Computational Linguistics and Chinese Language Processing
Vol. 18, No. 1, March 2013, pp. 19-46

Learning to Find Translations and Transliterations on the Web based on Conditional Random Fields

Joseph Z. Chang[*], Jason S. Chang[+], and Jyh-Shing Roger Jang[#]

Abstract

In recent years, state-of-the-art cross-linguistic systems have been based on parallel corpora. Nevertheless, it is difficult at times to find translations of a certain technical term or named entity even with a very large parallel corpora. In this paper, we present a new method for learning to find translations on the Web for a given term. In our approach, we use a small set of terms and translations to obtain mixed-code snippets returned by a search engine. We then automatically annotate the data with translation tags, automatically generate features to augment the tagged data, and automatically train a conditional random fields model for identifying translations. At runtime, we obtain mixed-code webpages containing the given term and run the model to extract translations as output. Preliminary experiments and evaluation results show our method cleanly combines various features, resulting in a system that outperforms previous works.

Keywords: Machine Translation, Cross-lingual Information Extraction, Wikipedia, Conditional Random Fields.

1. Introduction

The phrase translation problem is critical to many cross-language tasks, including statistical machine translation, cross-lingual information retrieval, and multilingual terminology (Bian & Chen, 2000; Kupiec, 1993). Such systems typically use a bilingual lexicon or a parallel corpus to obtain phrase translations. Nevertheless, the out of vocabulary problem (OOV) is difficult to overcome, even with a very large training corpus, due to the Zipf nature of word

[*] Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan
E-mail: joseph.nthu.tw@gmail.com
The author for correspondence is Joseph Z. Chang.
[+] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
E-mail: jason.jschang@gmail.com
[#] Department of Computer Sciences and Information Engineering, National Taiwan University, Taiwan
E-mail: jang@csie.ntu.edu.tw

distribution and the fact that new words, technical terms, and named entities arise frequently. On the other hand, the advent of the Internet has lead to an unprecedented buildup of multilingual texts. Specifically, there are an abundance of webpages consisting of mixed-code text, namely text written in more than one language. We observe that the mixed-code webpages typically are written in one language but interspersed with some sentential or phrasal translations written in another language. By retrieving and identifying such translation counterparts on the Web, we can cope with the OOV problem caused by the limited coverage of dictionaries and parallel corpora.

Consider a Wikipedia title, "*Named-entity recognition*". The best places to find the Chinese translations for this technical term are probably not some parallel corpus or dictionary, but rather mixed-code webpages that mention it in both Chinese and English. The following example is a snippet returned by the *Bing* search engine for the query "*named entity recognition*" requesting Chinese language webpages:

> http://zh.wikipedia.org/zh-hk/問答系統: 從系統內部來看,問答系統使用了大量有別於傳統資訊檢索系統自然語言處理技術，如自然語言剖析(Natural Language Parsing)、問題分類(Question Classification)、專名辨識(Named Entity Recognition)等等。

In this snippets, the author mentioned several technical terms in Chinese (*e.g.*, 自然語言剖析 *zhiran yuyan poxi*, 問題分類 *wenti fenlei*, and 專名辨識 *zhuanming bianshi*), followed by the source terms in brackets (*Natural Language Parsing*, *Question Classification*, *and Named Entity Recognition*, respectively). The term-translation pairs in the above example follow the parenthetical translation surface pattern in the form of "*Chinese translation (English term)*". This pattern is only one of many surface patterns found on the Web that may indicate a term-translation pair. In the following examples, we show different surface patterns of translation pairs found on the Web, with Chinese translations underlined and the counterpart English terms italicized:

(a) 血液學檢驗(*hematology*) – 白血球分類

(b) [巴黎最美的橋] 亞歷山大三世橋 *Pont Alexandre III*

(c) 胰島素泵的臨床應用及護理進展 progress on nursing of clinical application of *insulin pump*

(d) 國外組織美國職棒大聯盟 (*Major League Baseball*，簡稱: *MLB*，或大聯盟)

(e) [食記]義美蔥油餅 Imei *green onion pancake*

(f) [食記]義美蔥油餅 *Imei* green onion pancake . . .

Examples (a) and (b) show Chinese translations occurring near or next to an English phrase. There are also cases (*e.g.*, Example (c)) where the translation (*e.g.*,胰島素泵 yidaoshu pang) and the English phrase (*e.g., insulin pump*) are far apart. Example (d) shows another form of parenthetical translation pattern, where translations are right next to the English term (*Major League Baseball*). Examples (e) and (f) show two term translation pairs interwoven in the same text (義美 yi-mei transliterated into *Imei* and 蔥油餅 cong-you-bing translated into *green onion pancake*).

For a given English term, such translations can be extracted by classifying the Chinese characters in the snippets as either translation or otherwise. Intuitively, we can cast the problem as a sequence labeling task. To be effective, we need to associate each token (*i.e.*, Chinese character or word) with some features to characterize the likelihood of the token being part of the translation. For example, by exploiting some external knowledge sources (*e.g.*, bilingual dictionaries), we derive that the Chinese character "辨" (*bian*) in the Chinese word "辨識" (*bian-shi, recognition*) is likely to be part of the translation of "*named entity recognition.*"

In this paper, we present a new method that automatically obtains such labeled data and generates features for training a conditional random fields (CRF) model that is capable of identifying translation or transliteration in mixed-code snippets returned by search engines (*e.g., Google* or *Bing*). The system uses a small set of phrase-translation pairs to obtain search engine snippets that may contain both an English term and its Chinese translation from search engines. The snippets then are tagged automatically to train a CRF sequence labeler. We describe the training process in more detail in Section 4.

At run-time, we start with a given phrase (*e.g.*, "*named-entity recognition*"), which is transformed into a query with a setup to retrieve webpages in the target language (*e.g.*, Chinese). We then retrieve mixed-code snippets returned by the search engine and extract translations within the snippets. The identified translations can be used to supplement a bilingual terminology bank (*e.g.*, adding multilingual titles to existing Wikipedia); alternatively, they can be used as additional training data for a machine translation system, as described in Lin, Zhao, Van Durme, and Paşca (2008).

Most previous works focus on extracting translation pairs where the counterpart terms appear near one another in the webpage, based on a limited set of short patterns. In our approach, we extract term and translation pairs that are near or far apart, and are not limited by a set of predefined patterns. We have evaluated our method based on English-Chinese

language links in Wikipedia as the gold standard. Results show that our method produces output for 80% of the test cases with an exact match precision of 43%, outperforming previous works.

The rest of the paper is organized as follows. In the next Section 2, we survey the related work that also aimed to mine translations from the Web. In Section 3, we give brief descriptions on resources we make use of. In Section 4, we describe in detail the problem statement and the proposed method. Finally, we report evaluation results and error analysis in Section 5.

2. Related Work

In machine translation, a source text is typically translated one sentence at a time, while cross-lingual information retrieval involves phrasal translation. The proposed methods for phrase translation in the literature rely on either handcrafted bilingual dictionaries, transliteration tables, or bilingual corpora. For example, Knight and Graehl (1998) described and evaluated a multi-stage machine translation method for performing backwards transliteration of Japanese names and technical terms into English, while Bian and Chen (2000) described cross-language information access to multilingual collections on the Internet. Recently, Smadja, McKeown, and Hatzivassiloglou (1996) proposed an algorithm for producing collocation and translation pairs, including noun and verb phrases, in bilingual corpora. Similarly, Kupiec (1993) propose an algorithm for finding noun phrase correspondence in bilingual corpora for bilingual lexicography and machine translation. Koehn and Knight (2003) described a noun phrase translation subsystem that improves word-based statistical machine translation methods.

Some methods in the literature also have aimed to exploit mixed code webpages for word and phrase translation. Nagata, Saito, and Suzuki (2001) presented a system for finding English translations for a given Japanese technical term in search engine results. Their method extracts English phrases appearing near the given Japanese term, and it scores translation candidates based on co-occurrence counts and location. Cao and Li (2002) proposed an EM algorithm for finding translation for base noun phrases on the Web. Kwok *et al*. (2005) focused on named entity phrases and implemented a cross-lingual name finder based on Chinese-English webpages. Wu, Lin, and Chang (2005) proposed a method for learning a set of surface patterns to find terms and translations occurring in short distance. Mixed-code webpage snippets were obtained by querying a search engine with English terms for Chinese webpages. They discovered that the most frequent pattern is where the translation immediately followed by the source term, with the coverage rate of 46%. Their results also indicate the stricter parenthetical pattern covers less than 30% of the translation instances.

Researchers also have explored the hyperlinks in webpages as a source of bilingual

information. Lu, Chien, and Lee (2004) proposed a method for mining terms and translations from anchor text directly or transitively. In a follow-up project, Cheng *et al*. (2004) proposed a method for translating unknown queries with web corpora for cross-language information retrieval. Similarly, Gravano and Henzinger (2006) also proposed systems and methods for using anchor text as parallel corpora for cross-language information retrieval.

In a study more closely related to our work, Lin *et al.* (2008) proposed a method that performs word alignment between Chinese translations and English phrases within parentheses in crawled webpages. Their paper also proposed a novel and automatic evaluation method based on Wikipedia. The main difference from our work is that the alignment process in Lin *et al*. (2008) is done heuristically using a competitive linking algorithm proposed by Melamed (2000), while we use a learning-based approach to align words and phrases. Moreover, in their method, only *parenthetical translations* are considered. With only the parenthetical pattern, their method is able to extract a significant number of translation pairs from crawled webpages without a given list of target English phrases. By restricting to parenthetical surface patterns however, many translation pairs in webpages may not be captured, including term-translation pairs that are further apart. In our work, we exploit surface patterns differently as a soft constraint in a CRF model and use an approach similar to Lin *et al*. (2008) to evaluate our results.

In contrast to the previous work in phrase and query translation, we present a learning-based approach that uses annotated data to develop the system. Nevertheless, we do not require human intervention to prepare the training data, but instead make use of language links in Wikipedia to automatically obtain the training data. The annotated data is further augmented with features indicative of translation and transliteration relations obtained from external lexical knowledge sources publicly-available on the Web. The trained CRF sequence labeler then is used to find translations on the Web for a given term.

3. Resources

In this work, we rely on several resources that are available on the Internet. These resources are used for different purposes: the seed data are used for obtaining and labeling training data, the gold standard is used for automatic evaluation, and the external knowledge sources are used for generating features.

3.1 Wikipedia

Wikipedia is an online encyclopedia compiled by volunteers around the world. Anyone on the Internet can edit existing entries or create new entries to add to Wikipedia. Owing to the number of its participants, Wikipedia has achieved both high quantity and a quality comparable to traditional encyclopedias compiled by experts (Giles, 2005). Due to these

reasons, Wikipedia has become the largest and most popular reference tool.

We extracted bilingual title pairs from the English and Chinese editions of Wikipedia as the gold standard for evaluation and as seeds to automatically collect and label training data from the Internet by querying search engines.

The number of entries in English Wikipedia grew at an exponential rate from 2001 to 2008, with some 20,000 new articles created monthly by thousands of volunteers around the world, making it an excellent source for finding new words and terms. As of February 2, 2012, the English Wikipedia had 3,861,652 articles, making it the most well-established edition for all 285 languages.

Entries on the same topic among different language editions of Wikipedia are interlinked via the so-called language links. Nevertheless, only a small percentage of English articles are linked to editions of other languages. The Chinese Wikipedia contains only 398,206 articles, making it roughly one-tenth the size of the English Wikipedia. Furthermore, only 5% of the entries in the English Wikipedia contain language links to their Chinese counterparts. The proposed method can be used to find the translations of those English terms, thus speeding up the process of building a more complete multilingual Wikipedia. As will be described in Section 4, we extracted the titles of English-Chinese article pairs connected by language links for training and testing purposes.

The content of Wikipedia is freely downloadable online.[1] We used the *Google Freebase Wikipedia Extraction* (*WEX*) instead of the official raw dump. The *WEX* is a processed version of the official dump, with the Wikipedia syntax transformed into XML. The *WEX* database can be freely downloaded online.[2]

3.2 WordNet

WordNet is a freely available, handcrafted lexical semantic database for English.[3] Starting its development in 1985 at Princeton University by a team of cognition scientists, WordNet was originally intended to support psycho-linguistic research. Over the years, WordNet has become increasingly popular in the fields of information retrieval, natural language processing, and artificial intelligent. Through each release, WordNet has grown into a comprehensive database of concepts in the English language. As of today, the stable 3.0 version of WordNet contains 207,000 semantic relations between 150,000 words organized in over 115,000 senses.

Senses inWordNet are represented as synonym sets (*synsets*). A synset with a definition contains one or more words, or *lemmas*, that express the same meaning. In addition, WordNet

[1] http://en.wikipedia.org/wiki/Wikipedia:Database_download

[2] http://wiki.freebase.com/wiki/WEX

[3] http://wordnet.princeton.edu/

provides other information for each synset, including example sentences and estimated frequency. For example, the synset {*block, city_block*} is defined as *a rectangular area in a city surrounded by streets*, whereas synset {*block, cube*} is defined as *a three-dimensional shape with six square or rectangular sides*. WordNet also records various semantic relations between its senses. These relations includes *hypernyms*, *hyponyms*, *coordinate terms*, *holonym* and *meronym*.

3.3 Sinica Bilingual WordNet

The *Sinica Bilingual WordNet* is part of the publicly accessible *Sinica Bilingual Ontological WordNet* (*Sinica BOW*) (Huang, 2003). In this work, we treat the *Sinica Bilingual WordNet* as a bilingual dictionary, and use it as an external knowledge source to generate features for training the CRF model.

The Sinica Bilingual WordNet is a hand-crafted English-Chinese version of the original Princeton WordNet 1.6. It was compiled by collecting all possible Chinese translations of a synset's lemmas from various online bilingual dictionaries before a team of translators manually edited the acquired translations. For each synset, the translators selected at most three appropriate lexicalized words as translation equivalents.

The *Sinica BOW* system can be freely-accessible online.[4] The *Sinica Bilingual WordNet* database can also be licensed for download.[5]

3.4 NICT Bilingual Technical Term Database

The *NICT Bilingual Technical Term Database* is a resource freely available online.[6] In addition to the *Sinica Bilingual WordNet*, we also used the NICT database to generate features. While the Sinica Bilingual WordNet mainly contains common nouns, the NICT database mainly contains technical terms and proper nouns. By combining the two resources, we can generate translational features covering both common nouns and proper nouns.

The NICT Bilingual Technical Term Database is maintained by committees in the National Academy for Educational Research of Taiwan (formerly National Institute for Compilation and Translation). The goal is to pursuit more uniform and standardized translations for technical terms used in textbooks, patents, national standards, and open source software. It contains over 1.1 million Chinese-English term translation pairs arranged into 72 categories (Table 9) and is kept up to date by constantly including more terms. Any user can suggest a new term and translation to the committees to be added to the database.

---

[4] http://BOW.sinica.edu.tw/

[5] http://www.aclclp.org.tw/doc/bw_agr_e.PDF

[6] http://terms.nict.gov.tw/

3.5 Google Web 1T N-grams

In 2006, *Google* published a ngram dataset based on public webpages through Linguistics Data Consortium for licensing.[7] The Google Web 1T corpus is a 24 GB (gzip compressed) corpus that consists of n-grams ranging from unigram to five-grams generated from approximately 1 trillion words in publicly accessible Web pages. In this work, we use the Web 1T corpus to filter unlinked entries in the English Wikipedia with high frequency on the Web for manual evaluation.

4. Method

Submitting an English phrase (*e.g.*, "*named-entity recognition*") to search engines to find translations or transliteration is a good strategy used by many translators (Quah, 2006). Unfortunately, the user has to sift through snippets to find the translations. Such translations usually exhibit characteristics related to word translation, word transliteration, surface patterns, and proximity to the occurrences of the given phrase. To find translations for a given term on the Web, a promising approach is automatically learning to extract phrasal translations or transliterations of a given query using the conditional random fields (CRF) model. To avoid human effort in preparing annotated data for training the model, we use an automatic procedure to retrieve and tag mixed-code search engine snippets using a set of bilingual Wikipedia titles. We also propose using external knowledge sources (*i.e.*, bilingual dictionaries, name lists and terminology banks) to generate translational and transliterational features.

4.1 Problem Statement

We focus on the issue of finding translations in mixed code snippets returned by a search engine. The translations are identified, tallied, ranked, and returned as the output of the system. The returned translations can be used to supplement existing multilingual terminology banks, or used as additional training data for a machine translation system. Therefore, our goal is to return several reasonably precise translations that are available on the Web for the given phrase.

Problem Statement: Given a phrasal term *P* and a full-text search engine *SE* (*e.g.*, *Bing* or *Google*) that operates over a mixed-code document collection (*e.g.*, the Web), our goal is to retrieve a probable translation *T* of *P* via *SE*.

For this, we extract a set of translation candidates, c_1, ..., c_m from a set of mixed-code snippets, s_1, ..., s_n returned by *SE*, such that these candidates are likely to be translations *T* of *P*.

[7] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13

(1) Retrieve mixed-code snippets and tag translations (Section 4.3.1)
(2) Generate translation features (Section 4.3.2)
(3) Generate transliteration features (Section 4.3.3)
(4) Generate distance features (Section 4.3.4)
(5) Train a CRF model for classifying translations (Section 4.3.4)

Figure 1. Outline of the training phase.

In the rest of this section, we describe our solution to this problem. First, we briefly introduce the Conditional Random Fields (CRF) model in Section 4.2. We describe a strategy (see Figure 1) for obtaining training data for identifying translation in snippets returned by *SE* (Section 4.3.2). This strategy relies on a set of term-translation pairs for training, derived from Wikipedia language links (Section 4.3.1). We will also describe our method for exploiting external knowledge sources to generate translation features (Section 4.3.2), transliteration features (Section 4.3.3), and distance features (Section 4.3.4) for sequence labeling. Finally, in Section 4.4, we describe how to extract and filter translations at run-time by applying the trained sequence labeler.

4.2 Conditional Random Fields

Sequence labeling is the task of assigning labels from a finite set of categories to a sequence of observations. This problem is encountered in the field of computational linguistics, as well as in many other fields, including bio-informatics, speech recognition, and pattern recognition.

Traditionally, the sequence labeling problem are often solved using the Hidden Markov Model (HMM) or Maximum Entropy Markov Model (MEMM). Both HMM and MEMM are directed graph models in which every outcome is conditioned on the corresponding observation node and the previous outcomes (*i.e.*, Markov property).

Conditional Random Fields (CRF), proposed by Lafferty, McCallum, and Pereira (2001), is considered the state-of-the-art sequence labeling algorithm. One of the major differences of CRF is that it is modeled as an undirected graph. For sequence labeling, the CRF graph is structured as an undirected linear chain (chained CRF). CRF obeys the Markov property with respect to the undirected graph, as every outcome is conditioned on its neighboring outcomes and potentially the entire observation sequence. In our case, the outcomes are B, I, O labels that indicate a sequence of Chinese characters in the search engine snippets that is likely the translation or transliteration of the given English term. The information available (the observable) for sequence labeling are the characters in the snippets themselves, and the three types of features we generate.

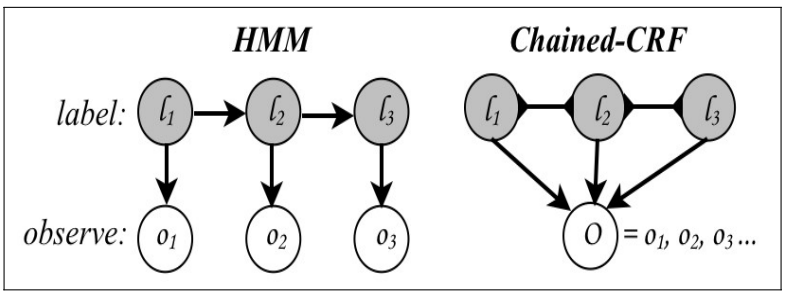

Figure 2. Simplified view of HMM and CRF.

4.3 Preparing Data for CRF Classifier

We attempt to learn to find translations or transliterations for given phrases on the Web. For this, we make use of language links in Wikipedia to obtain seed data, retrieve mixed-code snippets returned by a search engine, and augment feature values based on external knowledge sources. Our learning process is shown in Figure 1.

4.3.1 Retrieving and Tagging Snippets

In the first stage of the training phase, we extracted Wikipedia English titles and their Chinese counterparts using the language links as the seed data for training. We use the English titles to query a search engine (*e.g.*, *Google* or *Bing*) with the target Web page language set to Chinese. This strategy will bias the search engine to return Chinese web pages interspersed with some English phrases. We then automatically labeled each Chinese character in the returned snippets, using the common *BIO* notation, with *B*, *I*, *O* indicating the beginning, inside, and outside of translations, respectively (*e.g.*, 支援向量機 *zhiyuan-xiangliang-ji*). An additional *E* tag is used to indicate the occurrences of the given term (*e.g.*, *support vector machine*).

1. …1995/O 年/O 提/O 出/O 的/O **支/B 持/I 向/I 量/I 機/I** (/O *support/E vector/E machine/E*，/O SVM/O)/O 以/O 訓/O 練/O …
2. …發/O 光/O 原/O 理/O 不/O 同/O。/O **光/B 通/I 量/I** *luminous/E flux/E* 光/O 源/O 在/O 單/O 位/O 時/O 間/O …

***Figure 3. Examples of tagged snippets for title pairs* "support vector machine", "支持向量機" *and* "luminous flux", "光通量".**

The output of this stage is a set of tagged snippets that can be used to train a statistical sequence classifier for identifying translations. A sample of two tagged snippets, automatically generated from bilingual Wikipedia titles are shown in Figure 3. The *E* tags are designed to provide proximity cues for labeling the translation and capture common surface patterns of the phrase and translation in mixed code data. For example, in Figure 3, the translation 支持向量機(*zhichi xiangliang ji*) is tagged with one B tag and four I tags,

followed by the left parenthesis and three *E* tags. The translation 光通量 (*guangtong liang*) is tagged with one *B* tag and two *I* tags, immediately followed by two *E* tags. Such sequences (*i.e. B I I I I O E E E*, and *B I I E E*) are two of many common patterns.

Note that we do not attempt to produce word alignment information, as done in Lin *et al.* (2008). In contrast, we only use the BIO labeling scheme to indicate phrasal translations, leading to a smaller number of parameters required to be estimated during the training process.

4.3.2 Generating Translation Features

We generate translation features using external bilingual resources with the ϕ^2 score proposed by Gale and Church (1991) to measure the correlations between an English word and a Chinese character:

$$\phi^2 = \frac{[P(e,f)P(\bar{e},\bar{f}) - P(\bar{e},f)P(e,\bar{f})]^2}{P(e)P(f)P(\bar{e})P(\bar{f})} \quad (1)$$

where *e* is an English word and *f* is a Chinese character occurring in bilingual phrase pairs.

Table 1. Example of a Chinese-English dictionary with three entries.

Chinese	English
社交工程	social engineering
社群網路	social network
社群媒體	social media

Table 2. Example of English word and Chinese character probability.

w	Count(w)	P(w)	P($\bar{w}$)	e	f	Count(e,f)	P(e,f)
社	3	1.00	0.00	social	社	3	1.00
群	2	0.67	0.33	social	群	2	0.67
交	1	0.33	0.67	social	交	1	0.33
網	1	0.33	0.67	network	社	1	0.33
social	3	1.00	0.00	network	群	1	0.33
media	1	0.33	0.67	network	交	0	0.00
network	1	0.33	0.67	network	網	1	0.33

In our case, the ϕ^2 scores are calculated by counting the occurrence of Chinese characters and English words in the publicly-available bilingual dictionaries or termbanks. To illustrate, we use a tiny Chinese-English dictionary in Table 1 with only three entries to explain how the probabilities are calculated. We treat each entry in the dictionary as an event, and calculate the probability of each Chinese character and English word by counting the

number of events containing them, as shown in Table 2. Similarly, we can calculate the joint probability of an English word and a Chinese character by counting their co-occurrences in the dictionary.

Table 3. Three contingency tables indicating co-occurrence and none co-occurrence.

	vector	
向	793	9,960
	97	1,975,642

	vector	
量	768	21,907
	122	1,963,695

	machine	
機	3,381	28,566
	491	1,954,054

In Table 3, we show the contingency table calculated by counting co-occurrences in Bilingual WordNet and NICT termbank for (向 *xiang*, *vector*), (量 *liang*, *vector*), and (機 *ji*, *machine*). The statistical association between an English word (*e.g.*, *vector*) and its translation (*e.g.*, 向 (*xiang*)) is indicated by the high count of co-occurrences, as well as the lower values of two inverse diagonal cells. From the contingency tables, we can calculate the corresponding ϕ^2 scores for 向 *xiang* 量 *liang*, and 機 *ji*: 0.06530, 0.02880, and 0.09068.

Table 4. Example ϕ^2 scores.

	support	vector	machine
提	0.00000	0.00000	0.00000
出	0.00000	0.00000	0.00000
的	0.00000	0.00000	0.00000
支	**0.09075**	0.00000	0.00000
持	**0.00058**	0.00000	0.00000
向	0.00000	**0.06530**	0.00000
量	0.00000	**0.02880**	0.00000
機	0.00000	0.00000	**0.09067**

	luminous	flux
發	**0.00432**	0.00000
光	**0.01028**	6.0E-06
原	0.00000	0.00000
理	0.00000	0.00000
不	1.4E-06	0.00000
光	**0.01028**	6.0E-06
通	0.00000	**0.06410**
量	0.00000	**0.00793**

To generate features for each token, we calculate the following logarithmic value of ϕ^2:

$$feat_{translation}(f) = 9 + log(\underset{e \in E}{argmax}\phi^2(e, f)) \quad (2)$$

where *e* is a word in the given English phrase *E*, and *f* is the Chinese character in a snippet. This feature value is rounded to a whole number in order to limit the number of distinct feature values. In Table 4, we show the ϕ^2 scores of each Chinese character in snippets from searching Google with the given terms, *i.e.*, *support vector machine* and *luminous flux*. Notice that there are some noisy feature values in the second example: the Chinese characters in the word 發光 (*faguang*, *glow* or *illuminate*) has non-zero ϕ^2 scores. However, the tagger potentially can overcome such noise by relying on other features, such as the distance feature (Section 4.3.4). Moreover, in most cases there are multiple snippets for a given term, from which we can confidently identify the translations with higher frequencies. As an example, we

show two snippets tagged with translation features in Figure 4. In this example, the translation characters are given feature values ranging from 2 to 7, while non-translation ones are mostly 0.

1. ... 1995/0 年/0 提/0 出/0 的/0 **支/7 持/2 向/6 量/5 機/7** (/0 *support*/*E* *vector*/*E machine*/*E*，/0 SVM/0)/0 以/0 訓/0 練/0 ...
2. ... 發/0 光/5 原/0 理/0 不/0 同/0 。/0 **光/5 通/7 量/5** *luminous*/*E* *flux*/*E* 光/5 源/0 在/0 單/0 位/0 時/0 間/0 ...

***Figure 4. Example of two snippets tagged with translation features given the terms* "support vector machine" *and* "luminous flux".**

4.3.3 Generating Transliteration Features

We generate the additional features related to transliteration using some external knowledge resources. It is important to include transliteration in the feature set, since many named entities or technical terms are transliterated in full or partially into a foreign language. Thus, the translation feature described in Section 4.3.2 alone is not enough. For this, we collect transliterated titles from the entries connected with language links across the English and the Chinese Wikipedia to calculate correlation between the target transliteration characters and English sublexical strings.

We observed that names of persons and geographic locations are mostly transliterated, and that the entries titled with names of persons or locations can be extracted easily from Wikipedia using the categories of each entry. As will be described in Section 5, we extracted Wikipedia articles tagged with categories that match "*Birth in ...*" to find articles describing a person, and categories that matches "*Cities in ...*" and "*Capitals in ...*" to find titles describing a geographic location. We show some named entities in Table 6.

Table 5. English words segmentation for Chinese-English syllable alignment.

Chinese Transliteration	Chinese Romanization	English Named Entity	Possible Segmentations
喬布斯	qiao-bu-si	jobs	j-o-bs, j-ob-s, **jo-b-s**
瓊喬	qiong-qiao	jonjo	j-onjo, jo-njo, **jon-jo**, jonj-o
喬瑟夫	qiao-se-fu	joseph	j-o-seph, j-os-eph, j-ose-ph, j-osep-h, jo-s-eph, **jo-se-ph**, jo-sep-h, jos-e-ph, ...
喬凡尼	qiao-fan-ni	giovanni	g-i-ovanni, g-io-vanni, g-iov-anni, ..., gio-va-nni, **gio-van-ni**, gio-vann-i, ...

Table 6. Force alignment results of Chinese and English transliteration examples.

Chinese Transliteration	Chinese Romanization	English Syllables
喬布斯	qiao-bu-si	jo-b-s
瓊喬	qiong-qiao	jon-jo
喬瑟夫	qiao-se-fu	jo-se-ph
喬凡尼	qiao-fan-ni	gio-van-ni
拉喬利納	la-qiao-li-na	ra-joe-li-na
奧喬亞	ao-qiao-ya	o-cho-a

After obtaining the transliteration pairs from Wikipedia, we align the Chinese and English syllables. In Chinese, every character always represents one syllable. Nevertheless, the counterpart "syllables" in an English word are not as easy to determine. These counterparts are not syllables in the regular sense, for some counterpart "syllables" may contain a single consonant. We assume every extracted Chinese and English transliteration pairs contain the same number of syllables, *i.e*., equal to the number of Chinese characters. We also assume the syllables are transliterated in order. Under these assumptions, we can segment the English words into a number of segments equal to the number of characters in its Chinese transliteration, and align the English segments and Chinese characters in order. For example, as shown in Table 5, the English name *Joseph* is transliterated into three Chinese characters, or syllables, 喬瑟夫 *qiao-se-fu*, therefore, all possible segmentations include: *j-o-seph*, *j-os-eph*, *j-ose-ph*, *j-osep-h*, *jo-s-eph*, *jo-se-ph*, *jo-sep-h*, *jos-e-ph*, ..., etc.

We use the Expectation-Maximization (EM) algorithm to estimate the conditional probabilities $P(f|e)$ modeling the correlation between the Romanized Chinese characters and the English counterpart. For Chinese characters that have ambiguous pronunciations, we use the Romanization of the most frequent pronunciation according to the Chinese Electronic Dictionary from Academia Sinica, available for download from the The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).[8] In the E-step, the expectation of the log-likelihood of each segmentation candidates are evaluated using the current estimation of $P(f|e)$. In the M-step, the conditional probability estimations are updated based on the maximum likelihood estimation (MLE) of the E-step. A few examples of the segmentation results are shown in Table 6.

[8] http://www.aclclp.org.tw/use_ced.php

Table 7. Conditional probability of Chinese Romanized Chinese character with English syllable. Note that many Chinese characters typically shared the same Romanization.

Rom. Chinese	English Tr.	Cnt(f,e)	P(f|e)
qiao	geo	140	0.38
	jo	66	0.18
	joe	41	0.11
bu	b	1090	0.58
	bu	301	0.16
	br	122	0.07
si	s	5626	0.69
	es	292	0.04
	st	226	0.03

After aligning the syllables in the transliteration pairs, we then calculate the conditional probability of the Romanized Chinese character and its English counterpart. Example output of three Romanized Chinese characters and their top English counterparts is shown in Table 7.

Nevertheless, generating transliteration features for each Chinese character (Romanized) tends to produce a lot of false positives. Therefore, we assume that a named entity is transliterated into at least two Chinese characters, and generate the transliteration features of a Chinese character taking into consideration the preceding and following characters. Admittedly, we probably missed some transliteration cases, such as *Jean* and 琴 (*qin*), but that represents a small loss.

In general, this strategy works quite well for our purpose. For example, given the character sequence 喬布斯(*qiao-bu-si*) and the term *Steve Jobs*, to calculate the transliteration score for the Chinese character 布(*bu*), we calculate the probability of 喬布(*qiao-bu*) and 布斯(*bu-si)* being part of transliteration of *Steve* or *Jobs*:

$$\begin{aligned} P(bu \mid steve) &= max(P(qiao-bu \mid steve), P(bu-si \mid steve)) \\ P(bu \mid jobs) &= max(P(qiao-bu \mid jobs), P(bu-si \mid jobs)) \end{aligned} \tag{3}$$

To calculate the conditional probability for the Chinese bi-characters 喬布 *qiao-bu* given the English term *jobs*, we generate all substring *xy* of *jobs*, into which *qiao-bu* can be transliterated:

$$\begin{aligned} &P(qiao-bu \mid jobs) = \underset{xy \in jobs}{argmax}(P(qiao \mid x) \mid (bu \mid y)) \\ &xy \in jobs \text{ denotes string } xy \text{ is a substring of } jobs \end{aligned} \tag{4}$$

With this probabilistic value, we then generate the transliteration feature values in a similar way as described in Section 4.3.2:

$$feat_{transliteration}(f) = 9 + log(\underset{e \in E}{argmax}\, P(f \mid e)) \quad (5)$$

1. ... 法-fa/0 國-guo/0 立-li/0 體-ti/2 主-zhu/2 義-yi/0 畫-hua/0 家-jia/4 **喬-qiao/7 治-zhi/7 ・/0 布-bu/8 拉-la/8 克-ke/4** (/0 *georges/E braque/E*)/0 ...
2. ... 第-di/0 62/0 屆-jie/0 **艾-ai/3 美-mei/3 獎-jiang/0** 頒-ban/0 獎-jiang/0 典-dian/0 禮-li/0 》/0(/0 the/0 62nd/0 *Emmy/E Award/E*)/0 ...

Figure 5. Example of transliteration features given* Georges Braque *to find the Chinese transliteration "*喬治・布拉克*" and given* Emmy Award *to find "*艾美獎*"

We show two examples of the data tagged with transliteration feature values in Figure 5. In the first example, given the phrase *Georges Braque*, the name of a French painter, to find its Chinese transliteration "喬治・布拉克 (*qiao-zhi bu-la-ke*)". The respective feature scores for each of the characters in the transliteration are *7 7 0 8 8 4*. The symbol "・" with a feature value of zero, is commonly used in Chinese name transliteration to identify the boundary of first and last name in foreign names, and it can be identified as part of the answer by its surrounding transliteration feature scores and the surface pattern. Also in the first example, the Chinese character 家(*jia*), the second syllable of 畫家(*hua-jia*, painter), has a noisy non-zero feature value of *four*, due to the fact that the English syllable *geo* is often transliterated into this Chinese syllable *jia*. In the second example, the given phrase is *Emmy Award*, where the first part of the phrase *Emmy* is transliterated into 艾美(*ai-mei*), and the second part of the phrase *Award* is translated in to 獎(*jiang*). The Chinese characters 艾 and 美 both have a feature value of 3, while all other characters in the example have a feature value of zero. We also show this example tagged with all types of feature values we generate in Table 8.

4.3.4 Generating Distance Features

Finally, we generate the distance features and train a CRF model. The distance feature is intended to exploit the fact that translations tend to occur near the source term, as pointed out in Nagata *et al*. (2001) and Wu *et al*. (2005). Therefore, we incorporated the distance as an additional feature type, to impose a soft constraint on the locational relations between a translation and its English counterpart.

An example showing all three kinds of features and labels is shown in Table 8. This example shows that the given term *Emmy Award* has a Chinese counterpart that is part transliteration (*Emmy* with a transliteration 艾美 *ai-mei*) and part translation (*Award* with the translation 獎 *jiang*). This is a typical case that our method is designed to handle using both

translational and transliterational features. Finally, we use the labeled data with three kinds features to train a CRF model.

Table 8. Example training data.

	word	TR	TL	distance	label
	第	0	0	14	O
	62	0	0	13	O
(62nd)	屆	0	0	12	O
	艾	3	0	11	B
(Emmy)	美	3	0	10	I
(Award)	獎	0	5	9	I
	頒	0	0	8	O
(awarding)	獎	0	0	7	O
	典	0	0	6	O
(ceremony)	禮	0	0	5	O
	》	0	0	4	O
	(	0	0	3	O
	the	0	0	2	O
	62nd	0	0	1	O
	Emmy	0	0	0	E
	Award	0	0	0	E
	)	0	0	-1	O

4.4 Runtime Translation Extraction

Once the CRF model is automatically trained, we attempt to find translations for a given phrase using the procedure in Figure 6.

In Step 1, the system submit the given phrase as query to a search engine (*SE*) to retrieve snippets. Then, for each token in each snippet, we generate three kinds of features (Step 2). This process is exactly the same as in the training phase. In Step 3, we run the CRF model on the snippets to generate labels. Then, in Step 4, we extract the Chinese strings with a sequence of *B*, *I*, ..., *I* tags as translation candidates.

Finally, in Step 5, we compute the frequency of all of the candidates identified in all snippets, and output the candidate with the highest frequency as output. When there is a tie

with multiple candidates with the same highest frequency, one of them is randomly selected as the output.

```
Procedure FindTranslation(P, SE):

(1)Submit P as a query to SE
       to retrieve a set of mixed-code snippets s1, s2, s3, ..., sn

   for each snippet si in snippets s1, s2, s3, ..., sn:
        for each Chinese character in si:
(2)         Generate the three features base on P

(3)Run the CRF model on snippets with features for BIO labels

   for each snippet si in snippets s1, s2, s3, ..., sn:
(4)    Extract Chinese tagged with BI sequence as candidates

(5)Output the candidate with highest redundancy (frequency).
  (In case of a tie, randomly select one of the most frequent.)
```

Figure 6. Pseudocode of the runtime phase.

5. Evaluation

We extracted the titles of English and Chinese articles that are connected through language links in Wikipedia using the Wikipedia dump created on 2010/08/16 (Google, 2010). We used a short list of stop words based on the rules pointed out by Lin *et al*. (2008) to exclude titles that are for administrative or other purposes. We obtained a total of 155,310 article pairs, from which we randomly selected 13,150 and 2,181 titles as seeds to obtain the training and test data, respectively, as described in Section 4.3.1. We then used the English-Chinese Bilingual WordNet[9] and NICT terminology bank (terms.nict.gov.tw/download_main.php) to generate translational features, in an effort to cover both common nouns and technical terms. The bilingual WordNet, translated from the original Princeton WordNet 1.6 has 99,642 synset entries, each with multiple lemmas and multiple translations, forming a total of some 850,000 translation pairs. The NICT database has over 1.1 million term translation pairs in 72 categories and covers a wide variety of different fields. See Table 9 for the numbers of entries in each of the 72 categories.

[9] http://www.aclclp.org.tw/doc/bw_agr_e.PDF

Table 9. Categories of the NICT term database.

Category	Count	Category	Count
Pharmacy	1,673	Material Science (Polymer)	3,422
Bacterial Immunology	2,063	Material Science (Ceramics)	2,292
Phylogenetic	1,756	Agricultural Machinery	3,060
Psychopathology	1,067	Science Education	5,289
Psychology	5,741	Industrial Engineering	5,400
Physics/Chemistry Equipments	17,279	Astronomy	6,091
Comparative Anatomy	6,013	Music	2,922
Education	2,198	Food Science and Technology	35,666
Sociology	2,825	Foreign Names	57,054
Human Anatomy	5,796	Mineralogy	28,032
Pathology	7,307	Lab Animal and Comparative Medicine	8,220
Sports	1,708	Dance	10,564
Soil Science	1,240	Statistic	7,370
Forestry	7,954	Meteorology	20,061
Fertilizer Science	1,155	Animal Husbandry	21,466
Hydraulic Engineering	4,601	Mining and Metallurgical Engineering	13,914
Electronic Engineering	7,627	Computer	101,389
Agricultural Promotion	669	Textile Science and Technology	2,2761
Accounting	4,884	Meteorology	17,789
Civil Engineering	16,745	Endocrinology	2,577
Aeronautics and Astronautics	23,751	Chemical Engineering	22,386
Electrical Engineering	20,058	Communications Engineering	16,899
Engineering Graphics	4,766	Biology (Plants)	42,730
Mathematics	16,708	Mechanism and Machine Theory	2,085
Foundry	5,314	Shipbuilding Engineering	30,701
Mechanical Engineering	35369	Physics	22,077
Earth Science	30673	Zoology	29,586
Geology	22780	Marine	37,329
Marketing	1667	Chemistry (Compound)	19,258
Veterinary Medicine	24,990	Fish	29,730
Nuclear Energy	38,462	Economics	8,891
Production Automation	2,560	Marine Geology	31,015
Surveying	14,371	Power Engineering	69,546
Ecology	7,495	Chemistry (Others)	25,273
Mechanics	10,716	Administration	3,743
Materials Science (Metal)	7,665	Journalism and Communication	4,419

For transliterational features, we extracted person or location entries in Wikipedia using such categories as "*Birth in ...*" to find titles for a person, and categories such as "*Cities in ...*" and "*Capitals in ...*" to find titles for a geographic location. A total of some 15,000 bilingual person names and 24,000 bilingual place names were obtained and forced aligned.

To compare our method with previous work, we used a similar evaluation procedure as described in Lin *et al*. (2008). We ran the system and produced the translations for these 2,181 test data, and we automatically evaluated the results using the metrics of coverage and exact match precision based on the Wikipedia language links. We removed all search snippets from the *wikipedia.org* domain to ensure a strict separation of training and test datasets.

This precision rate is an underestimation since a term may have many alternative translations that do not match exactly with the single reference translation. To obtain a more accurate estimate of the real precision rate, we resorted to manual evaluation.

We selected a small part of the 2,181 English phrases and manually evaluated the results. We report the results of automatic evaluation in Section 5.1 and the results of manual evaluation in Section 5.2.

5.1 Automatic Evaluation

In this section, we describe the evaluation based on the set of 2,181 English-Chinese title pairs extracted from Wikipedia as the gold standard and automatically evaluate coverage (applicability) and exact match precision. Coverage is measured by the percentage of titles for which the proposed system produces some translations.

When translations were extracted, we selected the most frequent translations as output, and checked for exact match against the reference answer. Table 10 shows the results we obtained as compared to the results reported by Lin *et al.* (2008).

We explored the performance differences of the systems employing different set of features. The systems evaluated are as follows:

- **Full**: the proposed system trained with all feature types.
- **-TL** : the proposed system trained without the transliteration feature.
- **-TR** : the proposed system trained without the translation feature.
- **-TL-TR** : the proposed system only using the distance feature. No external knowledge used.
- **LIN En-Ch** : the results reported in the Lin *et al*. paper for their system targeting Chinese parenthetical translations.
- **LIN** En-Ch : the results reported in the Lin *et al*. paper for their system targeting English parenthetical translations

Table 10. Automatic evaluation results.

system	coverage	exact match	top5 exact match
Full (En-Ch)	80.4%	43.0%	56.4%
-TL	83.9%	27.5%	40.2%
-TR	81.2%	37.4%	50.3%
-TL-TR	83.2%	21.1%	32.8%
LIN En-Ch	59.6%	27.9%	not reported
LIN Ch-En	70.8%	36.4%	not reported
LDC (En-Ch)	10.8%	4.8%	N/A
NICT (En-Ch)	24.2%	32.1%	N/A

• **LDC** : the LDC2.0 English to Chinese bilingual dictionary with 161,117 translation pairs. (reported in Lin *et al.*)

• **NICT** : the freely available NICT technical term bilingual dictionary with 1,138,653 translation pairs.

Notice that, although Lin *et al.* (2008) also used bilingual Wikipedia title pairs for evaluation, they used an earlier snapshot of Wikipedia and worked with full webpages crawled from the Internet without a list of given terms. We worked with the list of English terms given as input, but worked only with search engine snippets. In the previous work, all of the bilingual title pairs extracted from Wikipedia were used for evaluation. In our work, only a portion of the title pairs were used for evaluation and the rest were used for generating the training data. It is often difficult to compare systems with different experimental settings. Nevertheless, the evaluation results seem to indicate that the proposed method compares favorably with the results reported in the previous work.

With a given target English term as input, the proposed system uses a search engine to retrieve a relevant portion of limited webpages, and attempts to find the Chinese translation within the retrieved text. The proposed system extracts translations in all cases without being limited by a set of a few surface patterns, and has a significantly higher coverage and precision rate than the previous method that rely on the parenthetic patterns only.

As shown in Table 10, we found using external knowledge to generate features improves system performance significantly. Adding translation feature (-TL) or transliteration feature (-TR) improves exact match precision by about 6% and 16%, respectively. Due to the fact that many Wikipedia titles are fully or partially transliterated into Chinese, the transliteration feature was found to be more important than the translation feature.

The results also clearly show that finding translations on the Web has the advantage of

better coverage than simply looking up phrases in a terminology bank (with a coverage rate of 24%), or a bilingual dictionary (with a coverage rate of 11%). Although using the NICT terminology bank or LDC bilingual dictionary directly has the worst performance, using them as external knowledge sources improves the performance of the CRF model significantly.

Overall, the full system performed the best, finding translations for 8 out of 10 phrases with an average exact match precision rate of over 40%. Nearly 60% of the exact matches appear in the Top 5 candidates. Leaving out the transliteration feature degraded the precision rate by 16%, far more than leaving out the translation feature. This is to be expected, since English Wikipedia has considerably more named entities with transliterated counterparts in Chinese.

5.2 Manual Evaluation

In this section, we present two sets of manual evaluation. In Section 5.2.1, we manually evaluate the results produced by the full system.

5.2.1 Error Analysis on Automatic Evaluation

Since an English phrase is often translated into several Chinese counterparts, evaluation based on exact match against a single reference answer leads to under-estimation. Therefore, we asked a human judge to examine and mark the output of our full system. The judge was instructed to mark each output as **A**: correct translation alternative, **B**: correct translation but with a difference sense from the reference, **P**: partially correct translation, and **E**: incorrect translation.

Table 11 shows 24 randomly selected translations that do not match the relevant reference translations. Half of the translations (12) are correct translations (**A** and **B**), while a third (8) are partially correct translation (**P**). Notice that it is a common practice to translate only the surname of a foreign person. So, four of the eight partial translations may be considered as correct.

In Table 12, we show extracted candidates and frequency counts for 8 example terms. Translation candidates are marked using the same *A*, *B*, *P*, and *E* tags as in Table 11, plus an additional tag, *M*, to indicate an exact match. For the given term *money laundering*, the system extracted 27 exact matches (洗錢), and 2 correct alternatives (洗黑錢) and only 1 erroneous output from 30 snippets returned from the search engine. While technical terms like *money laundering* tend to have literal translations and result in more exact matches, movie titles are often translated into Chinese with completely different meanings. For example, the official Chinese title for the movie, *Music and Lyrics* in Taiwan is “K-歌-情人” (meaning *karaoke-song-lover*). Given such a title as input, the system was able to extract 18 partial

matches and 2 exact matches base on surface patterns and modest translation feature value for *music* and 歌(*ge*, *song*). For the given term *colony*, the system extracted 菌落(*colony of fungi or bacteria*), a correct translation with a different sense. Other extracted answers include: transliteration, 科羅尼海島酒店(*Island Colony*), the name of a hotel, and the exact-match translation, 殖民地(*foreign control territory*). For the given term *bubble sort*, the partial translation 排序(*sort*) makes the top-1 translation (with a count of 20), while the top-2 to top-5 are either exact-match or acceptable translations.

Table 11. Cases failing the exact match test.

English Wiki	Chinese Wiki	Extracted	
Pope Celestine IV	塞萊斯廷四世	切萊斯廷四世	A
Huaneng Power International	華能國際	華能國際電力	A
Shangrao	上饒市	上饒	A
Aurora University	震旦大學	奧羅拉大學	A
Fujian	福建省	福建	A
Dream Theater	夢劇場	夢劇場合唱團	A
Coturnix	鶉屬	鵪鶉	A
Waste	垃圾	廢物	A
Allyl alcohol	烯丙醇	丙烯醇	A
Machine	機械	工具機	A
Colony	殖民地	菌落	B
Collateral	落日殺神	抵押	B
Ludwig Erhard	路德維希・艾哈德	艾哈德	P
John Woo	吳宇森	約翰	P
Osman I	奧斯曼一世	奧斯曼	P
Itumeleng Khune	伊圖梅倫・庫內	庫內	P
Naphthoquinone			P
Base analog	鹼基類似物	鹼基類	P
Chinese Paladin	仙劍奇俠傳	神劍	P
Bubble sort	冒泡排序	排序	P
The Love Suicides at Sonezaki	曾根崎情死	夏目漱石	E
Survivor's Law II	律政新人王II	金石良緣	E
Phichit	批集府	朗家庭主婦	E
Ammonium	銨	過硫酸銨	E

Note that this learning-based approach to mining translation and transliteration on the Web is an original contribution of our work. Previous works such as Wu *et al*. (2005); Lin *et al*. (2008), simply used occurrence statistics to identify translations, which is roughly equivalent to our translational or transliterational features (see Section 4.3.2 and Section 4.3.3). While Lin *et al*. used prefixes of 3 letters to provide a makeshift model of transliteration, we model the name-transliteration relations directly using an EM algorithm. Moreover, we also take note of their pattern of appearance to allow more effective extraction of relevant translations with the distance feature (see Section 4.3.4). It is important to note that combining features inherent in a training data, as well as derived from external knowledge sources in a machine learning model allow us to cover more relevant translations, while filtering out many invalid candidates.

Table 12. Extracted candidates and frequencies.

given term	freq	candidate	
money laundering	27	洗錢	**M**
	2	洗黑錢	A
	1	洗錢宣傳	E
Music and Lyrics	18	歌情人	P
	2	K歌情人	**M**
flyback transformer	14	變壓器	P
	3	回掃變壓器	**M**
	2	返馳式變壓器	A
	2	返馳變壓器	A
colony	15	菌落	B
	2	科羅尼海島酒店	B
	2	殖民地	**M**
Osman I	8	奧斯曼	P
	5	奧斯曼一世	**M**
bubble sort	20	排序	P
	19	泡排序	A
	17	氣泡排序	**M**
	9	泡沫排序	A
	4	泡泡排序	A

6. Conclusion and Future Work

We have presented a new method for mining translations on the Web for a given term. In our work, we use a set of terms and translations as seeds to obtain mixed-code snippets returned by a search engine, such as *Google* or *Bing*. We then automatically convert the snippets into a tagged sequence of tokens, automatically augment the data with features obtained from external knowledge sources, and automatically train a CRF model for sequence labels. At runtime, we submit a query consisting of the given term to a search engine, tag the returned snippets using the trained model, and finally extract and rank the translation candidates for output. Preliminary experiments and evaluations show our method cleanly combining various features, resulting in an integrated, learning-based system capable of finding both term translations and transliterations.

Many avenues exist for future research and improvement of our system. For example, existing query expansion methods to retrieve more webpages containing translation for the given phrases could be implemented (Zhang *et al*., 2005). Translation features related to word parts (*e.g*., *-lite* in the term *zeolite*) could be used to improve identification of translations. Additionally, an interesting direction to explore is to identify phrase types and length (*i.e*., base NP and NP prep. NP) and train type-specific CRF models for better results. In addition, natural language processing techniques such as word stemming, word lemmatization, or derivational morphological transformation could also be attempted to improve recall and precision.

Another interesting direction to explore is using a robot to crawl webpages and filter mixed-code data to derive the translation features. With the crawled web pages, we can extract translations offline, without having to work with a search engine and its limited returned snippets.

Yet another direction of research would be to enhance the effectiveness of translation features by working on the level of Chinese words instead of characters. For that, we could either use an existing, general-purpose word segmenter or carry out self-organized word segmentation (Sproat & Shih, 1990) to produce word-based translation features.

Reference

Bian, G.-W., & Chen, H.-H. (2000). Cross-language information access to multilingual collections on the internet. *Journal of the American Society for Information Science*, 51(3), 281-296.

Cao, Y., & Li, H. (2002). Base noun phrase translation using web data and the em algorithm. In *Proceedings of the 19th international conference on computational linguistics*, volume 1, 1-7.

Chang, J. Z., Chang, J. S., & Jang, R. J.-S. (2012). Learning to find translations and transliterations on the web. In *Proceedings of the 50th annual meeting of the association for computational linguistics*, volume 2, 130-134.

Cheng, P.-J., Teng, J.-W., Chen, R.-C., Wang, J.-H., Lu, W.-H., & Chien, L.-F. (2004). Translating unknown queries with web corpora for cross-language information retrieval. In *Proceedings of the 27th annual international acm sigir conference on research and development in information retrieval*, 146-153.

Fellbaum, C. (1998). *Wordnet: An electronic lexical database*. MIT Press.

Gale, W. A., & Church, K. W. (1991). Identifying word correspondence in parallel texts. In *Proceedings of the workshop on speech and natural language*, 152-157.

Giles, J. (2005). Internet encyclopaedias go head to head. *Nature*, 438(7070), 900-901.

Google. (2010). Freebase data dumps (August 16th, 2010 ed.). http://download.freebase.com/datadumps/.

Gravano, L., & Henzinger, M. H. (2006). Systems and methods for using anchor text as parallel corpora for cross-language information retrieval (No. 7146358).

Huang, C.-R. (2003). Sinica bow: integrating bilingual wordnet and sumo ontology. In *Proceedings of 2003 International Conference on Natural Language Processing and Knowledge Engineering*, 825-826.

Huang, F., Vogel, S., & Waibel, A. (2003). Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *Proceedings of the acl 2003 workshop on multilingual and mixed-language named entity recognition*, 15, 9-16.

Knight, K., & Graehl, J. (1998). Machine transliteration. *Computational Linguistics*, 24(4), 599-612.

Koehn, P., & Knight, K. (2003). Feature-rich statistical translation of noun phrases. In *Proceedings of the 41st annual meeting on association for computational linguistics*, volume 1, 311-318.

Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st annual meeting on association for computational linguistics*, 17-22.

Kwok, K., Deng, P., Dinstl, N., Sun, H., Xu, W., Peng, P., & Doyon., J. (2005). Chinet: a chinese name finder system for document triage. In *Proceedings of 2005 international conference on intelligence analysis*.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning*, 282-289.

Li, Y., & Grefenstette, G. (2005). Translating chinese romanized name into Chinese idiographic characters via corpus and web validation. In *Proceedings of coria 2005*, 323-338.

Lin, D., Zhao, S., Van Durme, B., & Paşca, M. (2008). Mining parenthetical translations from the web by word alignment. In *Proceedings of acl-08: Hlt*, 994-1002.

Lu, W.-H., Chien, L.-F., & Lee, H.-J. (2004). Anchor text mining for translation of web queries: A transitive translation approach. *ACM Trans. Inf. Syst.*, 22(2), 242-269.

Melamed, I. D. (2000). Models of translational equivalence among words. *Computational Linguistics*, 26(2), 221-249.

Nagata, M., Saito, T., & Suzuki, K. (2001). Using the web as a bilingual dictionary. In *Proceedings of the workshop on data-driven methods in machine translation*, volume 14, 1-8.

Qu, Y., & Grefenstette, G. (2004). Finding ideographic representations of Japanese names written in latin script via language identification and corpus validation. In *Proceedings of the 42nd annual meeting on association for computational linguistics*.

Quah, C. K. (2006). *Translation and technology*. Palgrave Macmillan.

Smadja, F., McKeown, K. R., & Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: a statistical approach. *Computational Linguistics*, 22(1), 1-38.

Sproat, R. W., & Shih, C. (1990). A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4), 336-351.

Wu, J.-C., Lin, T., & Chang, J. S. (2005). Learning source-target surface patterns for web-based terminology translation. In *Proceedings of the acl 2005 on interactive poster and demonstration sessions*, 37-40.

Zhang, Y., Huang, F., & Vogel, S. (2005). Mining translations of oov terms from the web through cross-lingual query expansion. In *Proceedings of the 28th annual international acm sigir conference on research and development in information retrieval*, 669-670.