Named Entity Translation: Extended Abstract

Yaser Al-Onaizan Kevin Knight

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{yaser,knight}@isi.edu

1 Introduction

Named entity phrases are being introduced in news stories on a daily basis in the form of personal names, organizations, locations, temporal phrases, and monetary expressions. While the identification of named entities in text has received significant attention (e.g., (Mikheev et al., 1999) and (Bikel et al., 1999)), translation of all named entities has not. This translation problem is especially challenging because new phrases can appear out of nowhere, and because many are domain specific, not to be found in bilingual dictionaries.

In this paper, we describe a system for Arabic-English named entity translation, though the technique is applicable to any language and does not require especially difficult-to-obtain resources.

2 Producing Translation Candidates

Our translation is carried out by first generating a list of translation candidates, as will be described in this section, then re-scoring them using different monolingual clues (Section 3).

Named entity phrases can be identified fairly accurately. In addition to identifying phrase boundaries, systems also provide the category and sub-category of the phrase (e.g., **ENTITY NAME**, and **PERSON**). Different types of named entities are translated differently and hence our candidate generator has a specialized module for each type. Numerical and temporal expressions typically use a limited set of vocabulary words (e.g., names of months, days of the week) and can be translated fairly easily using simple translation patterns. Therefore, we will not address them in this paper. Instead we will focus on person names, locations, and organizations. But before we present further details, we will discuss how words can be transliterated (i.e., "sounded-out").

2.1 Transliteration

Transliteration is the process of replacing words in the source language with their approximate phonetic or spelling equivalents in the target language. Transliteration between languages that use similar alphabets and sound systems is very simple. However, transliterating names from Arabic into English is a non-trivial task, mainly due to the differences in their sound and writing systems. Short vowels are rarely written in Arabic in newspaper text, which makes pronunciation highly ambiguous. Also, there is no one-to-one correspondence between Arabic sounds and English sounds. For example, English **P** and **B** are both mapped into Arabic "ب"; Arabic "ح" and "هـ" into English **H**; and so on.

(Stalls and Knight, 1998) present an Arabic-to-English back-transliteration system based on the source-channel framework. The transliteration process is based on a generative model of how an English name is transliterated into Arabic. It consists of several steps, each is defined as a probabilistic model represented as a finite state machine. First, an English word is generated according to its unigram probabilities $P(w)$. Then, the English word is pronounced with probability $P(e|w)$, which is collected directly from an English pronunciation dictionary. Finally, the English phoneme sequence is converted into Arabic writing with probability $P(a|e)$.

One serious limitation of this method is that only English words with known pronunciations can be produced. Also, human translators often transliterate words based on how they are spelled in the source language. For example, *Graham* is transliterated into Arabic as "غراهام" and not as "غرام". To address these limitations, we extend this approach by using a new spelling-based model in addition to the phonetic-based model.

The spelling-based model we propose directly maps English letter sequences into Arabic letter sequences $P(a|w)$, which are trained on a small English/Arabic name list without the need for English pronunciations. Since no pronunciations are needed, this list is easily obtainable for many language pairs. We also extend the $P(w)$ to include a letter trigram model in addition to the word unigram model. This makes it possible to generate words that are not already defined in the word unigram model.

2.2 Producing Candidates for Person Names

Person names are almost always transliterated. The translation candidates for typical person names are generated using a transliteration module. For example, the name "بيل كلينتون" is transliterated into: *Bell Clinton*, *Bill Clinton*, *Bill Klington*, etc. Finite-state devices produce lattices containing the candidates, as well as n-best lists.

2.3 Producing Candidates for Location and Organization Names

Words in organization and location names, on the other hand, are either translated or transliterated, and it is not clear when a word must be translated and when it must be transliterated. So to generate translation candidates for a given phrase, words in the phrase are first translated using a bilingual dictionary and they are also transliterated. Our candidate generator combines the dictionary entries and k-best transliterations for each word in the given phrase into a regular expression that accepts all possible permutations of word translation/transliteration combinations. This regular expression is then matched against a large English news corpus. All matches are then scored according to their individual word translation/transliteration scores. The scored matches form the list of translation candidates. For example, the candidate list for "خليج الخنازير" includes *Bay of Pigs* and *Gulf of Pigs*.

3 Re-Scoring Candidates

Once a ranked list of translation candidates is generated for a given phrase, several monolingual English resources are used to help re-rank the list. The first re-scoring factor we use is the normalized straight Web count. For the "بيل كلينتون" example, the top two translation candidates are *Bell Clinton* with transliteration score 1.1×10^{-09} and *Bill Clinton* with score 6.7×10^{-10}. The Web frequency counts of these two names are: 146 and $840,844$ respectively. This gives us revised scores of 1.9×10^{-13} and 6.68×10^{-10}, respectively, which leads to the correct translation being ranked highest.

In some cases straight Web counting does not help the re-scoring. For example, the top two translation candidates for "دونالد مارون" are *Donald Martin* and *Donald Marron*. Their straight Web counts are 2992 and 2509, respectively. These counts do not change the candidates list ranking. We next seek a more accurate counting method by counting phrases only if they appear within a certain context. Using search engines, this can be done using the boolean operator **AND**. For the previous example, we use the fact that the person mentioned is the *CEO* of *Paine Webber*. In this case we get the counts 0 and 357 for *Donald Martin* and *Donald Marron*, respectively. This is enough to get the correct translation as the top candidate.

4 Extending the Candidates List

The re-scoring methods described above assume that the correct translation is in the candidates list. When it is not in the list, the re-scoring will fail. To address these situations, we need to be able to extrapolate from the candidate list. We do this by searching for the correct translation rather than generating it. We do that by using sub-phrases from the candidates list or by searching for documents in the target language similar to the one being translated. For example, for a person name, instead of searching for the full name, we search for the first name and the last name separately. Then, we use the IdentiFinder named entity identifier (Bikel et al., 1999) to identify all named entities in the top n retrieved documents for each sub-phrase. All named entities of the type of the named entity in question (e.g., PERSON) found in the retrieved documents and that contain the sub-phrase used in the search are added to the list of translation candidates, and the re-scoring is repeated.

To illustrate this method, consider the name "كوفي عنان." Our translation module proposes: *Coffee Annan*, *Coffee Engen*, *Coffee Anton*, *Coffee Anyone*, and *Covey Annan* but not the correct translation **Kofi Annan**. Using a search engine, we retrieve the top n matching documents for each of the names *Coffee*, *Covey*, *Annan*, *Engen*, *Anton*, and *Anyone*. All person names found in the retrieved documents that contain any of the first or last names we used in the search are added to the list of translation candidates. We hope that the correct translation is among the names found in the retrieved documents. The re-scoring procedure is applied once more on the expanded candidates list. In this example, this leads to the correct translation being ranked as top.

To address cases where neither the correct translation nor any of its sub-phrases can be found in the list of translation candidates, we also perform the above extrapolation procedure using contextual information such as the title of the original document to find similar documents in the target language. This is especially useful when translating named entities in news stories of international importance where the same event will most likely be reported in many languages including the target language.

5 Evaluation

This section presents our preliminary evaluation results. Our evaluation corpus consists of 21 Arabic newspaper articles with named entity phrases hand-tagged and translated to English. The Arabic phrases are paired with their English translations to create the gold-standard translation. In order to evaluate human performance at this task, we asked a bilingual speaker (a native of Arabic) to translate the Arabic named entity phrases given the text they appear in. The errors made by the original human translator turned out to be numerous, ranging from simple spelling errors (e.g., *Custa Rica* vs. *Costa Rica*) to more serious errors such as transliteration errors (e.g., *John Keele* vs. *Jon Kyl*) and other translation errors (e.g., *Union Reserve Council* vs. *Federal Reserve Board*). The error rates in the human translations were 40%, 13.9%, and 28.3% for person, location, and organization names respectively. The overall error rate was 26.3%.

The translations obtained by our system show promising results. The error rates for the results obtained by the candidate generator were 40%, 45%, and 68.3% for person, location, and organization names respectively. After the re-ranking procedure is applied, the error rates were reduced to 22.8%, 31%, and 56.7% respectively. If we measure how often the correct answer is in the top-20 list of proposed candidates, error rates are 15.2%, 29.5%, and 45%, respectively.

Bibliography

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1/3).

Nancy Chinchor. 1997. MUC-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference. http://www.muc.saic.com/*.

Andrei Mikheev, Marc Moens, and Calire Grover. 1999. Named entity recognition without gazetteers. In *Proceedings of the EACL*.

Bonnie G. Stalls and Kevin Knight. 1998. Translating names and technical terms in Arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.