. Assume that we can minimize query 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 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 [9, 17].

In the next two sections we will describe in turn the NP translation model and the dependency translation model. Both models are of the form of linear models in Equation (6). For each model, we will first describe a generative translation model (consisting of a series of component models) under the framework of source-channel models. Then, we derive feature functions (e.g., from the component models) used in the linear models.

5. NP Translation Model

The use of NP as a unit of translation is motivated by two observations. First, most English NPs are translated to Chinese as NPs. For example, on a 60K-sentence-pair word-aligned English-Chinese bilingual corpus, we found more than 80% of English NPs being aligned to their translated Chinese NPs. Second, as pointed out in [14], word selection can almost always be resolved depending solely upon the internal context of the NP.

The use of *translation template* between English and Chinese NP patterns is the fundamental to our NP translation model. 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. The concept of translation templates is very similar to that of alignment templates in [16]. Formally, a NP translation template, denoted by $\mathbf{z}$, 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 an in-house parser 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 2, where (a) is an English sentence with each word marked by its POS tag and

(a)	[[the/DT/1 sales NNP/2] of/IN/3 [Chinese/NNP/4 ships/NNP/5]] …
(b)	中国/1 船舶/2 销售/3
(c)	(1, ε) (2, 3) (3, ε) (4, 1) (5, 2)
(d)	$\mathbf{z}_1$ = (*E* = [*DT NNP*], *C* = [*NNP*], *A* = {(1, ε), (2,1)}) $\mathbf{z}_2$ = (*E* = [*NNP-1 NNP-2*], *C* = [*NNP-1 NNP-2*], *A* = {(1, 1), (2, 2)}) $\mathbf{z}_3$ = (*E* = [*Base-NP-1 of Base-NP-2*], *C* = [*Base-NP-2 Base-NP-1*], *A* = {(1, 2), (2, ε), (3, 1)})

Figure 2. NP translation templates patterns

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.

5.1 Generative Model

Given an English NP **e**, we search among all possible translations the most probable Chinese NP **c*** as

$$\mathbf{c}^* = \arg\max_{\mathbf{c}} P(\mathbf{c}\,|\,\mathbf{e}) = \arg\max_{\mathbf{c}} P(\mathbf{c})P(\mathbf{e}\,|\,\mathbf{c}) \tag{7}$$

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

$$P(\mathbf{c}) = P(c_1)P(c_2\,|\,c_1)\prod_{j=3\ldots J} P(c_j\,|\,c_{j-2}c_{j-1}) \tag{8}$$

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

$$P(\mathbf{e}\,|\,\mathbf{c}) = \sum_{\mathbf{z}} P(\mathbf{z}\,|\,\mathbf{c})P(\mathbf{e}\,|\,\mathbf{z},\mathbf{c}) \tag{9}$$

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 **z** = (*E*, *C*, *A*), we call **z** *applicable* to **c** if **c** matches the NP pattern *C*. Let $C(\mathbf{c}, \mathbf{z})$ be the number of occurrences of **c** to which **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

$$P(\mathbf{z}\,|\,\mathbf{c}) = \frac{C(\mathbf{c},\mathbf{z})}{C(\mathbf{c})} \tag{10}$$

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

$$P(\mathbf{e}\,|\,\mathbf{z},\mathbf{c}) = P(\mathbf{e}\,|\,(E,C,A),\mathbf{c}) = \prod_{(i,j)\in A} P(e_i\,|\,c_j) \tag{11}$$

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

$$P(e\,|\,c) = \frac{C(c,e)}{C(c)} \tag{12}$$

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 (11) is a deficient model since the constraint $\Sigma_{\mathbf{e}} P(\mathbf{e}\,|\,\mathbf{z}, \mathbf{c}) = 1$ does not hold, as discussed in [16]. However it is not necessary to normalize it since we only use the model as a feature function for ranking translation candidates. 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 2 (d). As shown in Figure 2 (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 2 (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 (9) into Equation (7), we finally get

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

Notice that different component models in Equation (13) are trained on different corpora of different sizes. The dynamic value ranges of different component model probabilities can be so different (e.g., $P(\mathbf{e}\,|\,\mathbf{z}, \mathbf{c})$ of Equation (11) is not a *probability* but a *score*) that it is inappropriate to combine all these models through simple multiplication as in Equation (13). Moreover, models that are poorly trained (e.g., due to lack of training data) should be less weighted than well-trained ones. One way to balance the impact of these models 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. It is thus 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.

5.2 Feature Functions

We used three feature functions. They are derived from the above three component models in Equation (13), respectively.

1. **Chinese language model feature.** It is defined as the logarithm of the Chinese trigram model of Equation (8), i.e., $h_{LM}(\mathbf{c}) = \log P(\mathbf{c}) = \log P(c_1)P(c_2)\Pi_{i=3\ldots J}P(c_j\,|\,c_{j-2}\,c_{j-1})$.
2. **Translation template selection model feature.** It is defined as the logarithm of $P(\mathbf{z}\,|\,\mathbf{c})$, i.e., $h_{TS}(\mathbf{z}, \mathbf{c}) = \log P(\mathbf{z}\,|\,\mathbf{c})$.
3. **Word selection model feature.** It is defined as the logarithm of $P(\mathrm{e}\,|\,\mathrm{z}, \mathrm{c})$ of Equation (11), i.e., $h_{WS}(\mathbf{e}, \mathbf{z}, \mathbf{c}) = \log P(\mathbf{e}\,|\,(E, C, A), \mathbf{c}) = \log\Pi_{(i,j)\in A}P(e_i\,|\,c_j)$.

Notice that the linear model of Equation (6) does not take into account the sum on **z** in Equation (13), because considering the sum in decoding directly is computationally expensive. Therefore, we approximate the sum during 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 translation template, 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 a best-first decoder to find top n translation candidates according to Equation (6) where only two features, h_{LM} and h_{WS}, are used.
4. **Fusion and reranking.** We fusion all retained translation candidates, and rerank them according to Equation (6), where all features are applied.

We can see that the sum on **z** is approximated in two steps: First, for each **z**, we find the best translation. Second, we select the translation among all retained best translations according to the linear model.

6. Dependency Translation Model

A dependency is denoted by a triple (w_1, r, w_2), representing a syntactic dependency relation r between two words w_1 and w_2. Among all the dependency relations, we only consider the four types that can be detected precisely using our parser and cannot be handled by the NP translation model: (1) subject-verb, (2) verb-object, (3) adjective-noun, and (4) adverb-verb.

Similar to that of the NP translation model, the dependency translation model is also developed based on two hypotheses. First, dependencies have the best cohesion properties across languages [7]. 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.

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. 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). [10, 27] also showed that more than 80% of dependency relations of the above four types have one-to-one mappings between English and Chinese.

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.

6.1 Generative 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

$$\mathbf{c}_t^* = \arg\max_{\mathbf{c}} P(\mathbf{c}_t \mid \mathbf{e}_t) = \arg\max_{\mathbf{c}} P(\mathbf{c}_t) P(\mathbf{e}_t \mid \mathbf{c}_t) \quad (14)$$

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

$$P(\mathbf{c}_t) = \frac{C(\mathbf{c}_t)}{N} \quad (15)$$

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

$$P(\mathbf{e}_t \mid \mathbf{c}_t) = P(e_1 \mid c_1) P(e_2 \mid c_2) \delta(r_e, r_c) \quad (16)$$

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 (12). 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 (16) 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 $sim(e, c)$. Notice that e and c are not necessary to be a translation pair but just a pair of *cross-lingual synonyms*, 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.

In this study, we follow the method of [10, 27] to estimate the value of $sim(e, c)$. The advantage of the method is that $sim(e, c)$ can be learned on unrelated English and Chinese dependency triple corpora.

6.2 Feature Functions

We see from Equations (14) and (16) 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 (15), and (2) $sim(e, c)$. Similar to the NP translation model, we define a feature function for each type of factors, and combine them under the framework of linear models as shown in Equation (6). The two types of features are defined as follows.

1. **Chinese language model feature.** It is defined as the logarithm of the model of Eq. (14), i.e., $h_{LM}(\mathbf{c}_t) = \log P(\mathbf{c}_t)$.
2. **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.

7. Experiments

7.1 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 [8]. 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 dictionary contains 401,477 English entries, including 109,841 words, and 291,636 phrases.

The Okapi system with BM2500 weighting [21] is used as the basic retrieval system. The main evaluation metric is interpolated 11-point average precision. Statistical significance test (i.e., t-test) is also employed.

7.2 Results

The main results are shown in Tables 1 to 3 (i.e., average precisions) and Figures 2 and 3 (i.e., precision-recall curves). To investigate the effectiveness of our models for query translation, three baseline methods are compared, denoted by ML, ST and BST, respectively.

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 [24]. This 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 is the co-occurrence translation model described in Section 2. We implemented a variant, called decaying co-occurrence model [10]. The word similarity is defined as

Table 1: 11-point average precision (AP) for short queries on TREC-9 dataset (* indicates that the improvement is statistically significant.)

	Translation Model	AP	% 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 2: 11-point average precision (AP) for long queries on TREC-9 dataset (* indicates that the improvement is statistically significant.)

	Translation Model	AP	% 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%*

Table 3: 11-point average precision (AP) for long queries on TREC5&6 dataset (* indicates that the improvement is statistically significant.)

	Translation Model	AP	% 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%*

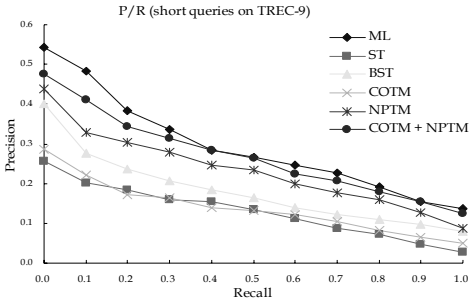

Figure 3: Precision-Recall curves for short queries on TREC-9 dataset.

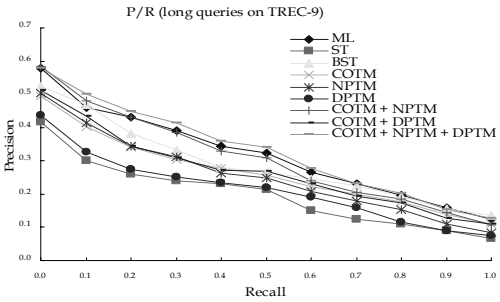

Figure 4: Precision-Recall curves for long queries on TREC-9 dataset.

$$sim(w_i, w_j) = MI(w_i, w_j) \times D(w_i, w_j) \quad (17)$$

where $MI(.)$ is the mutual information between two words, and is estimated on a Chinese newspaper corpus. $D(.)$ is a penalty function, indicating that the mutual information between words decreases exponentially with the increase of the distance between them. It is defined as

$$D(w_i, w_j) = \exp(-a * (Dis(w_i, w_j) - 1)) \quad (18)$$

where α is the decaying rate ($\alpha = 0.8$ in our experiments), and $Dis(w_i,w_j)$ is the average intra-sentence distance between w_i and w_j in the Chinese newspaper corpus.

NPTM is the NP translation model described in Section 5. The translation template selection model (i.e., value of $h_{TS}(\mathbf{z},\mathbf{c})$) is trained on a word-aligned bilingual corpus containing approximately 60K English-Chinese sentence pairs. Translation templates are first extracted automatically from the corpus using an in-house chunking parser, and then filtered by a linguist. The probability $P(\mathbf{z} \mid \mathbf{c})$ is then estimated according to Equation (10). For each Chinese NP pattern, there are 4.21 translation templates on average. The word selection model (i.e., $h_{WS}(\mathbf{e}, \mathbf{z}, \mathbf{c})$) are computed according to Equation (12) using the same word-aligned bilingual corpus. The Chinese trigram model (i.e., $h_{LM}(\mathbf{c})$) is trained on a word-segmented Chinese corpus consisting of about 1 billion words.

DPTM is the dependency translation model described in Section 6. $sim(e, c)$ is estimated using two unrelated English and Chinese corpora (i.e., 87-97 WSJ newswires for English and 80-98 People's Daily articles for Chinese). An English and Chinese parser NLPWIN [12] is used to extract dependency triples in both corpora. Notice that NLPWIN is a rule-based parser and performs well only when the input is a grammatical sentence, so we only tested DPTM on long queries (i.e., to parse the descriptions and narratives).

The experimental results in Tables 1 to 3 and Figures 2 and 3 give rise to the following observations.

First of all, we see that that COTM brings statistically significant improvements over ST for long queries but its improvement over ST for short queries is marginal. This is expected because COTM resolves translation ambiguities with resort to context terms. Long queries contain much richer contextual information than short queries.

Secondly, we see that NPTM achieves substantial improvements over ST for both long and short queries, and even outperforms BST for short queries, as shown in Rows 3 and 5 in Table 1. 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 1). 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.

Thirdly, we find that the use of DPTM leads to an effectiveness well below that with COTM and NPTM. For example, as shown in Table 2 (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. So this "counter-performance" is not surprising. A further analysis shows that from the 11 queries, NLPWIN extracted 52 dependency triples which appear at least 5 times in the corpus. The 52 triples include 12 verb-object dependency triples, 8 sub-verb triples, 32 adjective-noun triples and no adv-verb triple. For these queries, the dependency triple translation has positive impact on the methods of ST and COTM for 10 out of the 11 queries, which leads to a statistically significant improvement of 58% over ST, and 11% over COTM for the 11 queries.

Finally, we see that as expected, the combined models (using the sequential combining approach described in Section 3) 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.

8. Related Work and Discussion

Co-occurrence information has been utilized by several recent studies [2, 3, 10, 11, 15] to deal with the translation selection problem for CLIR. One potential problem of most proposed co-occurrence model is the use of the approximate word selection algorithm. As described in Section 2, each query term translation is actually determined independently. To remedy the problem, Liu et al. [15] 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 using larger, linguistic-motivated units of translation. The basic idea is that the translations of a set of query terms that need to be jointed optimized only when they are really correlated tightly such as query words 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 translation units (NPs and dependencies) we defined are informed by syntactic analysis.

The NP translation model is inspired by recent work on phrase-based SMT. Our NP translation template is very similar to the template-based translation model described in [16]. The use of hierarchical structure in our NP translation templates can be viewed as a special case of the hierarchical phrase-based model in [4]. There are however two major differences between our work and that of [4, 16]. First, the NPs that we deal with are syntactically well-defined constitutes. [4, 16] 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 [16], 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 allow more dedicated modeling. Koehn [14] presents a pretty comprehensive piece of work along this line. The rich feature set used for NP translation, presented in [14], might also improve the accuracy of our method.

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, [5] 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, people also use a parser in one language, and map the extracted linguistic structure to the other language [19, 25], 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 [4, 23]. While most previous work 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.

9. Conclusions

This paper presents three statistical query translation models for dealing with the problem of query translation ambiguity. The models differ in the use of translation unit and the use of linguistic information. The co-occurrence model is based on word translation. It 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, use larger, linguistically motivated translation units, and can exploit linguistic dependency constraints between words in NPs or in higher level dependencies. Our experiments of CLIR on TREC Chinese collections show that models using larger and more specific unit of translation are always better, if the models can be well trained, because more specific models could model more information. 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 NP and dependency models. Both are constructed under the framework of linear models, where different information is combined as feature functions. This combination method is very effective flexible to incorporate more types of information or knowledge when it is available.

It is well known that statistical translation models will perform better with larger translation units. It is also well-known that models using larger translation units require more training data. Thus, our work can be viewed as finding a tradeoff between specificity and trainability. Given a limited amount of training data, we always try to make the model as specific as possible. Recently, people have tried to automatically collect bilingual corpora from web [20, 26]. Since the web provides a potentially unlimited data source, it turns out to be a very promising research area.

References

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

[2] Ballesteros, L. and Croft, W. B. 1998. Resolving ambiguity for cross-language retrieval. In *SIGIR*.

[3] 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.

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

[5] Ding, Y. and Palmer, M. Machine translation using probabilistic synchronous dependency insertion grammars. In: *ACL 2005*.

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

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

[8] 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).

[9] Gao, J., Qi, H., Xia, X., and Nie, J-Y. 2005b. Linear discriminant model for information retrieval. In: *SIGIR 2005*.

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

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

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

[13] 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.

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

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

[16] Och, F. J. 2002. *Statistical machine translation: from single-word models to alignment templates*. Ph. D. thesis. RWTH Aachen.

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

[18] 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.

[19] Quirk, C. Menezes, A., and Cherry, C. 2005. Dependency treelet translation: syntactically informed phrasal SMT. In: *ACL 2005*.

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

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

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

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

[24] Xu, J., and Weischedel, R. 2000b. TREC-9 cross-language retrieval at BBN. In: *TREC-9,* NIST, Gaithersbury, MD.

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

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

[27] 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.