Unsupervised Translation Induction for Chinese Abbreviations using Monolingual Corpora

Zhifei Li and **David Yarowsky**
Department of Computer Science and Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD 21218, USA
zhifei.work@gmail.com and yarowsky@cs.jhu.edu

Abstract

Chinese abbreviations are widely used in modern Chinese texts. Compared with English abbreviations (which are mostly acronyms and truncations), the formation of Chinese abbreviations is much more complex. Due to the richness of Chinese abbreviations, many of them may not appear in available parallel corpora, in which case current machine translation systems simply treat them as unknown words and leave them untranslated. In this paper, we present a novel *unsupervised* method that automatically extracts the relation between a full-form phrase and its abbreviation from monolingual corpora, and induces translation entries for the abbreviation by using its full-form as a bridge. Our method does not require any additional annotated data other than the data that a regular translation system uses. We integrate our method into a state-of-the-art baseline translation system and show that it consistently improves the performance of the baseline system on various NIST MT test sets.

1 Introduction

The modern Chinese language is a highly abbreviated one due to the mixed use of ancient single-character words with modern multi-character words and compound words. According to Chang and Lai (2004), approximately 20% of sentences in a typical news article have abbreviated words in them. Abbreviations have become even more popular along with the development of Internet media (e.g., online chat, weblog, newsgroup, and so on). While English words are normally abbreviated by either their first letters (i.e. acronyms) or via truncation, the formation of Chinese abbreviations is much more complex. Figure 1 shows two examples for Chinese abbreviations. Clearly, an abbreviated form of a word can be obtained by selecting *one or more* characters from this word, and the selected characters can be at *any* position in the word. In an extreme case, there are even re-ordering between a full-form phrase and its abbreviation.

Full-form	Abbreviation	Translation
香港 总督	港督	Hong Kong Governor
安全 理事会	安理会	Security Council

Figure 1: Chinese Abbreviations Examples

While the research in statistical machine translation (SMT) has made significant progress, most SMT systems (Koehn et al., 2003; Chiang, 2007; Galley et al., 2006) rely on parallel corpora to extract translation entries. The richness and complexness of Chinese abbreviations imposes challenges to the SMT systems. In particular, many Chinese abbreviations may not appear in available parallel corpora, in which case current SMT systems treat them as unknown words and leave them untranslated. This affects the translation quality significantly.

To be able to translate a Chinese abbreviation that is unseen in available parallel corpora, one may annotate more parallel data. However, this is very expensive as there are too many possible abbreviations and new abbreviations are constantly created. Another approach is to transform the abbreviation

into its full-form for which the current SMT system knows how to translate. For example, if the baseline system knows that the translation for "香港 总督" is "Hong Kong Governor", and it also knows that "港督" is an abbreviation of "香**港** 总**督**" , then it can translate "港督" to "Hong Kong Governor".

Even if an abbreviation has been seen in parallel corpora, it may still be worth to consider its full-form phrase as an *additional* alternative to the abbreviation since abbreviated words are normally semantically ambiguous, while its full-form contains more context information that helps the MT system choose a right translation for the abbreviation.

Conceptually, the approach of translating an abbreviation by using its full-form as a bridge involves four components: identifying abbreviations, learning their full-forms, inducing their translations, and integrating the abbreviation translations into the baseline SMT system. None of these components is trivial to realize. For example, for the first two components, we may need manually annotated data that tags an abbreviation with its full-form. We also need to make sure that the baseline system has at least one valid translation for the full-form phrase. On the other hand, integrating an additional component into a baseline SMT system is notoriously tricky as evident in the research on integrating word sense disambiguation (WSD) into SMT systems: different ways of integration lead to conflicting conclusions on whether WSD helps MT performance (Chan et al., 2007; Carpuat and Wu, 2007).

In this paper, we present an unsupervised approach to translate Chinese abbreviations. Our approach exploits the *data co-occurrence* phenomena and does not require any additional annotated data except the parallel and monolingual corpora that the baseline SMT system uses. Moreover, our approach integrates the abbreviation translation component into the baseline system in a natural way, and thus is able to make use of the minimum-error-rate training (Och, 2003) to automatically adjust the model parameters to reflect the change of the integrated system over the baseline system. We carry out experiments on a state-of-the-art SMT system, i.e., Moses (Koehn et al., 2007), and show that the abbreviation translations consistently improve the translation performance (in terms of BLEU (Papineni et al., 2002)) on various NIST MT test sets.

2 Background: Chinese Abbreviations

In general, Chinese abbreviations are formed based on three major methods: *reduction*, *elimination* and *generalization* (Lee, 2005; Yin, 1999). Table 1 presents examples for each category.

Among the three methods, *reduction* is the most popular one, which generates an abbreviation by selecting one or more characters from each of the words in the full-form phrase. The selected characters can be at any position of the word. Table 1 presents examples to illustrate how characters at different positions are selected to generate abbreviations. While the abbreviations mostly originate from noun phrases (in particular, named entities), other general phrases are also abbreviatable. For example, the second example "Save Energy" is a verb phrase. In an extreme case, reordering may happen between an abbreviation and its full-form phrase. For example, for the seventh example in Table 1, a monotone abbreviation should be "一核厂", however, "核一厂" is a more popular ordering in Chinese texts.

In *elimination*, one or more words of the original full-form phrase are eliminated and the rest parts remain as an abbreviation. For example, in the full-form phrase "清华 大学", the word "大学" is eliminated and the remaining word "清华" alone becomes the abbreviation.

In *generalization*, an abbreviation is created by generalizing parallel sub-parts of the full-form phrase. For example, "三防 (three preventions)" in Table 1 is an abbreviation for the phrase "防火、防盗、防交通事故 (fire prevention, theft prevention, and traffic accident prevention)". The character "防 (prevention)" is common to the three sub-parts of the full-form, so it is being generalized.

3 Unsupervised Translation Induction for Chinese Abbreviations

In this section, we describe an *unsupervised* method to induce translation entries for Chinese abbreviations, even when these abbreviations never appear in the Chinese side of the parallel corpora. Our basic idea is to automatically extract the relation between a full-form phrase and its abbreviation (we refer the relation as *full-abbreviation*) from monolingual corpora, and then induce translation entries for the abbreviation by using its full-form phrase as a bridge.

Category	Full-form	Abbreviation	Translation
Reduction	北京 大学	北大	Peking University
	节约 能源	节能	Save Energy
	香港 总督	港督	Hong Kong Governor
	外交 部长	外长	Foreign Minister
	人民 警察	民警	People's Police
	安全 理事会	安理会	Security Council
	第一 核能 发电厂	核一厂	No.1 Nuclear Energy Power Plant
Elimination	清华 大学	清华	Tsinghua University
Generalization	防火、防盗、防交通事故	三防	Three Preventions

Table 1: Chinese Abbreviation: Categories and Examples

Our approach involves five major steps:

- Step-1: extract a list of English entities from English monolingual corpora;
- Step-2: translate the list into Chinese using a baseline translation system;
- Step-3: extract *full-abbreviation* relations from Chinese monolingual corpora by treating the Chinese translations obtained in Step-2 as full-form phrases;
- Step-4: induce translation entries for Chinese abbreviations by using their full-form phrases as bridges;
- Step-5: augment the baseline system with translation entries obtained in Step-4.

Clearly, the main purpose of Step-1 and -2 is to obtain a list of Chinese entities, which will be treated as full-form phrases in Step-3. One may use a named entity tagger to obtain such a list. However, this relies on the existence of a Chinese named entity tagger with high-precision. Moreover, obtaining a list using a dedicated tagger does not guarantee that the baseline system knows how to translate the list. On the contrary, in our approach, since the Chinese entities are translation outputs for the English entities, it is ensured that the baseline system has translations for these Chinese entities.

Regarding the data resource used, Step-1, -2, and -3 rely on the English monolingual corpora, parallel corpora, and the Chinese monolingual corpora, *respectively*. Clearly, our approach does not require any additional annotated data compared with the baseline system. Moreover, our approach utilizes both Chinese and English monolingual data to help MT, while most SMT systems utilizes only the English monolingual data to build a language model. This is particularly interesting since we normally have enormous monolingual data, but a small amount of parallel data. For example, in the translation task between Chinese and English, both the Chinese and English Gigaword have billions of words, but the parallel data has only about 30 million words.

Step-4 and -5 are natural ways to integrate the abbreviation translation component with the baseline translation system. This is critical to make the abbreviation translation get performance gains over the baseline system as will be clear later.

In the remainder of this section, we will present a specific instantiation for each step.

3.1 English Entity Extraction from English Monolingual Corpora

Though one can exploit a sophisticated named-entity tagger to extract English entities, in this paper we identify English entities based on the capitalization information. Specifically, to be considered as an entity, a continuous span of English words must satisfy the following conditions:

- all words must start from a capital letter except for function words "of", "the", and "and";
- each function word can appear only once;
- the number of words in the span must be smaller than a threshold (e.g., 10);
- the occurrence count of this span must be greater than a threshold (e.g., 1).

3.2 English Entity Translation

For the Chinese-English language pair, most MT research is on translation from Chinese to English, but here we need the reverse direction. However, since most of statistical translation models (Koehn et al., 2003; Chiang, 2007; Galley et al., 2006) are *symmetrical*, it is relatively easy to train a translation system to translate from English to Chinese, except that we need to train a Chinese language model from the Chinese monolingual data.

It is worth pointing out that the baseline system may not be able to translate all the English entities. This is because the entities are extracted from the English monolingual corpora, which has a much larger vocabulary than the English side of the parallel corpora. Therefore, we should remove all the Chinese translations that contain any untranslated English words before proceeding to the next step. Moreover, it is desirable to generate an n-best list instead of a 1-best translation for the English entity.

3.3 *Full-abbreviation* Relation Extraction from Chinese Monolingual Corpora

We treat the Chinese entities obtained in Section 3.2 as full-form phrases. To identify their abbreviations, one can employ an HMM model (Chang and Teng, 2006). Here we propose a much simpler approach, which is based on the *data co-occurrence* intuition.

3.3.1 Data Co-occurrence

In a monolingual corpus, relevant words tend to appear together (i.e., co-occurrence). For example, *Bill Gates* tends to appear together with *Microsoft*. The co-occurrence may imply a relationship (e.g., *Bill Gates* is the founder of *Microsoft*). By inspection of the Chinese text, we found that the *data co-occurrence* phenomena also applies to the *full-abbreviation* relation. Table 2 shows an example, where the abbreviation "冬奥会" appears in the title while its full-form "冬季奥运会" appears in the text of the same document. In general, the occurrence distance between an abbreviation and its full-form varies. For example, they may appear in the same sentence, or in the neighborhood sentences.

Title	都灵**冬奥会**开幕式将激情上演
Text	新华社都灵2月9日电(记者丁莹阎涛)第20届**冬季奥运会**的开幕式将于当地时间10日晚8点在都灵奥林匹克体育场正式揭开神秘的面纱。

Table 2: Data Co-occurrence Example for the *Full-abbreviation* Relation (冬季奥运会,冬奥会) meaning "winter olympics"

3.3.2 *Full-abbreviation* Relation Extraction Algorithm

By exploiting the *data co-occurrence* phenomena, we identify possible abbreviations for full-form phrases. Figure 2 presents the pseudocode of the *full-abbreviation* relation extraction algorithm.

Relation-Extraction$(Corpus, Full\text{-}list)$

```
1  contexts ← NIL
2  for i ← 1 to length[Corpus]
3      sent1 ← Corpus[i]
4      contexts ← UPDATE(contexts, Corpus, i)
5      for full in sent1
6          if full in Full-list
7              for sent2 in contexts
8                  for abbr in sent2
9                      if RL(full, abbr) = TRUE
10                         Count[abbr, full]++
11 return Count
```

Figure 2: *Full-abbreviation* Relation Extraction

Given a monolingual corpus and a list of full-form phrases (i.e., $Full\text{-}list$, which is obtained in Section 3.2), the algorithm returns a $Count$ that contains *full-abbreviation* relations and their occurrence counts. Specifically, the algorithm linearly scans over the whole corpus as indicated by line 1. Along the linear scan, the algorithm maintains $contexts$ of the current sentence (i.e., $sent1$), and the $contexts$ remember the sentences from where the algorithm identifies possible abbreviations. In our implementation, the $contexts$ include current sentence, the title of current document, and previous and next sentence in the document. Then, for each ngram (i.e., *full*) of the current sentence (i.e., $sent1$) and for each ngram (i.e., $abbr$) of a context sentence (i.e., $sent2$), the algorithm calls a function RL, which decides whether the *full-abbreviation* relation holds between *full* and $abbr$. If RL returns TRUE, the count table

(i.e., $Count$) is incremented by one for this relation. Note that the filtering through the full-form phrases list (i.e., $Full\text{-}list$) as shown in line 6 is the key to make the algorithm efficient enough to run through large-size monolingual corpora.

In function RL, we run a simple alignment algorithm that links the characters in $abbr$ with the words in *full*. In the alignment, we assume that there is no reordering between *full* and $abbr$. To be considered as a valid *full-abbreviation* relation, *full* and $abbr$ must satisfy the following conditions:

- $abbr$ must be shorter than *full* by a relative threshold (e.g., 1.2);
- each character in $abbr$ must be aligned to *full*;
- each word in *full* must have at least one character aligned to $abbr$;
- $abbr$ must *not* be a continuous sub-part of *full*;

Clearly, due to the above conditions, our approach may not be able to handle all possible abbreviations (e.g., the abbreviations formed by the *generalization* method described in Section 2). One can modify the conditions and the alignment algorithm to handle more complex *full-abbreviation* relations.

With the count table $Count$, we can calculate the relative frequency and get the following probability,

$$P(full|abbr) = \frac{Count[abbr, full]}{\sum Count[abbr, *]} \quad (1)$$

3.4 Translation Induction for Chinese Abbreviations

Given a Chinese abbreviation and its full-form, we induce English translation entries for the abbreviation by using the full-form as a bridge. Specifically, we first generate n-best translations for each full-form Chinese phrase using the baseline system.[1] We then post-process the translation outputs such that they have the same format (i.e., containing the same set of model features) as a regular phrase entry in the baseline phrase table. Once we get the translation entries for the full-form, we can replace the full-form Chinese with its abbreviation to generate translation entries for the abbreviation. Moreover, to deal with the case that an abbreviation may have several candidate full-form phrases, we normalize the feature values using the following equation,

$$\Phi_j(e, abbr) = \Phi_j(e, full) \times P(full|abbr) \quad (2)$$

where e is an English translation, and Φ_j is the j-th model feature indexed as in the baseline system.

3.5 Integration with Baseline Translation System

Since the obtained translation entries for abbreviations have the same format as the regular translation entries in the baseline phrase table, it is relatively easy to add them into the baseline phrase table. Specifically, if a translation entry (signatured by its Chinese and English strings) to be added is not in the baseline phrase table, we simply add the entry into the baseline table. On the other hand, if the entry is already in the baseline phrase table, then we *merge* the entries by *enforcing* the translation probability as we obtain the same translation entry from two different knowledge sources (one is from parallel corpora and the other one is from the Chinese monolingual corpora).

Once we obtain the augmented phrase table, we should run the minimum-error-rate training (Och, 2003) with the augmented phrase table such that the model parameters are properly adjusted. As will be shown in the experimental results, this is critical to obtain performance gain over the baseline system.

4 Experimental Results

4.1 Corpora

We compile a parallel dataset which consists of various corpora distributed by the Linguistic Data Consortium (LDC) for NIST MT evaluation. The parallel dataset has about 1M sentence pairs, and about 28M words. The monolingual data we use includes the English Gigaword V2 (LDC2005T12) and the Chinese Gigaword V2 (LDC2005T14).

4.2 Baseline System Training

Using the toolkit Moses (Koehn et al., 2007), we built a phrase-based baseline system by following

[1] In our method, it is guaranteed that each Chinese full-form phrase will have at least one English translation, i.e., the English entity that has been used to produce this full-form phrase. However, it does not mean that this English entity is the *best* translation that the baseline system has for the Chinese full-form phrase. This is mainly due to the *asymmetry* introduced by the different LMs in different translation directions.

the standard procedure: running GIZA++ (Och and Ney, 2000) in both directions, applying refinement rules to obtain a many-to-many word alignment, and then extracting and scoring phrases using heuristics (Och and Ney, 2004). The baseline system has eight feature functions (see Table 8). The feature functions are combined under a log-linear framework, and the weights are tuned by the minimum-error-rate training (Och, 2003) using BLEU (Papineni et al., 2002) as the optimization metric.

To handle different directions of translation between Chinese and English, we built two trigram language models with modified Kneser-Ney smoothing (Chen and Goodman, 1998) using the SRILM toolkit (Stolcke, 2002).

4.3 Statistics on Intermediate Steps

As described in Section 3, our approach involves five major steps. Table 3 reports the statistics for each intermediate step. While about 5M English entities are extracted and 2-best Chinese translations are generated for each English entity, we get only 4.7M Chinese entities. This is because many of the English entities are untranslatable by the baseline system. The number of *full-abbreviation* relations[2] extracted from the Chinese monolingual corpora is 51K. For each full-form phrase we generate 5-best English translations, however only 210k ($<$51K$\times$5) translation entries are obtained. This is because the baseline system may have less than 5 unique translations for some of the full-form phrases. Lastly, the number of translation entries added due to abbreviations is very small compared with the total number of translation entries (i.e., 50M).

Measure	Value
number of English entities	5M
number of Chinese entities	4.7M
number of *full-abbreviation* relations	51K
number of translation entries added	210K
total number of translation entries	50M

Table 3: Statistics on Intermediate Steps

[2]Note that many of the "abbreviations" extracted by our algorithm are not true abbreviations in the linguistic sense, instead they are just continuous-span of words. This is analogous to the concept of "phrase" in phrase-based MT.

4.4 Precision on *Full-abbreviation* Relations

Table 4 reports the precision on the extracted *full-abbreviation* relations. We classify the relations into several classes based on their occurrence counts. In the second column, we list the fraction of the relations in the given class among all the relations we have extracted (i.e., 51K relations). For each class, we randomly select 100 relations, manually tag them as correct or wrong, and then calculate the precision. Intuitively, a class that has a higher occurrence count should have a higher precision, and this is generally true as shown in the fourth column of Table 4. In comparison, Chang and Teng (2006) reports a precision of 50% over relations between *single-word* full-forms and *single-character* abbreviations. One can imagine a much lower precision on general relations (e.g., the relations between *multi-word* full-forms and *multi-character* abbreviations) that we consider here. Clearly, our results are very competitive[3].

Count	Fraction (%)	Precision (%)	
		Baseline	Ours
$(0, 1]$	35.2	8.9	42.6
$(1, 5]$	33.8	7.8	54.4
$(5, 10]$	10.7	8.9	60.0
$(10, 100]$	16.5	7.6	55.9
$(100, +\infty)$	3.8	12.1	59.9
Average Precision (%)		8.4	**51.3**

Table 4: *Full-abbreviation* Relation Extraction Precision

To further show the advantage of our relation extraction algorithm (see Section 3.3), in the third column of Table 4 we report the results on a simple baseline. To create the baseline, we make use of the *dominant* abbreviation patterns shown in Table 5, which have been reported in Chang and Lai (2004). The abbreviation pattern is represented using the format "(*bit pattern*|*length*)" where the *bit pattern* encodes the information about how an abbreviated form is obtained from its original full-form *word*, and the *length* represents the number of characters in the full-form *word*. In the *bit pattern*, a "1" indicates that the character at the corresponding position of the full-form word is kept in the abbreviation, while a "0" means the character is deleted. Now we dis-

[3]However, it is not a strict comparison because the dataset is different and the *recall* may also be different.

Pattern	Fraction (%)	Example
(1|1)	100	(中, 中)
(10|2)	87	(亚洲, 亚)
(101|3)	44	(理事会, 理会)
(1010|4)	56	(公民投票, 公投)

Table 5: Dominant Abbreviation Patterns reported in Chang and Lai (2004)

cuss how to create the baseline. For each full-form phrase in the randomly selected relations, we generate a baseline hypothesis (i.e., abbreviation) as follows. We first generate an abbreviated form for each *word* in the full-form *phrase* by using the dominant abbreviation pattern, and then concatenate these abbreviated words to form a baseline abbreviation for the full-form *phrase*. As shown in Table 4, the baseline performs significantly worse than our relation extraction algorithm. Compared with the baseline, our relation extraction algorithm allows arbitrary abbreviation patterns as long as they satisfy the alignment constraints. Moreover, our algorithm exploits the *data co-occurrence* phenomena to generate and rank hypothesis (i.e., abbreviation). The above two reasons explain the large performance gain.

It is interesting to examine the statistics on abbreviation patterns over the relations automatically extracted by our algorithm. Table 6 reports the statistics. We obtain the statistics on the relations that are manually tagged as correct before, and there are in total 263 unique *words* in the corresponding full-form *phrases*. Note that the results here are highly biased to our relation extraction algorithm (see Section 3.3). For the statistics on *manually* collected examples, please refer to Chang and Lai (2004).

4.5 Results on Translation Performance

4.5.1 Precision on Translations of Chinese Full-form Phrases

For the relations manually tagged as correct in Section 4.4, we manually look at the top-5 translations for the full-form phrases. If the top-5 translations contain at least one correct translation, we tag it as correct, otherwise as wrong. We get a precision of 97.5%. This precision is extremely high because the BLEU score (precision with brevity penalty) that one obtains for a Chinese sentence is normally between 30% to 50%. Two reasons explain such a high

Pattern	Fraction (%)	Example
(1|1)	100	(中, 中)
(10|2)	74.3	(亚洲, 亚)
(01|2)	7.6	(北京, 京)
(11|2)	18.1	(早餐, 早餐)
(100|3)	58.5	(伊拉克, 伊)
(010|3)	3.1	(行政院, 政)
(001|3)	4.6	(研究所, 所)
(110|3)	13.8	(奥运会, 奥运)
(101|3)	3.1	(理事会, 理会)
(111|3)	16.9	(科学家, 科学家)

Table 6: Statistics on Abbreviation Patterns

precision. Firstly, the full-form phrase is short compared with a regular Chinese sentence, and thus it is easier to translate. Secondly, the full-form phrase itself contains enough context information that helps the system choose a right translation for it. In fact, this shows the importance of considering the full-form phrase as an *additional* alternative to the abbreviation even if the baseline system already has translation entries for the abbreviation.

4.5.2 BLEU on NIST MT Test Sets

We use MT02 as the development set[4] for minimum error rate training (MERT) (Och, 2003). The MT performance is measured by lower-case 4-gram BLEU (Papineni et al., 2002). Table 7 reports the results on various NIST MT test sets. As shown in the table, our Abbreviation Augmented MT (AAMT) systems perform consistently better than the baseline system (described in Section 4.2).

Task	Baseline	AAMT	
		No MERT	With MERT
MT02	29.87	29.96	**30.46**
MT03	29.03	29.23	**29.71**
MT04	29.05	29.88	**30.55**
Average Gain		+0.52	**+1.18**

Table 7: MT Performance measured by BLEU Score

As clear in Table 7, it is important to re-run MERT (on MT02 only) with the augmented phrase table in order to get performance gains. Table 8 reports

[4]On the dev set, about 20K (among 210K) abbreviation translation entries are matched in the Chinese side.

the MERT weights with different phrase tables. One may notice the change of the weight in *word penalty* feature. This is very intuitive in order to prevent the hypothesis being too long due to the expansion of the abbreviations into their full-forms.

Feature	Baseline	AAMT
language model	0.137	0.133
phrase translation	0.066	0.023
lexical translation	0.061	0.078
reverse phrase translation	0.059	0.103
reverse lexical translation	0.112	0.090
phrase penalty	-0.150	-0.162
word penalty	**-0.327**	**-0.356**
distortion model	0.089	0.055

Table 8: Weights obtained by MERT

5 Related Work

Though automatically extracting the relations between full-form Chinese phrases and their abbreviations is an interesting and important task for many natural language processing applications (e.g., machine translation, question answering, information retrieval, and so on), not much work is available in the literature. Recently, Chang and Lai (2004), Chang and Teng (2006), and Lee (2005) have investigated this task. Specifically, Chang and Lai (2004) describes a hidden markov model (HMM) to model the relationship between a full-form phrase and its abbreviation, by treating the abbreviation as the *observation* and the full-form words as *states* in the model. Using a set of manually-created *full-abbreviation* relations as training data, they report experimental results on a *recognition* task (i.e., given an abbreviation, the task is to obtain its full-form, or the vice versa). Clearly, their method is *supervised* because it requires the *full-abbreviation* relations as training data.[5] Chang and Teng (2006) extends the work in Chang and Lai (2004) to automatically extract the relations between full-form phrases and their abbreviations. However, they have only considered relations between single-word phrases and single-character abbreviations. Moreover, the HMM model is computationally-expensive and unable to exploit the *data co-occurrence* phenomena that we have exploited efficiently in this paper. Lee (2005) gives a summary about how Chinese abbreviations are formed and presents many examples. Manual rules are created to expand an abbreviation to its full-form, however, no quantitative results are reported.

None of the above work has addressed the Chinese abbreviation issue in the context of a machine translation task, which is the primary goal in this paper. To the best of our knowledge, our work is the first to systematically model Chinese abbreviation expansion to improve machine translation.

The idea of using a bridge (i.e., full-form) to obtain translation entries for unseen words (i.e., abbreviation) is similar to the idea of using paraphrases in MT (see Callison-Burch et al. (2006) and references therein) as both are trying to introduce *generalization* into MT. At last, the goal that we aim to exploit monolingual corpora to help MT is in-spirit similar to the goal of using non-parallel corpora to help MT as aimed in a large amount of work (see Munteanu and Marcu (2006) and references therein).

6 Conclusions

In this paper, we present a novel method that automatically extracts relations between full-form phrases and their abbreviations from monolingual corpora, and induces translation entries for these abbreviations by using their full-form as a bridge. Our method is *scalable* enough to handle large amount of monolingual data, and is essentially *unsupervised* as it does not require any additional annotated data than the baseline translation system. Our method exploits the *data co-occurrence* phenomena that is very useful for relation extractions. We integrate our method into a state-of-the-art phrase-based baseline translation system, i.e., Moses (Koehn et al., 2007), and show that the integrated system consistently improves the performance of the baseline system on various NIST machine translation test sets.

Acknowledgments

We would like to thank Yi Su, Sanjeev Khudanpur, Philip Resnik, Smaranda Muresan, Chris Dyer and the anonymous reviewers for their helpful comments. This work was partially supported by the Defense Advanced Research Projects Agency's GALE program via Contract No. HR0011-06-2-0001.

[5] However, the HMM model aligns the characters in the abbreviation to the words in the full-form in an *unsupervised* way.

References

Chris Callison-Burch, Philipp Koehn, and Miles Osborne, 2006. Improved Statistical Machine Translation Using Paraphrases. *In Proceedings of NAACL 2006*, pages 17-24.

Marine Carpuat and Dekai Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. *In Proceedings of EMNLP 2007*, pages 61-72.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. *In Proceedings of ACL 2007*, pages 33-40.

Jing-Shin Chang and Yu-Tso Lai. 2004. A preliminary study on probabilistic models for Chinese abbreviations. *In Proceedings of the 3rd SIGHAN Workshop on Chinese Language Processing*, pages 9-16.

Jing-Shin Chang and Wei-Lun Teng. 2006. Mining Atomic Chinese Abbreviation Pairs: A Probabilistic Model for Single Character Word Recovery. *In Proceedings of the 5rd SIGHAN Workshop on Chinese Language Processing*, pages 17-24.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201-228.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. *In Proceedings of COLING/ACL 2006*, pages 961-968.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan,Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constrantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *In Proceedings of ACL*, Demonstration Session, pages 177-180.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. *In Proceedings of NAACL 2003*, pages 48-54.

H.W.D Lee. 2005. A study of automatic expansion of Chinese abbreviations. MA Thesis, The University of Hong Kong.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting Parallel Sub-Sentential Fragments from Non-Parallel Corpora. *In Proceedings of ACL 2006*, pages 81-88.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. *In Proceedings of ACL 2003*, pages 160-167.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. *In Proceedings of ACL 2000*, pages 440-447.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417-449.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *In Proceedings of ACL 2002*, pages 311-318.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing*, pages 901-904.

Z.P. Yin. 1999. Methodologies and principles of Chinese abbreviation formation. *In Language Teaching and Study*, 2:73-82.