Learning to Find Translations and Transliterations on the Web

Joseph Z. Chang
Department of Computer Science,
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
joseph.nthu.tw@gmail.com

Jason S. Chang
Department of Computer Science,
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
jschang@cs.nthu.edu.tw

Jyh-Shing Roger Jang
Department of Computer Science,
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
jang@cs.nthu.edu.tw

Abstract

In this paper, we present a new method for learning to finding translations and transliterations on the Web for a given term. The approach involves using a small set of terms and translations to obtain mixed-code snippets from a search engine, and automatically annotating the snippets with tags and features for training a conditional random field model. At run-time, the model is used to extracting translation candidates for a given term. Preliminary experiments and evaluation show our method cleanly combining various features, resulting in a system that outperforms previous work.

1 Introduction

The phrase translation problem is critical to machine translation, cross-lingual information retrieval, and multilingual terminology (Bian and Chen 2000, Kupiec 1993). Such systems typically use a parallel corpus. However, the out of vocabulary problem (OOV) is hard to overcome even with a very large training corpus due to the Zipf nature of word distribution, and ever growing new terminology and named entities. Luckily, there are an abundant of webpages consisting mixed-code text, typically written in one language but interspersed with some sentential or phrasal translations in another language. By retrieving and identifying such translation counterparts on the Web, we can cope with the OOV problem.

Consider the technical term *named-entity recognition*. The best places to find the Chinese translations for named-entity recognition are probably not some parallel corpus or dictionary, but rather mixed-code webpages. The following example is a snippet returned by the Bing search engine for the query, *named entity recognition*:

> ... 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 ...

This snippet contains three technical terms in Chinese (i.e., 自然語言剖析 *zhiran yuyan poxi*, 問題分類 *wenti fenlei*, 專名辨識 *zhuanming bianshi*), followed by source terms in brackets (respectively, *Natural Language Parsing*, *Question Classification*, and *Named Entity Recognition*). Quoh (2006) points out that submitting the source term and partial translation to a search engine is a good strategy used by many translators.

Unfortunately, the user still has to sift through snippets to find the translations. For a given English term, such translations can be extracted by casting the problem as a sequence labeling task for classifying the Chinese characters in the snippets as either translation or non-translation. Previous work has pointed out that such translations usually exhibit characteristics related to word translation, word transliteration, surface patterns, and proximity to the occurrences of the original phrase (Nagata et. al 2001 and Wu et. al 2005).

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 130–134,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

Thus, we also associate features to each Chinese token (characters or words) to reflect the likelihood of the token being part of the translation. We describe how to train a CRF model for identifying translations in more details in Section 3.

At run-time, the system accepts a given phrase (e.g., *named-entity recognition*), and then query a search engine for webpages in the target language (e.g., Chinese) using the advance search function. Subsequently, we retrieve mixed-code snippets and identify the translations of the given term. The system can potentially be used to assist translators to find the most common translation for a given term, or 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 et al. (2008).

2 Related Work

Phrase translation and transliteration is important for cross-language tasks. For example, Knight and Graehl (1998) describe and evaluate a multi-stage machine translation method for back transliterating English names into Japanese, while Bian and Chen (2000) describe cross-language information access to multilingual collections on the Internet.

Recently, researchers have begun to exploit mixed code webpages for word and phrase translation. Nagata et al. (2001) present a system for finding English translations for a given Japanese technical term using Japanese-English snippets returned by a search engine. Kwok et al. (2005) focus on named entity transliteration and implemented a cross-language name finder. Wu et al. (2005) proposed a method to learn surface patterns to find translations in mixed code snippets.

Some researchers exploited the hyperlinks in Webpage to find translations. Lu, et al. (2004) propose a method for mining translations of web queries from anchor texts. Cheng, et al (2004) propose a similar method for translating unknown queries with web corpora for cross-language information retrieval. Gravano (2006) also propose similar methods using anchor texts.

In a study more closely related to our work, Lin et al. (2008) proposed a method that performs word alignment between translations and phrases within parentheses in crawled webpages. They use heuristics to align words and translations, while we use a learning based approach to find translations.

	Token	TR	TL	Distance	Label
	第	0	0	14	O
	62	0	0	13	O
62th	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
	62th	0	0	1	O
	Emmy	0	0	0	E
	Award	0	0	0	E
	)	0	0	-1	O

Figure 1. Example training data.

In contrast to previous work described above, we exploit surface patterns differently as a soft constraint, while requiring minimal human intervention to prepare the training data.

3 Method

To find translations for a given term on the Web, a promising approach is automatically learning to extract phrasal translations or transliterations of phrase based on machine learning, or more specifically the conditional random fields (CRF) model.

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.

3.1 Preparing Data for CRF Classifier

We make use a small set of term and translation pairs as seed data to retrieve and annotate mixed-code snippets from a search engine. Features are generated based on other external knowledge sources as will be described in Section 3.1.2 and 3.1.3. An example data generated with given term *Emmy Award* with features and translation/non-translation labels is shown in Figure 1 using the common **BIO** notation.

3.1.1 *Retrieving and tagging snippets*. We use a list of randomly selected source and target terms as seed data (e.g., Wikipedia English titles and their

Chinese counterpart using the *language links*). We use the English terms (e.g., *Emmy Awards*) to query a search engine with the target webpage language set to the target language (e.g., Chinese), biasing the search engine to return Chinese webpages interspersed with some English phrases. We then automatically label each Chinese character of the returned snippets, with **B**, **I**, **O** indicating respectively ***beginning***, ***inside***, and ***outside*** of translations. In Figure 1, the translation 艾美獎 (*ai mei jiang*) are labeled as **B I I**, while all other Chinese characters are labeled as **O**. An additional tag of ***E*** is used to indicate the occurrences of the given term (e.g., *Emmy Awards* in Figure 1).

3.1.2 *Generating translation feature*. We generate translation features using external bilingual resources. The φ^2 score proposed by Gale and Church (1991) is used to measure the correlations between English and Chinese tokens:

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

where e is an English word and f is a Chinese character. The scores are calculated by counting co-occurrence of Chinese characters and English words in bilingual dictionaries or termbanks, where $P(e, f)$ represents the probability of the co-occurrence of English word e and Chinese character f, and $P(e,\bar{f})$ represents the probability the co-occurrence of e and any Chinese characters excluding f.

We used the publicly available English-Chinese Bilingual WordNet and NICT terminology bank to generate translation features in our implementation. The bilingual WordNet has 99,642 synset entries, with a total of some 270,000 translation pairs, mainly common nouns. The NICT database has over 1.1 million bilingual terms in 72 categories, covering a wide variety of different fields.

3.1.3 *Generating transliteration feature*. Since many terms are transliterated, it is important to include transliteration feature. We first use a list of name transliterated pairs, then use Expectation-Maximization (EM) algorithm to align English syllables Romanized Chinese characters. Finally, we use the alignment information to generate transliteration feature for a Chinese token with respect to English words in the query.

We extract person or location entries in Wikipedia as name transliterated pairs to generate transliteration features in our implementation. This can be achieved by examining the Wikipedia categories for each entry. A total of some 15,000 bilingual names of persons and 24,000 bilingual place names were obtained and forced aligned to obtain transliteration relationships.

3.1.4 *Generating distance feature*. In the final stage of preparing training data, we add the distance, i.e. number of words, between a Chinese token feature and the English term in question, aimed at exploiting the fact that translations tend to occur near the source term, as noted in Nagata et al. (2001) and Wu et al. (2005).

Finally, we use the data labeled with translation tags and three kinds feature values to train a CRF model.

3.2 Run-Time Translation Extraction

With the trained CRF model, we then attempt to find translations for a given phrase. The system begins by submitting the given phrase as query to a search engine to retrieve snippets, and generate features for each tokens in the same way as done in the training phase. We then use the trained model to tag the snippets, and extract translation candidates by identifying consecutive Chinese tokens labeled as **B** and **I**.

Finally, we compute the frequency of all the candidates identified in all snippets, and output the one with the highest frequency.

4 Experiments and Evaluation

We extracted the Wikipedia titles of English and Chinese articles connected through language links for training and testing. We obtained a total of 155,310 article pairs, from which we then randomly selected 13,150 and 2,181 titles as seeds to obtain the training and test data. Since we are using Wikipedia bilingual titles as the gold standard, we exclude any snippets from the *wikipedia.org* domain, so that we are not using Wikipedia article content in both training and testing stage. The test set contains 745,734 snippets or 9,158,141 tokens (Chinese character or English word). The reference answer appeared a total of 48,938 times or 180,932 tokens (2%), and an average of 22.4 redundant answer instances per input.

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
LCD (En-Ch)	10.8%	4.8%	N/A
NICT (En-Ch)	24.2%	32.1%	N/A

Table 1. Automatic evaluation results of 8 experiments: (1) Full system (2-4) -TL, -TR, -TL-TR : Full system deprecating TL, TR, and TL+TL features (5,6) LIN En-Ch and En-Ch : the results in Lin et al. (2008) (6) LDC: LDC E-C dictionary (7) NICT : NICT term bank.

English Wiki	Chinese Wiki	Extracted	Ev.
Pope Celestine IV	塞萊斯廷四世	切萊斯廷四世	A
Fujian	福建省	福建	A
Waste	垃圾	廢物	A
Collateral	落日殺神	抵押	B
Ludwig Erhard	路德維希·艾哈德	艾哈德	P
Osman I	奥斯曼一世	奥斯曼	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
Ammonium	銨	過硫酸銨	E

Table 2. Cases failing the exact match test.

Result	Count	Percentage
A+B: correct	53	55.8%
P: partially corr.	30	31.6%
E: incorrect	8	8.4%
N: no results	4	4.2%
total	95	100%

Table 3. Manual evaluation of unlink titles.

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 automatically evaluate the results using the metrics of ***coverage***, i.e. when system was able to produce translation candidates, and ***exact match precision***.

This precision rate is an under-estimations, since a term may have many alternative translations that does not match exactly with one single reference translation. To give a more accurate estimate of real precision, we resorted to manual evaluation on a small part of the 2,181 English phrases and a small set of English Wikipedia titles without a Chinese language link.

4.1 Automatic Evaluation

In this section, we describe the evaluation based on English-Chinese titles extracted from Wikipedia as the gold standard. Our system produce the top-1 translations by ranking candidates by frequency and output the most frequent translations. Table 1 shows the results we have obtained as compared to the results of Lin et al. (2008).

Table 1 shows the evaluation results of 8 experiments. The results indicate that using external knowledge to generate feature improves system performance significantly. By adding translation feature (TL) or transliteration feature (TR) to the system with no external knowledge features (-TL-TR) improves exact match precision by about 6% and 16% respectively. Because many Wikipedia titles are named entities, transliteration feature is the most important. Overall, the system with full features perform the best, finding reasonably correct translations for 8 out of 10 phrases.

4.2 Manual Evaluation

Evaluation based on exact match against a single reference answer leads to under-estimation, because an English phrase is often translated into several Chinese counterparts. Therefore, we asked a human judge to examine and mark the outputs 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 2 shows some translations generated by the full system that does not match the single reference translation. Half of the translations are correct translations (**A** and **B**), while a third are partially correct translation (**P**). Notice that it is a common practice to translate only the surname of a foreign person. Therefore, some partial translations may still be considered as correct (**B**).

To Evaluate titles without a language link, we sampled a list of 95 terms from the unlinked portion of Wikipedia using the criteria: (1) with a frequency count of over 2,000 in Google Web 1T. (2) containing at least three English words. (3) not a proper name. Table 3 shows the evaluation results. Interestingly, our system provides correct translations for over 50% of the cases, and at least partially correct almost 90% of the cases.

5 Conclusion and Future work

We have presented a new method for finding translations on the Web for a given term. In our approach, we use a small set of terms and translations as seeds to obtain and to tag mixed-code snippets returned by a search engine, in order to train a CRF model for sequence labels. This CRF model is then used to tag the returned snippets for a given query term to extraction translation candidates, which are then ranked and returned as output. Preliminary experiments and evaluations show our learning-based method cleanly combining various features, producing quality translations and transliterations.

Many avenues exist for future research and improvement. For example, existing query expansion methods could be implemented to retrieve more webpages containing translations. Additionally, an interesting direction to explore is to identify phrase types and train type-specific CRF model. In addition, natural language processing techniques such as word stemming and word lemmatization could be attempted.

References

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

Y. Cao and H. Li. Base Noun Phrase Translation Using Web Data and the EM Algorithm. 2002. In *Proceedings* of the 19th International Conference on Computational Linguistics (COLING'02), pp.127-133, 2002.

P. J. Cheng, J. W. Teng, R. C. Chen, J. H. Wang, W. H. Lu, and L. F. Chien. Translating unknown queries with web corpora for cross-language information retrieval. In Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval, pp.146-153, 2004.

F. Huang, S. Vogel, and A. Waibel. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *Proceeding of* the 41st ACL, Workshop on Multilingual and Mixed-Language Named Entity Recognition, Sapporo, 2003.

K. Knight, J. Graehl. Machine Transliteration. 1998. Computational Linguistics 24(4), pp.599-612, 1998.

P. Koehn, K. Knight. 2003. Feature-Rich Statistical Translation of Noun Phrases. In *Proceedings of the* 41st Annual Meeting on Association for Computational *Linguistics*, pp. 311-318, 2003.

J. Kupiec. 1993. An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In *Proceedings* of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 17-22, 1993.

KL Kwok, P Deng, N Dinstl, HL Sun, W Xu, P Peng, and Doyon, J. 2005. CHINET: a Chinese name finder system for document triage. In *Proceedings of 2005*

D. Lin, S. Zhao, B.V. Durme, and M. Paşca. 2008. Mining Parenthetical Translation from the Web by Word Alignment, In Proceedings of ACL 2008, pp. 994-1002, 2008.

Y. Li, G. Grefenstette. 2005. Translating Chinese Romanized name into Chinese idiographic characters via corpus and web validation. In *Proceedings of CORIA 2005*, pp. 323-338, 2005.

M. Nagata, T. Saito, and K. Suzuki. Using the Web as a bilingual dictionary. 2001. In *Proceedings of 39th.* ACL Workshop on Data-Driven Methods in Machine *Translation*, pp. 95-102, 2001.

Y. Qu, and G. Grefenstette. 2004. Finding Ideographic Representations of Japanese Names Written in Latin Script via Language Identification and Corpus Validation. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, pp.183-190, 2004.

CK Quah. 2006. Translation and Technolog*y*, *Palgrave* Textbooks in Translation and Interpretation, Palgrave MacMillan.

R Sproat and C Shih. Statistical Method for Finding Word Boundaries in Chinese Text, Computer Processing of Chinese and Oriental languages. 1990.

J. C. Wu, T. Lin and J. S. Chang. Learning Source-Target Surface Patterns for Web-based Terminology Translation. In Proceeding of the ACL 2005 on Interactive poster and demonstration sessions (ACLdemo '05). 2005.

Y Zhang, F Huang, S Vogel. 2005. Mining translations of OOV terms from the web through cross-lingual query expansion. In Proceedings of the 28th Annual International ACM SIGIR, pp.669-670, 2005.