Semi-Supervised Lexicon Mining from Parenthetical Expressions in Monolingual Web Pages

Xianchao Wu[†] **Naoaki Okazaki**[†] **Jun'ichi Tsujii**[†‡]

[†]Computer Science, Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[‡]School of Computer Science, University of Manchester
National Centre for Text Mining (NaCTeM)
Manchester Interdisciplinary Biocentre, 131 Princess Street, Manchester M1 7DN, UK

{wxc, okazaki, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper presents a semi-supervised learning framework for mining Chinese-English lexicons from large amount of Chinese Web pages. The issue is motivated by the observation that many Chinese neologisms are accompanied by their English translations in the form of parenthesis. We classify parenthetical translations into bilingual abbreviations, transliterations, and translations. A frequency-based term recognition approach is applied for extracting bilingual abbreviations. A self-training algorithm is proposed for mining transliteration and translation lexicons. In which, we employ available lexicons in terms of morpheme levels, i.e., phoneme correspondences in transliteration and grapheme (e.g., suffix, stem, and prefix) correspondences in translation. The experimental results verified the effectiveness of our approaches.

1 Introduction

Bilingual lexicons, as lexical or phrasal parallel corpora, are widely used in applications of multilingual language processing, such as statistical machine translation (SMT) and cross-lingual information retrieval. However, it is a time-consuming task for constructing large-scale bilingual lexicons by hand. There are many facts cumber the manual development of bilingual lexicons, such as the continuous emergence of neologisms (e.g., new technical terms, personal names, abbreviations, etc.), the difficulty of keeping up with the neologisms for lexicographers, etc. In order to turn the facts to a better way, one of the simplest strategies is to automatically mine large-scale lexicons from corpora such as the daily updated Web.

Generally, there are two kinds of corpora used for automatic lexicon mining. One is the purely monolingual corpora, wherein frequency-based expectation-maximization (EM, refer to (Dempster et al., 1977)) algorithms and cognate clues play a central role (Koehn and Knight, 2002). Haghighi et al. (2008) presented a generative model based on canonical correlation analysis, in which monolingual features such as the context and orthographic substrings of words were taken into account. The other is multilingual parallel and comparable corpora (e.g., Wikipedia[1]), wherein features such as co-occurrence frequency and context are popularly employed (Cheng et al., 2004; Shao and Ng, 2004; Cao et al., 2007; Lin et al., 2008).

In this paper, we focus on a special type of comparable corpus, *parenthetical translations*. The issue is motivated by the observation that Web pages and technical papers written in Asian languages (e.g., Chinese, Japanese) sometimes annotate named entities or technical terms with their translations in English inside a pair of parentheses. This is considered to be a traditional way to annotate new terms, personal names or other named entities with their English translations expressed in brackets. Formally, a parenthetical translation can be expressed by the following pattern,

$$f_1\ f_2\ ...\ f_J\ (e_1\ e_2\ ...\ e_I). \quad (1)$$

Here, $f_1\ f_2\ ...\ f_J (f_1^J)$, the pre-parenthesis text, denotes the word sequence of some language other than English; and $e_1\ e_2\ ...\ e_I (e_1^I)$, the in-parenthesis text, denotes the word sequence of English. We separate parenthetical translations into three categories:

[1]http://en.wikipedia.org/wiki/Main_Page

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 424–432,
Boulder, Colorado, June 2009.

Type	Examples with translations in *italic*
Abbreviation	对‖全球 气候 观测 系统 (GCOS) *to Global Climate Observing System (GCOS)*
Transliteration	品牌 将 在‖辛普顿- 特尔曼(Shipton-Tilman) *brand will be among Shipton-Tilman (Shipton-Tilman)*
Translation	定时炸弹，‖删除蝇(Cancelbots) *time bomb, Cancelbots (Cancelbots)*
Mixture	在 香港 上课 的 英国‖布拉福特 大学 (Bradford University) *the English Bradford University (Bradford University) that holds lessons in Hongkong*

Table 1: Parenthetical translation categories and examples extracted from Chinese Web pages. Mixture stands for the mixture of translation (*University*) and transliteration (*Bradford*). '‖' denotes the left boundary of f_1^J.

bilingual abbreviation, transliteration, and translation. Table 1 illustrates examples of these categories.

We address several characteristics of parenthetical translations that differ from traditional comparable corpora. The first is that they only appear in monolingual Web pages or documents, and the context information of e_1^I is unknown. Second, frequency and word number of e_1^I are frequently small. This is because parenthetical translations are only used when the authors thought that f_1^J contained some neologism(s) which deserved further explanation in another popular language (e.g., English). Thus, traditional context based approaches are not applicable and frequency based approaches may yield low recall while with high precision. Furthermore, cognate clues such as orthographic features are not applicable between language pairs such as English and Chinese.

Parenthetical translation mining faces the following issues. First, we need to distinguish parenthetical translations from parenthetical expressions, since parenthesis has many functions (e.g., defining abbreviations, elaborations, ellipsis, citations, annotations, etc.) other than translation. Second, the left boundary (denoted as ‖ in Table 1) of the pre-parenthesis text need to be determined to get rid of the unrelated words. Third, we need further distinguish different translation types, such as bilingual abbreviation, the mixture of translation and transliteration, as shown in Table 1.

In order to deal with these problems, supervised (Cao et al., 2007) and unsupervised (Li et al., 2008) methods have been proposed. However, supervised approaches are restricted by the quality and quantity of manually constructed training data, and unsupervised approaches are totally frequency-based without using any semantic clues. In contrast, we propose a semi-supervised framework for mining parenthetical translations. We apply a monolingual abbreviation extraction approach to bilingual abbreviation extraction. We construct an English-syllable to Chinese-pinyin transliteration model which is self-trained using phonemic similarity measurements. We further employ our cascaded translation model (Wu et al., 2008) which is self-trained based on morpheme-level translation similarity.

This paper is organized as follows. We briefly review the related work in the next section. Our system framework and self-training algorithm is described in Section 3. Bilingual abbreviation extraction, self-trained transliteration models and cascaded translation models are described in Section 4, 5, and 6, respectively. In Section 7, we evaluate our mined lexicons by Wikipedia. We conclude in Section 8 finally.

2 Related Work

Numerous researchers have proposed a variety of automatic approaches to mine lexicons from the Web pages or other large-scale corpora. Shao and Ng (2004) presented a method to mine new translations from Chinese and English news documents of the same period from different news agencies, combining both transliteration and context information. Kuo et al. (2006) used active learning and unsupervised learning for mining transliteration lexicon from the Web pages, in which an EM process was used for estimating the phonetic similarities between English syllables and Chinese characters.

Cao et al. (2007) split parenthetical translation mining task into two parts, transliteration detection and translation detection. They employed a transliteration lexicon for constructing a grapheme-based transliteration model and annotated boundaries manually to train a classifier. Lin et al. (2008) applied a frequency-based word alignment approach, Competitive Link (Melanmed, 2000), to determine the outer boundary (Section 7).

On the other hand, there have been many semi-supervised approaches in numerous applications

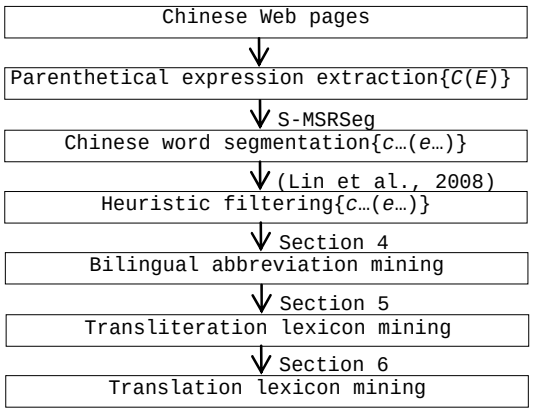

Figure 1: The system framework of mining lexicons from Chinese Web pages.

(Zhu, 2007), such as self-training in word sense disambiguation (Yarowsky, 2005) and parsing (McClosky et al., 2008). In this paper, we apply self-training to a new topic, lexicon mining.

3 System Framework and Self-Training Algorithm

Figure 1 illustrates our system framework for mining lexicons from Chinese Web pages. First, parenthetical expressions matching Pattern 1 are extracted. Then, pre-parenthetical Chinese sequences are segmented into word sequences by S-MSRSeg[2] (Gao et al., 2006). The initial parenthetical translation corpus is constructed by applying the heuristic rules defined in (Lin et al., 2008)[3]. Based on this corpus, we mine three lexicons step by step, a bilingual abbreviation lexicon, a transliteration lexicon, and a translation lexicon. The abbreviation candidates are extracted firstly by using a heuristic rule (Section 4.1). Then, the transliteration candidates are selected by employing a transliteration model (Section 5.1). Specially, $f_1^J(e_1^I)$ is taken as a transliteration candidate only if a word e_i in e_1^I can be transliterated. In addition, a transliteration candidate will also be considered as a translation candidate if not all e_i can be transliterated (refer to the mixture example in Table1). Finally, after abbreviation filtering and transliteration filtering, the remaining candidates are used for translation lexicon mining.

[2]http://research.microsoft.com/research/downloads/details/7a2bb7ee-35e6-40d7-a3f1-0b743a56b424/details.aspx

[3]e.g., f_1^J is predominantly in Chinese and e_1^I is predominantly in English

Algorithm 1 self-training algorithm

Require: L, $U = \{f_1^J(e_1^I)\}$, T, $\mathcal{M}$ $\quad \triangleright L$, (labeled) training set; U, (unlabeled) candidate set; T, test set; $\mathcal{M}$, the transliteration or translation model.

1: $Lexicon = \{\}$ $\quad \triangleright$ new mined lexicon
2: **repeat**
3: $\quad N = \{\} \triangleright$ new mined lexicon during one iteration
4: $\quad$ train $\mathcal{M}$ on L
5: $\quad$ evaluate $\mathcal{M}$ on T
6: $\quad$ **for** $f_1^J(e_1^I) \in U$ **do**
7: $\quad\quad topN = \{C' | \text{decode } e_1^I \text{ by } \mathcal{M}\}$
8: $\quad\quad N = N \cup \{(c, e_1^I) | c \in f_1^J \wedge \exists C' \in topN \; s.t. \; similarity\{c, C'\} \geq \theta\}$
9: $\quad$ **end for**
10: $\quad U = U - N$
11: $\quad L = unified(L \cup N)$
12: $\quad Lexicon = unified(Lexicon \cup N)$
13: **until** $|N| \leq \epsilon$
14: **return** $Lexicon$ $\quad \triangleright$ the output

Algorithm 1 addresses the self-training algorithm for lexicon mining. The main part is a loop from Line 2 to Line 13. A given seed lexicon is taken as labeled data and is split into training and testing sets (L and T). U=$\{f_1^J(e_1^I)\}$, stands for the (unlabeled) parenthetical expression set. Initially, a translation/transliteration model ($\mathcal{M}$) is trained on L and evaluated on T (Line 4 and 5). Then, the English phrase e_1^I of each unlabeled entry is decoded by $\mathcal{M}$, and the top-N outputs are stored in set $topN$ (Line 7$\sim$8). A similarity function on c (a word substring of f_1^J) and a top-N output C' is employed to make the decision of classification: the pair (c, e_1^I) will be selected as a new entry if the similarity between c and C' is no smaller than a threshold value θ (Line 8). After processing each entry in U, the new mined lexicon N is deleted from U and unified with the current training set L as the new training set (Line 10 and 11). Also, N is added to the final lexicon (Line 12). When $|N|$ is lower than a threshold, the loop stops. Finally, the algorithm returns the mined lexicon.

One of the open problems in Algorithm 1 is how to append new mined entries into the existing seed lexicon, considering they have different distributions. One way is to design and estimate a weight function on the frequency of new mined entries. For simplicity, we use a deficient strategy that takes the weights of all new mined entries to be one.

4 Bilingual Abbreviation Extraction

4.1 Methodology

The method that we use for extracting a bilingual abbreviation lexicon from parenthetical expressions is inspired by (Okzaki and Ananiadou, 2006). They used a term recognition approach to build a monolingual abbreviation dictionary from the Medical Literature Analysis and Retrieval System Online (MEDLINE) abstracts, wherein acronym definitions (e.g., *ADM* is short for *adriamycin*, *adrenomedullin*, etc.) are abundant. They reported 99% precision and 82-95% recall. Through locating a textual fragment with an acronym and its expanded form in pattern

$$\text{long form (short form)}, \tag{2}$$

they defined a heuristic formula to compute the long-form likelihood LH(c) for a candidate c:

$$\text{LH}(c) = \text{freq}(c) - \sum_{t \in T_c} \text{freq}(t) \times \frac{\text{freq}(t)}{\sum_{t \in T_c} \text{freq}(t)}. \tag{3}$$

Here, c is a long-form candidate; freq(c) denotes the frequency of co-occurrence of c with a short-form; and T_c is a set of nested long-form candidates, each of which consists of a preceding word followed by the candidate c. Obviously, for $t \in T_c$, Equation 3 can be explained as:

$$\text{LH}(c) = \text{freq}(c) - \mathbb{E}[\text{freq}(t)]. \tag{4}$$

In this paper, we apply their method on the task of bilingual abbreviation lexicon extraction. Now, the long-form is a Chinese word sequence and the short-form is an English acronym. We filter the parenthetical expressions in the Web pages with several heuristic rules to meet the form of pattern 2 and to save the computing time:

- the short-form (e_1^I) should contain only one English word ($I = 1$), and all letters in which should be capital;
- similar with (Lin et al., 2008), the pre-parenthesis text is trimmed with: $|c| \geq 10 \times |e_1^I| + 6$ when $|e_1^I| \leq 6$, and $|c| \geq 2 \times |e_1^I| + 6$, otherwise. $|c|$ and $|e_1^I|$ are measured in bytes. We further trim the remaining pre-parenthesis text by punctuations other than hyphens and dots, i.e., the right most punctuation and its left subsequence are discarded.

No.	Chinese long-form candidates	LH	T/F
1	肿瘤 相关 抗原 *Tumor-Associated Antigen*	172.5	T
2	硫 代 乙酰 胺 *thioacetamide*	79.9	T
3	胺 *amine*	33.8	F
4	抗原 *antigen*	24.5	F
5	相关 抗原 *associated antigen*	21.2	F
6	的 肿瘤 相关 抗原 *'s Tumor-Associated Antigen*	16.5	F
7	总 氨基酸 *total amino acid*	16.2	T

Table 2: Top-7 Chinese long-form candidates for the English acronym *TAA*, according to the LH score.

4.2 Experiment

We used SogouT Internet Corpus Version 2.0[4], which contains about 13 billion original Web pages (mainly Chinese) in the form of 252 gigabyte .txt files. In addition, we used 55 gigabyte (.txt format) Peking University Chinese Paper Corpus. We constructed a partially parallel corpus in the form of Pattern 1 from the union of the two corpora using the heuristic rules defined in (Lin et al., 2008). We gained a partially parallel corpus which contains 12,444,264 entries.

We extracted 107,856 distinct English acronyms. Limiting LH score ≥ 1.0 in Equation 3, we gained 2,020,012 Chinese long-form candidates for the 107,856 English acronyms. Table 2 illustrates the top-7 Chinese long-form candidates of the English acronym *TAA*. Three candidates are correct (T) long-forms while the other 4 are wrong (F). Wrong candidates from No. 3 to 5 are all subsequences of the correct candidate No. 1. No. 6 includes No. 1 while with a Chinese functional word *de* in the left most side. These error types can be easily tackled with some filtering patterns, such as 'remove the left most functional word in the long-form candidates', 'only keep the relatively longer candidates with larger LH score', etc.

Since there does not yet exists a common evaluation data set for the bilingual abbreviation lexicon, we manually evaluated a small sample of it.

[4]http://www.sogou.com/labs/dl/t.html

Of the 107,856 English acronyms, we randomly selected 200 English acronyms and their top-1 Chinese long-form candidates for manually evaluating. We found, 92 candidates were correct including 3 transliteration examples. Of the 108 wrong candidates, 96 candidates included the correct long-form with some redundant words on the left side (i.e., $c = (\text{word})^+$ correct long-form), the other 12 candidates missed some words of the correct long-form or had some redundant words right before the left parenthesis (i.e., $c = (\text{word})^*$ correct long-form $(\text{word})^+$ or $c = (\text{word})^*$ subsequence of correct long-form $\text{word})^*$). We classified the redundant word right before the correct long-form of each of the 96 candidates, *de* occupied 32, noun occupied 7, verb occupied 18, prepositions and conjunctions occupied the remaining ones.

In total, the abbreviation translation accuracy is 44.5%. We improved the accuracy to 60.5% with an additional *de* filtering pattern. According to former mentioned error analysis, the accuracy may further be improved if a Chinese part-of-speech tagger is employed and the non-nominal words in the long-form are removed beforehand.

5 Self-Training for Transliteration Models

In this section, we first describe and compare three transliteration models. Then, we select and train the best model following Algorithm 1 for lexicon mining. We investigate two things, the scalability of the self-trained model given different amount of initial training data, and the performance of several strategies for selecting new training samples.

5.1 Model description

We construct and compare three forward transliteration models, a phoneme-based model (English phonemes to Chinese pinyins), a grapheme-based model (English syllables to Chinese characters) and a hybrid model (English syllables to Chinese pinyins). Similar models have been compared in (Oh et al., 2006) for English-to-Korean and English-to-Japanese transliteration. All the three models are phrase-based, i.e., adjacent phonemes or graphemes are allowable to form phrase-level transliteration units. Building the correspondences on phrase level can effectively tackle the missing or redundant phoneme/grapheme problem during transliteration. For example, when *Aamodt* is transliterated into *a mō tè*[5], *a* and *d* are missing. The problem can be easily solved when taking *Aa* and *dt* as single units for transliterating.

Making use of Moses (Koehn et al., 2007), a phrase-based SMT system, Matthews (2007) has shown that the performance was comparable to recent state-of-the-art work (Jiang et al., 2007) in English-to-Chinese personal name transliteration. Matthews (2007) took transliteration as translation at the surface level. Inspired by his idea, we also implemented our transliteration models employing Moses. The main difference is that, while Matthews (2007) tokenized the English names into individual letters before training in Moses, we split them into syllables using the heuristic rules described in (Jiang et al., 2007), such that one syllable only contains one vowel letter or a combination of a consonant and a vowel letter.

English syllable sequences are used in the grapheme-based and hybrid models. In the phoneme-based model, we transfer English names into phonemes and Chinese characters into Pinyins in virtue of the CMU pronunciation dictionary[6] and the LDC Chinese character-to-pinyin list[7].

In the mass, the grapheme-based model is the most robust model, since no additional resources are needed. However, it suffers from the Chinese homophonic character problem. For instance, pinyin ai corresponds to numerous Chinese characters which are applicable to personal names. The phoneme-based model is the most suitable model that reflects the essence of transliteration, while restricted by additional grapheme to phoneme dictionaries. In order to eliminate the confusion of Chinese homophonic characters and alleviate the dependency on additional resources, we implement a hybrid model that accepts English syllables and Chinese pinyins as formats of the training data. This model is called hybrid, since English syllables are graphemes and Chinese pinyins are phonemes.

[5]The tones of Chinese pinyins are ignored in our transliteration models for simplicity.

[6]http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[7]http://projects.ldc.upenn.edu/Chinese/docs/char2pinyin.txt

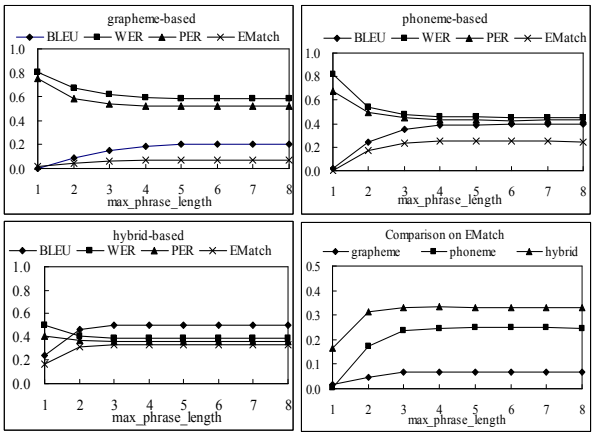

Figure 2: The performances of the transliteration models and their comparison on EMatch.

%	0t	1t	2t	3t	4t	5t	Strategy
5	.3879	.3937	.3971	.3958	.3972	.3971	top1 *em*
		.3911	.3979	.3954	.3974	.3965	top1 *am*
		.4062	.4182	.4208	**.4218**	.4201	top5 *em*
		.3987	.4177	.4190	.4192	.4189	top5 *am*
10	.4092	.4282	.4258	.4202	.4203	.4205	top1 *em*
		.4121	.4190	.4180	.4174	.4200	top1 *am*
		.4305	.4386	.4399	**.4438**	.4403	top5 *em*
		.4289	.4263	.4292	.4291	.4288	top5 *am*
20	.4561	.4538	.4562	.4550	.4543	.4551	top1 *em*
		.4532	.4578	.4544	.4545	.4541	top1 *am*
		.4624	**.4762**	.4754	.4748	.4746	top5 *em*
		.4605	.4677	.4677	.4674	.4679	top5 *am*
40	.4779	.4791	.4793	.4799	.4794	.4808	top1 *em*
		.4774	.4794	.4779	.4789	.4784	top1 *am*
		.4808	**.4811**	.4791	.4795	.4790	top5 *em*
		.4775	.4778	.4781	.4785	.4779	top5 *am*
60	.5032	.4939	.5004	.5012	.5012	.5016	top1 *em*
		.4919	.4988	.4990	.4994	.4990	top1 *am*
		.5013	.5063	.5059	**.5066**	.5065	top5 *em*
		.4919	.4960	.4970	.4977	.4962	top5 *am*
80	.5038	.4984	.4984	.5004	.5006	.4995	top1 *em*
		.4916	.4916	.4914	.4915	.4916	top1 *am*
		.5039	.5037	.5053	**.5054**	.5042	top5 *em*
		.4950	.5028	.5027	.5032	.5032	top5 *am*
100	.5045	.5077	.5053	.5067	.5063	.5066	top1 *em*
		.5045	.5054	.5046	.5050	.5055	top1 *am*
		.5108	.5102	.5111	.5108	**.5115**	top5 *em*
		.5105	.5106	.5100	.5094	.5109	top5 *am*

Table 3: The BLEU score of self-trained h4 transliteration models under four selection strategies. nt (n=1..5) stands for the n-th iteration.

5.2 Experimental model selection

Similar to (Jiang et al., 2007), the transliteration models were trained and tested on the LDC Chinese-English Named Entity Lists Version 1.0[8]. The original list contains 572,213 English people names with Chinese transliterations. We extracted 74,725 entries in which the English names also appeared in the CMU pronunciation dictionary. We randomly selected 3,736 entries as an open testing set and the remaining entries as a training set[9]. The results were evaluated using the character/pinyin-based 4-gram BLEU score (Papineni et al., 2002), word error rate (WER), position independent word error rate (PER), and exact match (EMatch).

Figure 2 reports the performances of the three models and the comparison based on EMatch. From the results, we can easily draw the conclusion that the hybrid model performs the best under the maximal phrase length (mpl, the maximal phrase length allowed in Moses) from 1 to 8. The performances of the models converge at or right after mpl = 4. The pinyin-based WER of the hybrid model is 39.13%, comparable to the pinyin error rate 39.6%, reported in (Jiang et al., 2007)[10]. Thus, our further self-training experiments are pursued on the hybrid model taking mpl to be 4 (short for h4, hereafter).

5.3 Experiments on the self-trained hybrid model

As former mentioned, we investigate the scalability of the self-trained h4 model by respectively using 5, 10, 20, 40, 60, 80, and 100 percent of initial training data, and the performances of using exact matching (*em*) or approximate matching (*am*, line 8 in Algorithm 1) on the top-1 and top-5 outputs (line 7 in Algorithm 1) for selecting new training samples. We used edit distance (ed) to measure the *em* and *am* similarities:

$$ed(c, C') = 0 \text{ or } < \text{syllable\_number}(C')/2. \quad (5)$$

When applying Algorithm 1 for transliteration lexicon mining, we decode each word in e_1^I respectively. The algorithm terminated in five iterations when we set the terminal threshold ϵ (Line 13 in Algorithm 1) to be 100.

For simplicity, Table 3 only illustrates the BLEU score of h4 models under four selection strategies. From this table, we can draw the following conclusions. First, with fewer initial training data, the improvement is better. The best relative improvements

[8]Linguistic Data Consortium catalog number: LDC2005T34 (former catalog number: LDC2003E01)

[9]Jiang et al. (2007) selected 25,718 personal name pairs from LDC2003E01 as the experiment data: 200 as development set, 200 as test set, and the remaining entries as training set.

[10]It should be notified that we achieved this result by using larger training set (70,989 vs. 25,718) and larger test set (3,736 vs. 200) comparing with (Jiang et al., 2007), and we did not use additional Web resources as Jiang et al. (2007) did.

are 8.74%, 8.46%, 4.41%, 0.67%, 0.68%, 0.32%, and 1.39%, respectively. Second, using top-5 and *em* for new training data selection performs the best among the four strategies. Compared under each iteration, using top-5 is better than using top-1; *em* is better than *am*; and top-5 with *am* is a little better than top-1 with *em*. We mined 39,424, 42,466, 46,116, 47,057, 49,551, 49,622, and 50,313 distinct entries under the six types of initial data with top-5 plus *em* strategy. The 50,313 entries are taken as the final transliteration lexicon for further comparison.

6 Self-Training for a Cascaded Translation Model

We classify the parenthetical translation candidates by employing a translation model. In contrast to (Lin et al., 2008), wherein the lengthes of prefixes and suffixes of English words were assumed to be three bytes, we segment words into morphemes (sequences of prefixes, stems, and suffixes) by Morfessor 0.9.2[11], an unsupervised language-independent morphological analyzer (Creutz and Lagus, 2007). We use the morpheme-level translation similarity explicitly in our cascaded translation model (Wu et al., 2008), which makes use of morpheme, word, and phrase level translation units. We train Moses to gain a phrase-level translation table. To gain a morpheme-level translation table, we run GIZA++ (Och and Ney, 2003) on both directions between English morphemes and Chinese characters, and take the intersection of *Viterbi* alignments. The English-to-Chinese translation probabilities computed by GIZA++ are attached to each morpheme-character element in the intersection set.

6.1 Experiment

The Wanfang Chinese-English technical term dictionary[12], which contains 525,259 entries in total, was used for training and testing. 10,000 entries were randomly selected as the test set and the remaining as the training set. Again, we investigated the scalability of the self-trained cascaded translation model by respectively using 20, 40, 60, 80, and 100 percent of initial training data. An aggressive similarity measurement was used for selecting new training samples:

$$\text{first\_char}(c) = \text{first\_char}(C') \ \wedge \ min\{ed(c, C')\}. \quad (6)$$

Here, we judge if the first characters of c and C' are similar or not. c was gained by deleting zero or more characters from the left side of f_1^J. When more than one c satisfied this condition, the c that had the smallest edit distance with C' was selected. When applying Algorithm 1 for translation lexicon mining, we took e_1^I as one input for decoding instead of decoding each word respectively. Only the top-1 output (C') was used for comparing. The algorithm stopped in five iterations when we set the terminal threshold ϵ to be 2000.

%	0t	1t	2t	3t	4t	5t
20	**.1406**	.1196	.1243	.1239	.1176	.1179
40	.1091	.1224	.1386	.1345	**.1479**	.1466
60	.1630	.1624	.1429	**.1714**	.1309	.1398
80	**.1944**	.1783	.1886	.1870	.1884	.1873
100	.1810	.1814	.1539	**.1981**	.1542	.1944

Table 4: The BLEU score of self-trained cascaded translation model under five initial training sets.

For simplicity, Table 4 only illustrates the BLEU score of the cascaded translation model under five initial training sets. For the reason that there are finite phonemes in English and Chinese while the semantic correspondences between the two languages tend to be infinite, Table 4 is harder to be analyzed than Table 3. When initially using 40%, 60%, and 100% training data for self-training, the results tend to be better at some iterations. We gain 35.6%, 5.2%, and 9.4% relative improvements, respectively. However, the results tend to be worse when 20% and 80% training data were used initially, with 11.6% and 3.0% minimal relative loss. The best BLEU scores tend to be better when more initial training data are available. We mined 1,038,617, 1,025,606, 1,048,761, 1,056,311, and 1,060,936 distinct entries under the five types of initial training data. The 1,060,936 entries are taken as the final translation lexicon for further comparison.

7 Wikipedia Evaluation

We have mined three kinds of lexicons till now, an abbreviation lexicon containing 107,856 dis-

[11] http://www.cis.hut.fi/projects/morpho/

[12] http://www.wanfangdata.com.cn/Search/ResourceBrowse.aspx

	En. to Ch.		Ch. to En.	
	Cov	EMatch	Cov	EMatch
Our Lexicon	22.8%	5.2%	23.2%	5.5%
Unsupervised	23.5%	5.4%	24.0%	5.4%

Table 5: The results of our lexicon and an unsupervised-mined lexicon (Lin et al., 2008) evaluated under Wikipedia title dictionary. Cov is short for coverage.

similar English acronyms with 2,020,012 Chinese long-form candidates; a transliteration lexicon with 50,313 distinct entries; and a translation lexicon with 1,060,936 distinct entries. The three lexicons are combined together as our final lexicon.

Similar with (Lin et al., 2008), we compare our final mined lexicon with a dictionary extracted from Wikipedia, the biggest multilingual free-content encyclopedia on the Web. We extracted the titles of Chinese and English Wikipedia articles[13] that are linked to each other. Since most titles contain less than five words, we take a linked title pair as a translation entry without considering the word alignment relation between the words inside the titles. The result lexicon contains 105,320 translation pairs between 103,823 Chinese titles and 103,227 English titles. Obviously, only a small percentage of titles have more than one translation. Whenever there is more than one translation, we take the candidate entry as correct if and only if it matches one of the translations.

Moreover, we compare our semi-supervised approach with an unsupervised approach (Lin et al., 2008). Lin et al. (2008) took $\varphi^2(f_j, e_i)$ score [14](Gale and Church, 1991) with threshold 0.001 as the word alignment probability in a word alignment algorithm, Competitive Link. Competitive Link tries to align an unlinked e_i with an unlinked f_j by the condition that $\varphi^2(f_j, e_i)$ is the biggest. Lin et al. (2008) relaxed the unlinked constraints to allow consecutive sequence of words on one side to be linked to the same word on the other side[15]. The left boundary inside f_1^J is determined when each e_i in e_1^I is aligned. After applying the modified Competitive Link on the partially parallel corpus which includes 12,444,264 entries (Section 4.2), we obtained 2,628,366 distinct pairs.

Table 5 shows the results of the two lexicons evaluated under Wikipedia title dictionary. The coverage is measured by the percentage of titles which appears in the mined lexicon. We then check whether the translation in the mined lexicon is an exact match of one of the translations in the Wikipedia lexicon. Through comparing the results, our mined lexicon is comparable with the lexicon mined in an unsupervised way. Since the selection is based on phonemic and semantic clues instead of frequency, a parenthetical translation candidate will not be selected if the in-parenthetical English text is failed to be transliterated or translated. This is one reason that explains why we earned a little lower coverage. Another reason comes from the low coverage rate of seed lexicons used for self-training, only 8.65% English words in the partially parallel corpus are covered by the Wanfang dictionary.

8 Conclusion

We have proposed a semi-supervised learning framework for mining bilingual lexicons from parenthetical expressions in monolingual Web pages. We classified the parenthesis expressions into three categories: abbreviation, transliteration, and translation. A set of heuristic rules, a self-trained hybrid transliteration model, and a self-trained cascaded translation model were proposed for each category, respectively.

We investigated the scalability of the self-trained transliteration and translation models by training them with different amount of data. The results shew the stability (transliteration) and feasibility (translation) of our proposals. Through employing the parallel Wikipedia article titles as a gold standard lexicon, we gained the comparable results comparing our semi-supervised framework with our implementation of Lin et al. (2008)'s unsupervised mining approach.

[13]English and Chinese Wikipedia pages due to 2008.09.23 are used here.

[14]$\varphi^2(f_j, e_i) = \frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)}$, where a is the number of $f_1^J(e_1^I)$ containing both e_i and f_j; $(a+b)$ is the number of $f_1^J(e_1^I)$ containing e_i; $(a+c)$ is the number of $f_1^J(e_1^I)$ containing f_j; and d is the number of $f_1^J(e_1^I)$ containing neither e_i nor f_j.

[15]Instead of requiring both e_i and f_j to have no previous linkages, they only require that at least one of them be unlinked and that (suppose e_i is unlinked and f_j is linked to e_k) none of the words between e_i and e_k be linked to any word other than f_j.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan) and Japanese/Chinese Machine Translation Project in Special Coordination Funds for Promoting Science and Technology (MEXT, Japan). We thank the anonymous reviewers for their constructive comments.

References

Cao, Guihong, Jianfeng Gao, and Jian-Yun Nie. 2007. A system to Mine Large-Scale Bilingual Dictionaries from Monolingual Web Pages. In *MT Summit XI*. pages 57–64, Copenhagen, Denmark.

Cheng, Pu-Jen, Yi-Cheng Pan, Wen-Hsiang Lu, and Lee-Feng Chien. 2004. Creating Multilingual Translation Lexicons with Regional Variations Using Web Corpora. In *ACL 2004*, pages 534–541, Barcelona, Spain.

Creutz, Mathias and Krista Lagus. 2007. Unsupervised Models for Morpheme Segmentation and Morphology Learning. *ACM Transactions on Speech and Language Processing*, 4(1):Article 3.

Dempster, A. P., N. M. Laird and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

Gale, W. and K. Church. 1991. Identifying word correspondence in parallel text. In *DARPA NLP Workshop*.

Gao, Jianfeng, Mu Li, Andi Wu, and Chang-Ning Huang. 2006. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. *Computational Linguistics*, 31(4):531–574.

Haghighi, Aria, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein 2008. Learning Bilingual Lexicons from Monolingual Corpora. In *ACL-08:HLT*. pages 771–779, Columbus, Ohio.

Jiang, Long, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named Entity Translation with Web Mining and Transliteration. In *IJCAI 2007*. pages 1629–1634, Hyderabad, India.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007 Poster Session*, pages 177–180.

Koehn, Philipp and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *SIGLEX 2002*, pages 9–16.

Kuo, Jin-Shea, Haizhou Li, and Ying-Kuei Yang. 2006. Learning Transliteration Lexicons from the Web. In *COLING-ACL 2006*. pages 1129–1136.

Lin, Dekang, Shaojun Zhao, Benjamin Van Durme, and Marius Paşca. 2008. Mining Parenthetical Translations from the Web by Word Alignment. In *ACL-08:HLT*, pages 994–1002, Columbus, Ohio.

Matthews, David. 2007. Machine Transliteration of Proper Names. A Thesis of Master. University of Edinburgh.

McClosky, David, Eugene Charniak, and Mark Johnson 2008. When is Self-Training Effective for Parsing? In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 561–568, manchester, UK.

Melamed, I. Dan. 2000. Models of Translational Equivalence among Words. *Computational Linguistics*, 26(2):221–249.

Och, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Oh, Jong-Hoon, Key-Sun Choi, and Hitoshi Isahara. 2006. A Comparison of Different Machine Transliteration Models. *Journal of Artifical Intelligence Research*, 27:119–151.

Okazaki, Naoaki and Sophia Ananiadou. 2006. Building an Abbreviation Dictionary Using a Term Recognition Approach. *Bioinformatics*, 22(22):3089–3095.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 311–318, Philadelphia.

Shao, Li and Hwee Tou Ng. 2004. Mining New Word Translations from Comparable Corpora. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 618–624, Geneva, Switzerland.

Wu, Xianchao, Naoaki Okazaki, Takashi Tsunakawa, and Jun'ichi Tsujii. 2008. Improving English-to-Chinese Translation for Technical Terms Using Morphological Information. In *Proceedings of the 8th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 202–211, Waikiki, Hawai'i.

Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts.

Zhu, Xiaojin. 2007. *Semi-Supervised Learning Literature Survery*. University of Wisconsin - Madison.