Improving Named Entity Translation by Exploiting Comparable and Parallel Corpora

Ahmed Hassan* Haytham Fahmy Hany Hassan
IBM Cairo Technology Development Center
Giza, Egypt
P.O. Box 166 Al-Ahram
hasanah,haythamf,hanyh@eg.ibm.com

Abstract

Translation of named entities (NEs), such as person, organization, country, and location names is very important for several natural language processing applications. It plays a vital role in applications like cross lingual information retrieval, and machine translation. Web and news documents introduce new named entities on regular basis. Those new names cannot be captured by ordinary machine translation systems. In this paper, we introduce a framework for extracting named entity translation pairs. The framework contains methods for exploiting both comparable and parallel corpora to generate a regularly updated list of named entity translation pairs. We evaluate the quality of the extracted translation pairs by showing that it improves the performance of a named entity translation system. We report results on the ACE 2007 Entity Translation (ET) pilot evaluation development set.

Keywords

Named Entity Translation, Machine Translation, Multilingual NLP

1 Introduction

The problem of named entity translation is receiving huge attention recently. Named entity translation is a very important component in several natural language processing applications. Applications like cross lingual information retrieval, and machine translation benefit the most of named entity translation.

Regularly updated documents such as news articles and web pages usually contains a large number of names. Those names are much more varied than common words, and changing continuously. As a matter of fact, new names are introduced in the news on regular basis. This makes it hard to construct a named entities dictionaries or translation pairs lists. This is very problematic for the task of named entity translation because translation systems suffer from the lack of enough training data. Hence, machine translation systems usually fail to capture those new names.

One way to go over this problem is to exploit the much more available comparable corpora. Comparable corpora are data sets written in different languages, and are not translations of each others (i.e. not parallel). However comparable corpora are somewhat related and convey the same or overlapping information content. In other words, they are texts that discuss similar subjects, and carry similar information content, yet are written in different languages. The most obvious examples of comparable corpora are the multilingual news feeds produced by news sources such as BBC, CNN, Xinhua, Agence France Press, ...etc.

Comparable corpora are widely available on the web and are regularly updated with new named entities. They are also available for several language pairs. They usually contain documents discussing the same problem or commenting on the same event. such documents usually use similar sets of named entities. However, extracting named entity translation pairs from comparable corpora is hard. Even though they contain texts carrying the same information content, those texts are not aligned and exhibits great differences.

Another resource that is available for some language pairs is parallel corpora. Aligned parallel corpora are multilingual corpora that have been specially formatted for side-by-side comparison. Parallel corpora that are aligned on sentence level can also be used for extracting named entity translation pairs.

In this paper, we introduce a general framework for extracting named entity translation pairs. The framework contains methods for extracting named entity translation pairs from both comparable and parallel corpora. The extracted pairs helped improve the performance of a named entity translation system.

The paper is organized as follows: in section 2 we discuss related work. A method for extracting named entity translation pairs from comparable corpora is presented in section 3. In section 4, we present another method for extracting named entity translation pairs from parallel corpora. Section 5 discusses experimental results while the conclusion is presented in section 6.

2 Related Work

[5] proposed an approach to extract NE trans-lingual equivalences based on the minimization of a linearly combined multi-feature cost. [9] proposed using statistical phrase translation models to find NE translations. [8] proposed an iterative algorithm that exploit the observation that that NEs have similar time distributions across comparable corpora, and that they are

*Now with the University of Michigan Ann Arbor (hassanam@umich.edu).

often transliterated. This approach was used in [10] to discover NEs, but in a single language, English, across two news sources. [6] proposed a method for mining key phrases translations from mixed-language web pages. [16] makes use of the observation that when English terms occur in Chinese web pages, and especially when they occur within brackets, they are very likely to be translations of an immediately preceding Chinese term. [15] extends this approach by using cross-lingual query expansion to find translations of out of vocabulary terms.

3 Extracting named entity translation pairs from comparable corpora

In this section, we introduce our approach for extracting translation pairs form comparable corpora . The proposed approach is composed of two main steps, the first step aligns bi-lingual documents, from the comparable corpus, based on their semantic content. The second step extracts two catalogs, lists, of named entities from each pair of aligned documents and then align the named entities to extract the translation pairs. A detailed description of the approach is presented in the following subsections. In this work, we deployed the approach on Arabic - English comparable corpus to extract named entities translation pairs, however it is worth mentioning that the approach is language independent and could be used with any languages pairs.

3.1 Aligning Documents from Comparable Corpora

Comparable corpora usually contain documents discussing the same events or carrying the same information content. However those documents are not aligned to one another. We introduce a new method for aligning multilingual documents based on the analysis of their semantic content. The method consists of two main phases. In the first phase, a set of candidate matching English documents are generated for each foreign, here Arabic, document. In the second phase, the best matching document among the set of candidates is identified. The two phases are discussed in details in the following subsections.

3.1.1 Generating candidate similar documents

The Arabic documents are translated into English, this translation is not assumed to be accurate neither perfect. It is used only for extracting English keywords corresponding to the Arabic document. Those keywords are used to construct a query to search for candidate similar English documents. English words are indexed by Lemurs Indri information retrieval engine [11]. The query words are stemmed using Porter stemmer [13] to remove the commoner morphological endings from English words. Function words, words that have little semantic meaning and mainly serve to express grammatical relationships, are also removed before performing the query. The best n documents returned by the query are used as the candidate similar documents.

Algorithm 1 Selecting the best matching document

Input: Two document D_1 and D_2
Output: similarity score
Algorithm:

1. Extract a bag of words B_1 from D_1
2. Extract a bag of words B_2 from D_2
3. Let $B = B_1 \cup B_2$
4. Remove duplicate words from B
5. For each word pair $w_1 \& w_2 \in B$:
   Let $sim_1 2 = WNet\_Sim(w_1, w_2)$
6. For each $w_i \in B$:
   Add a node with label w_i
7. For each word pair $w_1 \& w_2 \in B$:
   If$(sim_1 2 < T_1)$ add edge between w_1, and w_2
8. Run MCL clustering on G
9. For each word cluster c_i and each word $w_i \in c_i$
   (a) If $(w_i \in B_1)$
      i. Let $n = freq(w_i, B_1)$
      ii. add n instances with label d_1 to c_i
   (b) If $(w_i \in B_2)$
      i. Let $n = freq(w_i, B_2)$
      ii. add n instances with label d_2 to c_i
   (c) Remove w_i from c_i
10. Let $entropy = 0$;
11. For each cluster c_i:
    Let $entropy = entropy + entropy(c_i)$
12. $Docs\_Sim = entropy$

3.1.2 Selecting the best matching document

To select the best matching document, we exploit the fact that documents conveying the same information content tend to use similar words or similar classes of words. Hence, we can judge two documents as being similar, by analyzing the classes of words they use. We represent each document with a bag of words, which is used to measure the similarity between the two documents in the same language, The bag of words for each document is obtained by stemming all words in the two documents then removing all non contetn words,

D1		D2	
Word	Freq	Word	Freq
w1	2	w1	3
w3	3	w2	2
w4	3	w5	2
w6	3	w7	3
w8	3	w8	2
w10	2	w9	2
w11	3	w12	2
w14	2	w13	3

Fig. 1: *Example of the bag-of-words representation*

an example illustrating two bag of words of two documents is shown in figure 1. The two bag of words $B1$, and $B2$ are then merged into a unique list of words. The unique list of words will be first divided into several word clusters. Theses clusters will be used to analyze the semantic similarity between the two documents.

3.1.3 Generating word clusters

Using WordNet, we can measure the semantic similarity or relatedness between a pair of concepts (or word senses). We use the similarity measure described in [14] which finds the path length to the root node from the least common subsumer (LCS) of the two word senses which is the most specific word sense they share as an ancestor.

Using this word similarity measure we can construct an undirected graph G. The vertices of G are the words. Two vertices are connected with an edge if the similarity measure between them exceeds a certain threshold. It was noticed that the constructed graph consists of a set of semi isolated groups when the two underlaying documents are similar. This implies that using a graph clustering algorithm would eliminate the weak intra-group edges and produce separate groups or clusters representing similar words. We used Markov Cluster Algorithm (MCL) for graph clustering [2]. MCL is a fast and scalable unsupervised cluster algorithm for graphs based on simulation of stochastic flow. An illustration of this clustering process is shown in figure 2

3.1.4 Measuring documents similarity

Measuring the document similarity is based on the fact that documents carrying the same information content tend to use similar words or words belonging to the same class. The generated word clusters can be used to judge whether two documents are similar or not. If each cluster contains words that are evenly distributed between the two documents, then the documents are more likely to be similar. On the other hand, if most of the clusters contain words that belong to a single document, then the two documents are not using similar words and hence are different.

In order to use the word clusters to judge whether two documents are similar, the clusters must reflect the membership of the word in the document, rather than the word itself. To do that, we create a new cluster for each of the word clusters. The new clusters contain labels pointing to the first or the second document, rather than words. These labels are induced from the membership of words in the documents. For example, if the word $w1$ is a member of cluster $c1$, we check the bag of words for document $d1$. If $w1 \in d1$, we add n entries to $c1$ with the label $d1$, where n is the number of times $w1$ occurred in $d1$. After this operation, clusters illustrated in figure 2 will look as shown in figure 3 using the bags of words from figure 1.

The similarity between the two documents can be estimated by measuring the entropy [4] of the generated clusters.

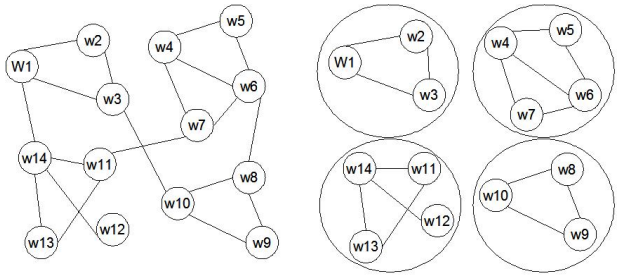

Fig. 2: *Clustering word graph*

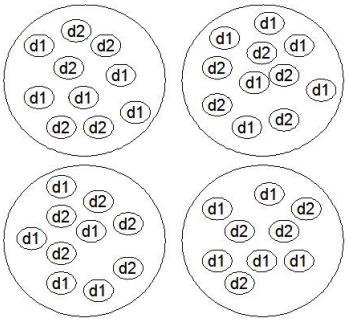

Fig. 3: *Clusters after adding documents labels*

$$entropy(c_i) = -\frac{|d_1|}{|d_1|+|d_2|} log(\frac{|d_1|}{|d_1|+|d_2|}) - \frac{|d_2|}{|d_1|+|d_2|} log(\frac{|d_2|}{|d_1|+|d_2|}) \quad (1)$$

$$Doc\_Sim = \sum_{c_i} entropy(c_i) \quad (2)$$

The cluster entropy will be high when both documents contribute equally to instances in the cluster. Hence, the summation of all clusters entropy will be high when the two documents tend to use the same classes of words. On the other hand, the summation will be low when there are several word classes that belong to a single document only. This agrees with the fact that documents conveying the same information content tend to use similar words or similar classes of words. An outline for the all the steps performed to find the best matching document is illustrated in algorithm 1.

3.2 Named Entity Similarity Model

For each candidate documents pair, the NE translation pairs extractor extracts all NE's from both documents. For each NE in language A, a list of candidate language B translations is constructed. Then, several scoring criteria are employed to select the best translation. Using the same example of English and Arabic documents, we build a list of candidate English translations for each Arabic NE, and try to select the best translation from the candidates list.

Ideally, the candidate translations list should only contain English NE's having the same type as the

Arabic NE. However, problems arise due to confusion between NE types at the NE detection phase. To overcome this, a confusion matrix is used to allow NE's with different but usually confused types to be checked. The matrix has an entry for each NE type pair that contains 1: for the same type, 0: for types that never get confused, and w: for types that are sometimes confused. w is a weighting factor that should represent how often the two types are confused. w will be multiplied with the score assigned to the English translation in order to favor translation pairs belonging to the same type.

Several similarity measures can be used to measure NE's similarity, namely: the phonetic transliteration similarity, phrase-based translation similarity, and word-based translation similarity. In the following subsections, we will describe each of the measures, and then discuss how the measures were combined.

3.2.1 Transliteration similarity

To measure the transliteration similarity between two NE's, we measure the length-normalized phonetic based edit distance between the two words. We use the *Editex* technique [17] that makes use of the phonetic characteristics of individual characters to estimate their similarity. *Editex* measures the phonetic distance between a pair of words by combining the properties of edit distances with a letter grouping strategy that groups letters with similar pronunciations.

We use a length-normalized phonetic edit distance to measure the phonetic similarity between the English NE, and a *womanized* Arabic NE. The length normalized *Editex* edit distance is given by:

$$Ed(w_1, w_2) = log(1 - \frac{Editex(w_1, w_2)}{max(|w_1|, |w_2|)}) \quad (3)$$

where $w1$, and $w2$ are two named entities, $Editex(w_1, w_2)$ is the *Editex* edit distance between w_1, and $w2$. $|w_1|$, and $|w_1|$ are the lengths of w_1, and w_2 respectively.

3.2.2 Phrase-based translation similarity

To estimate the translation similarity between two named entities, we use a phrase table like those used in machine translation systems. To measure the similarity between an English and an Arabic NE, we use an English-Arabic phrase table. Assuming that the two NE's are denoted by w_e, and w_a, we search the phrase table for all records $< P_e, P_a, Score >$ such that w_e, and w_a are substrings of P_e, and P_a respectively. Let $Score_{w_e \& w_a}$ be the sum of all scores of those records. Similarly, we calculate $Score_{w_e}$, and $Score_{w_a}$ by searching for all records such that w_e is a substring of P_e, and w_a is a substring of P_a respectively and then summing their scores.

$$Score_{w_e \& w_a} = \sum_{P_e \supset w_e \& P_a \supset w_a} Score(< P_e, P_a, Score >) \quad (4)$$

$$Score_{w_e} = \sum_{P_e \supset w_e} Score(< P_e, P_a, Score >) \quad (5)$$

$$Score_{w_a} = \sum_{P_a \supset w_a} Score(< P_e, P_a, Score >) \quad (6)$$

Hence, the phrase-based translation similarity can be expressed as:

$$PTSim(w_e, w_a) = \frac{Score_{w_e} + Score_{w_a}}{2 * Score_{w_e \& w_a}} \quad (7)$$

3.2.3 Combining Similarities

Some correct translation pairs score highly on one measure and not the other. Other correct pairs score highly on both measures. Some incorrect translation pairs may have moderate scores on both measures. Hence averaging or weighted averaging does not work well for combining scores. To combine scores, a simple *ORing* mechanism is employed. If only one measure hits, we use it. If both measures hit, we use the measure with the higher score. The score is then multiplied with the weight from the confusion matrix to favor NE's from the same type and from confusing types.

$$ts = TransSim(w_e, w_a) \quad (8)$$

$$ps = PTSim(w_e, w_a) \quad (9)$$

$$Sim(w_e, w_a) = w * max(ts, ps) \quad (10)$$

where w is a weighting factor from the confusion matrix, w_e and w_a are the English and Arabic named entities, $TransSim(w_e, w_a)$ is the transliteration similarity score, and $PTSim(w_e, w_a)$ is the phrase-based translation similarity score.

4 Extracting named entity translation pairs from parallel corpora

In this section, we show how we can make use of aligned parallel corpora to generate named entity translation pairs. We use an Arabic/English sentence level aligned parallel corpora to extract Arabic/English named entities translation pairs. The starting point for the NE pairs extraction algorithm is a word-level alignment between the parallel sentences obtained from a maximum entropy aligner similar to [7]. Word alignment was first introduced and used in the field of statistical machine translation [1], later on it was employed in several other NLP applications. The maximum entropy aligner takes an Arabic sentence ($a_{i:1 \geq i \leq m}$), and an English sentence ($e_{j:1 \geq j \leq n}$), then it generates the best alignment $a \rightarrow e$ by linking each Arabic word a_i with its most likely English translation e_j. The Arabic and English sentences are also tagged with an entity detection system similar to [3] that identifies all name, nominal, and pronoun mentions in the text.

The NE pairs extraction process starts with one of the languages, language A, and tries to find for each

name mention detected in language A a corresponding name mention in language B using the alignment information. The process is repeated starting from language B trying to find corresponding name mentions in language A. We obtain two entities alignment relations:

$$A_1 = (a, e(a)) : \forall a$$

$$A_2 = (e, a(e)) : \forall e$$

where a, and e represent all Arabic and English identified name mentions respectively.

The result of word alignment is not perfect. Hence, the pairs resulting directly from word alignment features should be handled with care. We subject the resulting pairs to several filters to exclude ill-formed pairs. The first filter excludes pairs if the length of any of their constituents exceeds a predefined threshold. This makes sure that entities that appeared due to errors in the alignment or entity identification are excluded. It is highly likely that entities forming a translation pair would have close lengths. Hence, the difference in length between the Arabic and English entities is also considered to filter out incorrect pairs.

Entities consisting of several words usually contain function words. The word alignment system usually fails to align function words between English, and Arabic sentences. To alleviate this problem, unconnected function words that lay within the entity are included in the generated pair.

The remaining pairs can be classified into three classes:

1. Pairs belonging to the intersection of the two entities alignments
2. Pairs belonging to the English/Arabic alignment only
3. Pairs belonging to the Arabic/English alignment only system

All the pairs mentioned above are added to the final translation pairs table, yet are assigned different weights. Pairs belonging to the intersection of the two entities alignments are highly precise pairs, and hence are assigned the highest weight. English entity detection annotators usually do better than their Arabic counterparts. Hence, higher confidence (weight) is given to pairs generated from the English/Arabic alignments.

5 Experimental Setup

5.1 ACE Entity Translation Evaluation

ACE is an evaluation conducted by NIST to measure Entity Detection and Tracking (EDT) and Relation Detection and Characterization (RDC). The EDT task is concerned with the detection of mentions of entities, and grouping them together by identifying their coreference. The RDC task detects relations between entities identified by the EDT task. Recently, ACE added a new Entity Translation (ET) task. The objective of the ET task is to take a document in a foreign language, and emit an English language catalog of the entities mentioned in the foreign document. We choose the ET task to show the performance of the approach we propose.

System	Precision	Recall	F-measure
Baseline	62.9	63.5	63.2
Baseline+Cmp	63.2	64.1	63.6
Baseline+Par	64.4	65.0	64.7
Baseline+.5P+.5C	63.0	63.7	63.4
Baseline+Par+Cmp	65.0	65.4	65.2

Table 1: *Precision, recall, and F measure for the baseline and modified systems*

5.2 The Baseline System

The baseline is a phrase-based statistical machine translation system. It relies on two major components: phrase translation models and DP-based phrase decoder [12]. The phrase translation pairs are extracted via word alignment, projection and extension algorithms. The baseline system was trained on LDC Arabic/English parallel corpus. The phrase-decoder utilizes different cost functions, like Translation cost, LM cost, Distortion cost, and Sentence length cost.

5.3 Test data

The test dataset comes from ACE entity translation evaluation development set. All entities in the test data were manually annotated. The test data is composed of 54 Arabic document, and 18,536 words. The test data came from both newswire and weblog data. The data contained 5911 manually annotated named entities. Among them GPE, PER, ORG, and LOC accounted for 2910, 1936, 977, and 88 entities respectively.

5.4 Experimental Results

To evaluate the system performance, we performed experiments to measure the document aligner performance and the extracted translation pairs quality.

To measure the document aligner performance, we picked a total of 380 news stories from Associated France Press (AFP) Arabic news. We also picked about 25000 news stories from English AFP stories. For each one of the Arabic documents, we used the system to find corresponding English document taking into consideration that we are not sure if an equivalent document does exist in the English side or not. Each proposed document pair proposed by the system has been evaluated, by a bilingual speaker, as follows: Similar: The two documents are similar and talking about the same event, Related : Same topic but presenting different aspects of the same news, and Not Similar: The two documents are not similar. The system proposed exact similar documents in 70% of the cases , while proposed related documents for 22% of the cases and proposed not similar documents in 8% of the cases. As related documents will most likely contain similar named entities, the total accuracy of the document aligner would be 92%.

To evaluate the effectiveness NE translation pairs extraction method, we test it on the ACE Arabic-

	Baseline			Baseline+Cmp			Baseline+Par			Baseline+0.5P+0.5C			Baseline+Par+Cmp		
Type	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GPE	68.5	67.2	67.8	67.4	65.9	66.7	68.8	67.5	68.1	67.7	66.3	67.6	70.0	68.7	69.4
LOC	31.1	37.1	33.8	29.9	36.0	32.7	30.8	37.1	33.7	30.6	37.1	33.6	31.1	37.1	33.8
ORG	32.4	32.3	32.4	32.1	32.1	32.1	32.9	32.9	32.9	32.2	32.1	32.1	32.9	32.9	32.9
PER	41.3	42.2	41.7	41.4	42.3	41.8	43.2	44.0	43.6	41.6	42.1	42.1	43.1	44.0	43.6

Table 2: *Results for different NE types*

English entity translation task. We extracted named entity translation pairs from comparable and parallel corpora and used them to improve the baseline system described above. Precision, recall, and F-measure for the baseline system, baseline with translation pairs from comparable data, baseline with translation pairs from parallel data, baseline with 50% of all data, and baseline with all data are shown in table 1. We notice from the table that both the precision and recall have been improved by the addition of the translation pairs extracted from parallel data, comparable data, or both.

To further study the system performance, we calculated the precision, recall, and F-measure for all systems described above for each named entity type. Those results are shown in table 2. GPE, LOC,PER, and ORG stand for geographical political entity, location, person, and organization respectively. We notice from the results that the system achieves better improvement for the GPE and PER types. It achieves less improvement for LOC, and ORG types. This is due to the low percentage of LOC and ORG NEs in the training and test data compared to that of the GPE, and PER NEs. In addition, ORG NEs are usually semantically translated word-by-word. This is usually done well by the word and phrase translation components in the baseline system.

6 Conclusion

Several natural language processing applications need a robust named entity translation system. Such system would certainly benefit from the existence of a named entity dictionary or translation list. In this paper, we presented an approach for exploiting both comparable and parallel corpora to extract named entity translation pairs. We used this approach to build a large dictionary of Arabic/English named entity translation pairs. The quality of the extracted pairs was evaluated by showing that it improves the performance of an ACE entity translation system.

References

[1] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelnik, J. Lafferty, R. Mercer, and P. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85, 1990.

[2] S. V. Dongen. A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

[3] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. A statistical model for multilingual entity detection and tracking. In *Proceedings of Human Language Technology Conference and The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, 2004.

[4] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Math Programming*, 79:191–215, 1997.

[5] F. Huang, S. Vogel, and A. Waibel. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Workshop on Multilingual and Mixed Language Named Entity Recognition*, 2002.

[6] F. Huang, Y. Zhang, and S. Vogel. Mining key phrase translations from web corpora. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

[7] A. Ittycheriah and S. Roukos. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

[8] A. Klementiev and D. Roth. Named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2006.

[9] R. Moore. Learning translation of named entity phrases from parallel corpora. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, 2003.

[10] Y. Shinyama and S. Sekine. Named entity discovery using comparable news articles. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2004.

[11] T. Strohman, D. Metzler, H. Turtle, and W. Croft. Indri: A language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2005.

[12] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29:79–133, 2003.

[13] C. Van Rijsbergen, S. Robertson, and M. Porter. New models in probabilistic information retrieval. British Library Research and Development Report, no. 5587, 1980.

[14] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 1994.

[15] Y. Zhang, F. Huang, and S. Vogel. Mining translations of oov terms from the web through cross-lingual query expansion. In *Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2005.

[16] Y. Zhang and P. Vines. Detection and translation of oov terms prior to query time. In *Proceedings of the 27th Annual International ACM SIGIR,*, 2004.

[17] J. Zobel and P. Dart. Phonetic string matching: Lessons from information retrieval. In *Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 1996.