Using Sublexical Translations to Handle the OOV Problem in MT

Chung-chi Huang
ISA
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
u901571@gmail.com

Ho-ching Yen Shih-ting Huang Jason S. Chang
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
{fi26.tw,koromiko1104,jason.jschang}@gmail.com

Abstract

We introduce a method for learning to translate out-of-vocabulary (OOV) words. The method focuses on combining sublexical/constituent translations of an OOV to generate its translation candidates. In our approach, wildcard searches are formulated based on our OOV analysis, aimed at maximizing the probability of retrieving OOVs' sublexical translations from existing resource of machine translation (MT) systems. At run-time, translation candidates of the unknown words are generated from their suitable sublexical translations and ranked based on monolingual and bilingual information. We have incorporated the OOV model into a state-of-the-art MT system and experimental results show that our model indeed helps to ease the negative impact of OOVs on translation quality, especially for sentences containing more OOVs (significant improvement).

1 Introduction

Many sentences are submitted to machine translate (MT) systems every day, and an increasing number of such translation services are available between various source and target language pairs. For example, both Google Translate[1] and Windows Live Translator[2] can promptly translate a block of text or a web page.

Due to the creativity and diversity in natural languages, not all source words are known to MT systems specifically their translation model (i.e., phrase table or syntax-based translation rules), in which case most of the current systems treat them as out-of-vocabulary (OOV) words and leave them untranslated. However, leaving unknown OOVs untouched in the output translation may degrade the overall translation quality since the lexical choices and reordering around the OOVs may be negatively impacted. The problem of OOV could be better handled if a model recognized and translated the constituents of an OOV word.

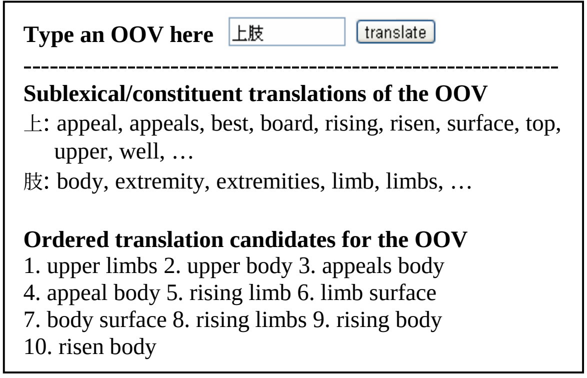
In general, the causes of unknown words can be mainly categorized into the following. OOVs result from segmentation error in the source language (e.g., 滲血症 is erroneously split into two words 滲 and 血症 which leads to an OOV 血症). Another source of OOVs can be attributed to name entity such as person, location and organization. Finally, OOVs may originate from low-frequency abbreviations (e.g., 體協 athletic association) and combination forms (e.g., eyebank, widebody) of common words (e.g., bank, body). In this paper, we focus on handling OOVs of the last type, abbreviation and combination, which, according to our OOV analysis in Section 4, accounts for one fourth of OOVs.

Consider the Chinese sentence "王燕 的 上肢 肌力 恢復 了 兩級" (the muscle strength of wang yan's upper limbs has regained by two levels). If MT systems do not cover "上肢" in the translation model, typically it, an OOV, will be sent out untouched to the output. Better result might be obtained by first finding translations for the OOV's

[1] http://translate.google.com
[2] http://www.microsofttranslator.com

constituents such as “upper” (for the 上 part) and “limbs” (for the 肢 part), and combining these sublexical translations (“upper” and “limbs”) to yield the reference translation: “upper limbs”. We can find these sublexical translations via wildcard queries, “上*” and “*肢” where * stands for any Chinese character. Intuitively, by extracting and combining the translations of OOVs’ sublexical constituents (i.e., Chinese ideographs), we could correctly translate the unknown.

We present a model that automatically retrieves translations of OOVs’ constituents from existing bilingual resource in a MT system that, if combined, are expected to translate OOVs. An example sublexical translation process for the unknown “上肢” is shown in Figure 1. Wildcard search may find related sublexical translations, for example, “*appeal, rise, upper, and surface*” for query “上*” and “*body, extremity, and limbs*” for query “*肢”, but some are not appropriate for the unknown. Our model constrains the choices of the sublexical translations and removes unlikely ones by analyzing a collection of monolingual and bilingual lexical databases. We describe the process in more detail in Section 3.



Type an OOV here 上肢

Sublexical/constituent translations of the OOV

上: appeal, appeals, best, board, rising, risen, surface, top, upper, well, …

肢: body, extremity, extremities, limb, limbs, …

Ordered translation candidates for the OOV

1. upper limbs 2. upper body 3. appeals body
4. appeal body 5. rising limb 6. limb surface
7. body surface 8. rising limbs 9. rising body
10. risen body

Figure 1. An example translation candidate list for the OOV “上肢”.

At run-time, for an OOV in a source sentence, our model retrieves a limited number of translations for its constituents and generates an ordered list of its translation candidates based on bilingual lexical correspondences and monolingual fluency. The ordered candidate list returned by the proposed model can provide translation choices for human translators directly, or can be incorporated into MT decoders to ease the negative impact of OOVs on translation quality.

2 Related Work

Recently, translating OOV words in the field of machine translation has received much attention. In this paper, we address one aspect of translating OOVs by combining translations of OOVs’ constituents retrieved using wildcard queries and MT system’s existing resources.

While this paper focuses on translating the source words with no translation equivalents in the bilingual resources of MT systems via sublexical translations, interesting approaches were presented to generate additional bitexts from comparable, but not parallel, bilingual texts (Fung and Cheung, 2004; Munteanu and Marcu, 2005). Adding more parallel data to MT systems, though may not be always available for some language pairs, tends to reduce the number of OOVs. On the other hand, some work began to extract translations for unknown words from external knowledge sources such as dictionaries and the Web. Unknown words were replaced by their definitions or translations in dictionaries (Vilar et al., 2007; Eck et al., 2008), or by translations mined from large-scale web data (Nagata et al., 2000; Cao and Li, 2002). In our method, translations reside within MT systems’ training corpus not from external knowledge.

Recent work has been done on translating different OOV cases: name entities (NE), compounds, and morphological variants. Knight and Graehl (1997) introduced a transliteration model to tackle proper names while Hassan and Sorensen (2005) presented a NE translating approach that combines NE translation and transliteration in a single framework. On the other hand, Cao and Li (2002) and Tanaka and Baldwin (2003) focused on translating compound words, especially noun phrases, via statistical approach and translation templates. Furthermore, in languages (e.g., Arabic) where morphological variants are a major cause of OOVs, much work was described to transform these variants into in-vocabulary word forms (Koehn and Knight, 2003; Yang and Kirchhoff, 2006; Arora et al., 2008). In contrast, we focus on translating OOVs resulting from abbreviations of source phrases or combination forms of common words. These two cover some portion of name entities and noun-noun and adjective-noun compounds (e.g., 邊貿 border trade (NN), and 新規 new regulations (AN)).

In the studies more closely related to our work, Marton et al. (2009) proposed a paraphrase model that replaces OOVs with in-vocabulary equivalents. Paraphrases were learnt based on word alignments computed over a large additional set of bitexts. And Mirkin et al. (2009) paraphrased OOV words via entailment rules derived from monolingual corpora and manually compiled synonym thesaurus. These studies are similar in spirit to our work. However, we do not address the problem via paraphrasing. In our model, the wildcard translation searches of OOVs' constituents might retrieve their source-language paraphrases with translations. Instead of using these source paraphrases, we directly use the target translations.

Recently, Li and Yarowsky (2008) presented an unsupervised method for extracting the mappings between full-form phrases and their abbreviations that are OOVs. The main difference from our work is that, in their approach, they need a reverse MT system and they focus on solving OOVs of name entities. Our approach generates translations for OOVs of abbreviations and combinations, which cover common words (e.g., 成名 became famous, 邊貿 border trade and 新規 new regulations) as well as name entities. In addition, our wildcard searches for translations of OOVs' constituents from existing resources of MT systems can be viewed as finding the translations of their full-form words.

3 The OOV Model

Submitting sentences with OOV words to MT systems does not work very well. They typically generate the corresponding target-language translations by matching exact words or phrases in their translation model. Unfortunately, OOVs have no matches and MT systems would ignore or directly copy them to the output. To translate an OOV, a promising approach is to automatically transform the exact-match lookup into a set of wildcard searches that are expected to find the common sublexcial translations of the OOV.

3.1 Problem Statement

We focus on finding translations of an OOV word from existing bilingual resource (e.g., MT phrase table) via constituent or sublexical lookups. These translation candidates are ranked and returned as the output of the model. The returned candidates can be examined by human translators directly, or passed on to MT decoders (e.g., Moses) to ease the impact of OOVs. Sublexical wildcard searches tend to lead to a lot of noise. Thus, it is crucial that sublexical translations be constrained to confident ones. At the same time, the set of the OOV's translation candidates cannot be so large that it overwhelms the users or the subsequent (typically computationally expensive) decoders. Therefore, our goal is to return a reasonable-sized set of translation candidates that contain suitable translations of the OOV word. We now formally state the problem that we are addressing.

Problem Statement: We are given a database of translation equivalents *TE* (e.g., MT phrase table) trained on a parallel corpus *C* (e.g., Hong Kong Parallel Text), large-scale monolingual corpus *CT* (e.g., English Gigaword), and an out-of-vocabulary word *O*. Our goal is to generate a ranked list of translation candidates that are likely to provide suitable translations for *O*. For this, we identify the constituents o_1, …, o_m of *O*, retrieve, and evaluate the translations of o_i from *TE* by partial matching o_i. The retrieved translations of o_i, if combined, are likely to translate *O*.

In the rest of this section, we describe our solution to this problem. First, we define a strategy for finding sublexical translations, translations of a constituent of an OOV word (Section 3.2). This strategy relies on reformulated wildcard searches in replace of the original exact-match derived from the OOV analysis on the development data (more details in Section 4.2). Then, we show how our model assembles sublexical translations of the OOV and ranks the assembled candidates at run-time according to bilingual and monolingual consideration (Section 3.3).

3.2 Finding Sublexical Translations

For a given OOV, we attempt to find relevant translations of its constituents. Our sublexical translating process is shown in Figure 2.

(1) Retrieve possible translations for a constituent of an OOV from translation equivalent *TE*
(2) Extract sublexical translations and restrain translation choices
(3) Prune less probable sublexical translations
(4) Output translation candidates for the constituent

Figure 2. Outline used to find sublexical translations.

Retrieving Translations. In the first stage of the translating process, we retrieve translations for a constituent of an OOV from a bank of translation equivalents *TE*. We transform the traditional exact-match search for an OOV's translations into a sequence of wildcard sublexical lookups for constituents' translations. By doing so, the OOV may be decomposed and translated. Figure 3 shows the algorithm for retrieving translations for a constituent of an OOV.

In Step (1) of the algorithm we initiate *TransCol* to collect possible translations of a constituent *c* of an OOV *O*. Based on the constituent *c* of *O* and its position in *O*, we formulate wildcard queries for *c*'s translations (Step (2)). We will describe how to formulate search queries according to position in Section 4. In Step (3), for each *query*, we retrieve translations from *TE* and append them to *TransCol*.

```
procedure RetrieveTranslations($c$,$O$,$TE$)
(1) $TransCol$ = $\phi$
(2) $Queries$ = formulateQuery($c$,position($c$, $O$))
    for each $query$ in $Queries$
(3)     $TransCol$ += { findTranslations($query$,$TE$) }
    Return $TransCol$
```

Figure 3. Retrieving possible sublexical translations.

Take the constituent "上" and "肢" of the OOV "上肢" for example. The wildcard queries "上*" and "*肢" generated by the algorithm yield the sets of translation pairs {<"上訴","appeal for">, <"上升","increasing of">,…,<"上段","upper block">} and {<"四肢","extremities">, <"四肢","four limbs">,…,<"義肢", "prosthesis">}, respectively.

Extracting and Restraining Translations. In the second stage, we extract the sublexical translations based on redundancy, and constrain the translation words to frequent ones in view of composing proper translations for the OOV.

The input to this stage is the possible translations of a constituent obtained from the previous stage, represented by <*source word*, *target phrase*> pairs. The output of this stage is a set of <*source word*, *target N-gram*> pairs, in which *target N-gram* cover different surface forms of the same lemma (e.g., "limb" and "limbs").

The method for extracting and selecting frequent sublexical translations involves generating target-language N-grams in *target phrase*, constraining the choices of target words by consulting a target-language lexicon, and filtering out infrequent words. Each step is discussed more detailed below.

For each <*source word*, *target phrase*> pair, we first generate target N-grams from the *target phrase*. Take <"四肢", "four limbs"> for instance. Target N-grams include "four", "limbs", and "four limbs". Second, content words (e.g., nouns and verbs) in the N-grams are constrained to ones seen in a lexical database (e.g., WordNet). Obviously, if a word is unseen in a lexicon, it is probably not a good translation. Third, we prune infrequent N-grams. To compare fairly, the occurrence count is accumulated over inflected word forms sharing the same lemma. For the purpose, we use a lemmatizer (Bird et al., 2008) in our implementation. Above method would yield commonly-seen sublexcial translations in the form of <*source word*, *target N-gram*>. Notice that target N-grams would cover many inflectional forms for the constituent, which is generally beneficial to the subsequent sentence translation task.

Pruning Less Probable Translations. In the third and final stage (Step (3) in Figure 2), we prune less probable sublexical translations of OOV based on bilingual associations. In the previous step, to extract the translations of the constituent *c*, each <*source word*, *target phrase*> pair is transformed to inflectional N-grams, <*source word*, *target N-gram*> pairs. Some N-grams, however, are less related to *c*. To achieve better computation efficiency and translation accuracy, we remove less probable sublexical N-gram translations before combining an OOV's sublexical translations.

First, we exploit a bilingual dictionary (e.g., bilingual WordNet) to build reference bilingual associations. In the following, we describe two approaches for building reference associations.

- *All-constituent approach:* For each entry <*source phrase*, *target phrase*> in the dictionary, we build bilingual associations between all constituents in the *source phrase* and all N-grams in the *target phrase*. Once a source constituent co-occurs with a target N-gram, an association between them is built.
- *Salient-constituent approach:* Associations are only registered between the salient constituent of the *source phrase* and all N-grams of the *target phrase* for each dictionary entry <*source phrase*, *target phrase*>. We define that a constituent of a *source phrase* is a salient constituent if it is most associable to

the *target phrase*. Formally speaking, the salient constituent *c** for the <*source phrase*, *target phrase*> is chosen satisfying

$$\arg\max_{c} Dice\left(c, target\ phrase\right) =$$

$$\arg\max_{c} \frac{2 \cdot Count(c, target\ phrase)}{Count(c) + Count(target\ phrase)}$$

where c denotes a constituent in *source phrase* and $Count(\cdot)$ the frequency in the dictionary. Note that the set of associations generated by this approach is a subset of that generated by all-constituent.

Once the bilingual associations are constructed, we prune <*source word*, *target N-gram*> pair of a query constituent *c* if (*c*, *target N-gram*) is unseen in the reference associations.

Notice that different approaches (all-constituent and salient-constituent approach) can be leveraged for different applications since all-constituent aims at high recall and salient-constituent high precision. In our implementation, we first refer to all-constituent associations to prune and maintain high recall rate. If there are still too many candidates, we then turn to the salient associations to prune more aggressively. After pruning, the output of this stage is a set of translation pairs expected to have strong constituent-translation associations. Also note that the reference bilingual associations in this stage could be modeled as soft constraint and the frequency of associations registered could be used as a feature for run-time candidate ranking.

```
procedure EvaluateCandidates(O, TE, C, CT)
      for each constituent c in OOV O
(1a)   SubTrans=RetrieveSublexTrans(c, O, TE)
(1b)   CandList[position(c, O)]=BilingualInfo(SubTrans,c,C)
(2a)Straight=CandList[1]
(2b)Inverted=CandList[|O|] // |O| denotes the length of O
      for each constituent position cp>1 in ascending
                                    constituent positions of O
(3a)    Straight ⊗ =CandList[cp]
      for each constituent position cp<|O| in descending
                                    constituent positions of O
(3b)    Inverted ⊗ = CandList[cp]
(4a)Straight=MonolingualInfo(Straight, CT)
(4b)Inverted=MonolingualInfo(Inverted, CT)
     Candidates = Straight + Inverted
(5)  RankedCandidates=Sort Candidates in decreasing
                                              order of P
(6)  Return top N RankedCandidates with P exceeding θ
```

Figure 4. Candidate generating and ranking at run-time.

3.3 Run-Time Candidate Ranking

Once the sublexical translations of an OOV are found, our model then generates and ranks translation candidates for the OOV using the procedure in Figure 4.

For each constituent *c* of the given OOV *O*, we first retrieve its probable sublexical translations *SubTrans*, including inflected forms, from *TE* using the method described in Section 3.2 (Step (1a)). *SubTrans* is a list of <*source word*, *target N-gram*> pairs, where *source word* contains a constituent of *O*. Then, we use bidirectional conditional probability to measure the association strength between the OOV's constituent *c* and its translation in *SubTrans*, and record such information at corresponding position (Step (1b)). Following the format in Step (1a), elements in *CandList* are of the form (*c*, <*source word*, *target N-gram*>, $P_{\text{sub}}(target\ N\text{-}gram|c) \times P_{\text{sub}}(c|target\ N\text{-}gram)$). Two-way conditional probability $P_{\text{sub}}(target\ N\text{-}gram|c)$ and $P_{\text{sub}}(c|target\ N\text{-}gram)$ are trained on bitexts *C* where the unit of token in the source language is constituent not word.

Once we acquire translations of each constituent of the given OOV, we are ready to generate the OOV's translation candidates. Although the translation scope of an OOV is much smaller than that of a whole sentence, re-ordering of an OOV's sublexical translations can still happen (e.g., "air adjustment" for "調氣" where "air" is aligned to "氣" and "adjustment" to "調"). For this, both straight and inverted candidates are generated during sublexical translation combination.

In Step (2), we initialize *Straight* and *Inverted* to collect the OOV's translation candidates by composing its sublexical translations in straight and inverted order. During candidate generation, *Straight* and *Inverted* iteratively cover more span of the OOV (Step (3)), collecting constituent translations and multiplying sublexical translation probabilities at the same time. For each assembled translation candidate *tc*, its translation probability (P_{trans}) is estimated by the product of two-way conditional probabilities of the sublexical translation pairs as:

$$\sqrt[|O|]{\prod_{c_i \in O} P_{\text{sub}}\left(c_i \middle| target\ N\text{-}gram_{i,j}\right) \times P_{\text{sub}}\left(target\ N\text{-}gram_{i,j} \middle| c_i\right)}$$

where c_i denotes a constituent of *O* and *target N-gram*$_{i,j}$ a sublexical translation for c_i composing *tc*.

Apart from bilingual information, we further leverage monolingual information to prune and estimate assembled translation candidates. In Step (4), we first prune less probable word combinations in the target language, and, for those which survive the pruning, further incorporate target language model probabilities P_{TLM} into *Straight*/*Inverted*. For pruning, we calculate MI value, regarded as a good measurement for the possibilities of word combinations, of each bigram w_1 and w_2 in an assembled candidate using $\text{MI}(w_1, w_2) = \log_2\left(\Pr(w_1, w_2)/\left(\Pr(w_1)\Pr(w_2)\right)\right)$

After merging straight and inverted cases, in Step (5) we rank translation candidate, *tc*, based on bilingual translation probabilities and target language model: $P(tc) = P_{\text{trans}}(tc)^{\lambda_1} \times P_{\text{TLM}}(tc)^{\lambda_2}$ where λ_i is the feature weight and $\sum \lambda_i$ equals to one. Target language model plays an important role in ranking translation candidates especially when straight and inverted candidates are all taken into account, leading to the same translation probability (P_{trans}). In that case, P_{TLM} helps to differentiate the fluency of the composed translations.

Finally, the N top-ranked candidates whose probabilities (P) exceed a threshold θ are returned as the likely translations of the given OOV. Notice that θ will be tuned for better translation quality. An example translation of an OOV “上肢” (upper limbs) is shown in Figure 1.

4 Experimental Setting

The OOV handling model was designed to find translations of OOV words from existing bilingual resources that are likely to help human translators or MT systems ease the negative impact of OOVs. As such, the model will be trained and evaluated over translation task on top of an existing MT system. More specifically, we incorporate the OOV model into an existing Chinese-to-English MT system and carry out the evaluation process.

4.1 Underlying SMT System and Data Sets

The proposed OOV model focused on translating OOV words by exploiting existing bilingual resource (e.g., phrase table) in a MT system. Therefore, our model was built on top of a statistical MT system which accepts translation suggestions of our OOV model. We used the state-of-the-art phrase-based MT system, Moses (Koehn et al., 2007), as our underlying decoder. It provides simple XML markup for plugging in external knowledge without changing any component such as translation or language model. In this paper, we leveraged the markup language to incorporate OOVs’ translation candidates.

To train Moses’ translation model, we used Hong Kong Parallel Text (LDC2004T08) and Xinhua News Agency (LDC2007T09). Chinese sentences were word segmented by the CKIP Chinese word segmenter (Ma and Chen, 2003). Common settings were used to run Moses: GIZA++ (Och and Ney, 2003) was used for word alignment, grow-diagonal-final for bidirectional word alignment combination, and phrase extraction heuristics in (Koehn et al., 2003) for bilingual phrase pairs. We exploited English Gigaword Third Edition (LDC2007T07) and SRILM toolkit (Stolcke, 2002) to build trigram language model.

In our OOV model, on the other hand, we leveraged WordNet 3.0 and bilingual WordNet (Huang et al., 2004) to filter sublexical translations (Section 3.2). To prune less probable translation candidates of OOVs at run-time (Section 3.3), we used Web 1T 5-gram First Edition (LDC2006T13) for MI calculation. As for the run-time candidate ranking (Section 3.3), we exploited the same parallel corpora and target-language corpus used for Moses to estimate bilingual translation probabilities (P_{trans}) and target-language fluency (P_{TLM}), respectively.

4.2 Query Formats and Bilingual Resource

Length	Number of OOVs	Percentage (%)
1	56	4.4
2	683	53.7
3	352	27.7
4	225	9
5	25	2
6+	42	3.3

Table 1. The number of OOVs w.r.t. OOVs’ lengths.

To study the problem of translating OOV words, we used NIST MT-08 test set consisting of 1,273 unknown words in 637 sentences out of a total of 1,357 sentences. Among these 1,273 distinct OOV words, we first inspected the number of OOVs with respect to their lengths, i.e., the number of characters (Table 1). One could see from Table 1 that OOVs of two characters accounted for more than half of the OOV cases. As a result, we focused on translating two-character unknown words. To further analyze the portion of OOV types, formulate appropriate query in replace of the exact-match search, and determine good bilingual resource for sublexical translation retrieval, we randomly selected 100 sentences containing only two-character OOVs from MT-08 set. OOVs in these 100 sentences were classified into 10 types shown in Table 2 according to their reference translations manually extracted from reference sentences. Since our model aimed to retrieve and combine translations of an OOV's constituents, it targets specifically at translating OOV words of the *Combination Forms* in Table 2.

Type	Description of the type	Example	Freq

Name Entity	Name entities could be transliterated	布希(bush)
		膠州(jiaozhou)
			12

Segmentation Error	Words erroneously split by the segmentation system	領式 (開領式) 會兒(這會兒)	16
Order Variants	Character sequence within is reversed without changing the original meaning	療治 (治療) (treat)	1
Writing Variants	Simplified vs. traditional Chinese characters	念書 (唸書) (study)	1
Informal	Words used in conversation or informal writing	看頭 (worth watching)	6
Domain Specific	Domain specific terminologies	勤務 (service support)	2
Old Use	Words rarely in use now	古稀 (60 years old)	8
Rare Paraphrase	Words could be translated by paraphrases	訪談 (interview)	25
Word + Suffix	Words composed by a content character (underscored) and a not translatable function character	忙著 (busy) 爐子 (stove)	4
Combination Form	Words could be translated by combining sublexical translations	上肢 (upper limbs) 肌力 (muscle strength)	25

Table 2. OOV types and their descriptions and examples.

Query Form	# translatable OOVs	Example	
		OOV	Matched
c_1* and c_2*	17	上肢(upper limbs)	上方 肢體
c_1* and *c_2	9	上肢(upper limbs)	上方 四肢
c_1 and c_2	2	震魔(quake demon)	地震 魔鬼
*c_1 and *c_2	1	鍾體(bell body)	時鐘 身體

Table 3. Query formulation with matched examples.

Intuitively, there are four ways to formulate the query for sublexical translations for a two-character OOV c_1, c_2. Table 3 shows that the first and second query formulation of adding wildcard * can retrieve most relevant translations. Therefore, our model adopted these two query forms.

Additionally, to determine the effectiveness of various bilingual resources for finding sublexical translations, we compared translation hit rates of OOVs of different resources based on above two query forms. Among Lin Yutang's dictionary (http://humanum.arts.cuhk.edu.hk/Lexis/Lindict/), LDC translation lexicon (LDC2002L27), and character-based and word-based phrase table, hit rates of *Combination Form* OOVs were 0.64, 0.68, 0.60, and 0.88, respectively. Word-based phrase table resulted in the highest hit rate, thus chosen as our bilingual resource for sublexical translation retrieval. Apart from its high hit rate, there are other advantages in using word-based phrase table: it includes different inflectional word forms and is more domain-relevant to the NIST MT test set.

4.3 Parameter Tuning

In this subsection, we describe the pilot experiment with the development data set of 50 sentences randomly selected from NIST MT-08 set. Each sentence contained at least one OOV. We used the data to fine-tune the two parameters in our system: the maximal number of translation candidates returned by the OOV model *N* and the filtering threshold θ used to prune improbable translation candidates at run-time, thus differentiating OOVs of *Combination Form* from those that are not.

The size of the returned translation suggestions for OOVs could not be so large that overwhelm subsequent users or decoders. Therefore, our goal was to return reasonable-size translation candidates for an OOV word with its correct translations ranked higher. To choose a suitable *N*, we exploited the sentences with *Combination Form* OOVs in the developing data and evaluated the performance of translating these OOVs using *hit rate* and *Mean Reciprocal Rank* (*MRR*). Here, *MRR* was defined as a measure of how much effort needed for a user to locate the first correct translation for the given OOV word in the ranked candidate list. Table 4 summarized the *hit rates* and *MRRs* at different values of *N*. We eventually set *N* to 10 considering coverage, *MRR*, and time complexity of decoding.

N	*hit rate*	*MRR*
5	8/25	0.27
10	11/25	0.28
20	12/25	0.28
40	12/25	0.28

Table 4. Hit rates and *MRR*s at different candidate sizes.

On the other hand, a threshold θ on the probability of translation candidate was used for pruning and determining the applicability of our model on OOV word (Some OOVs are not suitable for the solution in this paper). To select a suitable θ, we compared translation results at different levels of filtering thresholds on the developing data (Figure 5). As indicated, when the threshold was larger than -7, very few translation candidates were considered, leading to not much difference from the underlying Moses. On the other hand, when lower than -13, more noisy translations were incorporated, resulting in a decline in translation quality. We chose -12, the best performing threshold (probably achieved balanced performance between translations' precision and coverage), as our threshold to prune less probable candidates or to activate our OOV model.

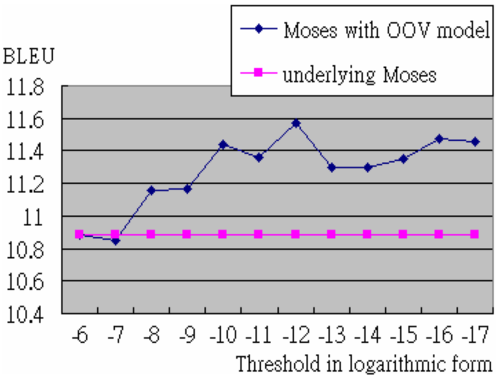


Figure 5. BLEU scores of different filtering thresholds.

5 Evaluation Results and Discussion

We report the experimental results in this section. First, we report the translation performance of the underlying MT system, Moses, with and without our OOV model using BLEU (Papineni et al., 2002). We then show example translations of some OOVs generated by our system and point out the future improvement of our OOV model.

5.1 Experimental Results

During the evaluation, we used NIST MT-06 test set, containing 1,664 sentences, for testing. In this test set, there were 933 distinct unknown words scattered in 859 sentences, and its number of OOVs with respect to OOVs' length was much alike to that of the developing set. In the experiment, out of the 933 OOV words, our model generated translation candidates for the 170 distinct two-character OOV words in 351 sentences. Note that the parameter θ in the OOV model determines its applicability, i.e., whether a combination-form translation is acceptable or not (for subsequent Moses). The produced combination-form translation candidates were incorporated into Moses using XML markup.

Table 5 shows the overall performance of the underlying Moses and the CST system (Moses with combined sublexical translations). Although the difference in BLEU score between them is not very significant, the improvement in brevity penalty (BP) is noticeable. That is, the CST system generated sentences closer to the reference translations in length. A slightly better BLEU score implies the additional words provided by our model achieved better or at least similar translation accuracy compared to the underlying Moses system.

System	BLEU	BP	# words
Moses	21.46	0.928	41052
CST	21.56	0.939	41707

Table 5. Performance of two systems (# sentence=1664).

System	BLEU	BP	# words
Moses	17.41	0.912	10833
CST	**17.83**	**0.951**	11583

Table 6. Performance of two systems (# sentence=351).

If we look at the performance of the 351 sentences in the test set for which our OOV model provided translation candidates, the CST system significantly outperformed Moses in BLEU and, encouragingly, improved the BP relatively by 4.4% (Table 6). The significance test was performed using bootstrap resampling in (Koehn, 2004).

The experimental results show that we were able to translate some portion of the OOV words without degrading the performance of an existing MT system and the translation quality of the sentences was substantially improved with our automatic translation suggestions for OOV words.

5.2 Example Translations

In this subsection, we examine example English translations for OOV words provided by our model and we point out future direction for our model.

Table 7(a) shows four examples in which the reference translations of the OOVs were ranked high by our OOV model (bold-faced) and the underlying decoder chose the correct translations for the OOVs. One may find that the combined words in the OOVs' translations belong to different part-of-speech (POS) sequences: “korean war” (AN), “border trade” (NN), “became famous” (VA) and “new regulations” (AN). This indicates wildcard search for sublexical translations of our model could handle combination-form OOVs of various POS combinations. In addition, as suggested by Example 4, accurate OOV translation indeed has an impact on lexical choice of OOV’s surrounding words or even on the overall fluency.

Example sentence 1	… 雙方 曾 在 一九五 0 年 至 一九五三 年 的 韓戰 時期 相互 敵對。
OOV (reference)	韓戰 (korean war)
Sorted translations	**korean war,** korea war, koreans war, korean armistice, korean military, …
Moses	… the two sides had been in years during 1950s to 一九五三 years 韓戰 period mutual hostility .
CST	… the two sides had been in years during 1950s to 一九五三 years of the **korean war** period mutual hostility .
Example sentence 2	... 中國 雲南 與 緬甸 的 貿易 (含 邊貿) 總 額 達 6.3 億 美元 …
OOV (reference)	邊貿 (border trade)
Sorted translations	**border trade**, bilateral trade, borderline trade, …, border trading, …
Moses	… china 's yunnan province and myanmar trade (including 邊貿) amounting to 630 million u.s. dollars …
CST	… china 's yunnan province and myanmar trade (including **border trade**)a total of 630 million u.s. dollars …
Example sentence 3	…突然 間 成名 多金 的 美麗 演員
OOV (reference)	成名(find fame, became famous)
Sorted translations	**became famous**, become famous, becoming famous, becomes famous, …
Moses	…suddenly 成名 , 多金 beautiful actors
CST	…suddenly **became famous**, 多金 beautiful actors
Example sentence 4	中國 電子 銀行業務 管理 新規 將 於 3月 1 日 起 施行
OOV (reference)	新規 (rules, regulations, new regulations)
Sorted translations	planning new, provides new, **new regulations**, new rules, …, new provisions, …
Moses	china 's electronic 銀行業務 新規 management will be held on march 1 date for the implementation of the
CST	china 's electronic 銀行業務 management of the **new regulations** will come into effect on march 1

Table 7. (a) Examples with OOVs correctly translated. Sorted translation candidates of OOVs and translations of the source sentences by Moses and CST are shown.

Additionally, Table 7(b) displays three example sentences where translations of OOV words partially match their reference. In Example 6, although our OOV model ranked the correct translation “three cars” at the top choice, Moses chose “three trucks” for the OOV probably based on local consideration of the target language model. Moreover, BLEU may underestimate CST’s performance: for Example 7, “salary payment”, though does not match the reference, may be an acceptable translation.

Example sentence 5	南韓 女子 競速 滑冰 選手 李寶拉 …
OOV (reference)	競速 (speed)
Sorted translations	speed competition, accelerating competition, …, speed races, …
Moses	south korean woman 競速 李寶拉 skating team …
CST	south korean woman **speed races** 李寶拉 skating team …
Example sentence 6	…58 公里 處 發生 三車 相撞 事故 , 一 輛 轎 車 、 一 貨車 和 一 輛 油罐車 撞 在一起 …
OOV (reference)	三車 (three-vehicle, three car)
Sorted translations	three cars, three vehicles, motor third, jeep three, three trucks, triple car, …
Moses	… 58 kilometers in 三車 collision incident , a car , a truck and a truck oil tanker collided together …
CST	… 58 kilometers in a collision of **three trucks** , cars and a truck and a truck oil tanker collided together …
Example sentence 7	… 根據 法律 規定 專門 制定 了 報刊 轉載 付酬的 標準 。
OOV (reference)	付酬 (payment , remuneration)
Sorted translations	remuneration paid, …, salary payment, paying salaries, pay salaries, pays salaries, …
Moses	… under the law specifically enacted which touched 付酬 standards .
CST	…under the law specifically enacted which touched on **salary payment** standards .

Table 7. (b) Examples with OOVs partially translated.

Examples 8-9 in Table 7(c) illustrate that there is room for future improvements of our system. In Example 8, despite the fact that our model found the correct sublexical translations “snow storm” of the OOV “風雪”, it reference translation is in the form of *one-word* compound “snowstorm”. Therefore, we would like to accommodate such cases by adding such compounds into our candidate lists. Furthermore, in the last example in Table 7(c), we observed the need for sense disambiguation of constituents (i.e., Chinese character). In this case, the character “班” in OOV “減班” may be associated with many senses, like “班級” (class), “航班” (flight), “值班” (shift), and “車班” (schedule), resulting in many possible choices of translations. Since it is difficult to disambiguate such constituent without the help of contextual information of the OOV word, we plan to incorporate OOVs’ contexts (other than OOVs themselves) into our module as features for better OOV translation quality.

Example sentence 8	…我們 終於 看到 風雪 已經 停 了 。
OOV (reference)	風雪 (snowstorm, blizzard, snow)
Sorted translations	snow storm, snow storms, ice storm, …
Moses	… we can finally see 風雪 has stopped .
CST	… we can finally see **snow storms** have stopped .
Example sentence 9	火車 的 班次 也 是 減班 。
OOV (reference)	減班 (reduce schedule, reduce frequency)
Sorted translations	reducing class, cutting class, reduced flights, cut flights, reduced class, reduce flights, reduce class, cutting flights, …
Moses	the class size , train frequency is 減班 .
CST	the class size , train frequency is **cutting class** .

Table 7. (c) Examples for future improvements.

6 Summary

We have introduced a method for generating translation suggestions for OOV words from a MT system's existing bilingual resource via sublexical translations. The promising experimental result indicates that as a preprocessing step before a state-of-the-art phrase-based decoder, Moses, our OOV model genuinely provides good translations for unknown words and that out-of-vocabulary words are in fact in-vocabulary on the sublexical level for languages such as Chinese. Apart from the future improvements we point out in Section 5.2 for our model, we would also like to incorporate paraphrasing techniques such as (Marton et al., 2009) and (Mirkin et al., 2009) for better OOV translation quality and coverage.

References

Karunesh Arora, Michael Paul, and Eiichiro Sumita. 2008. Translation of unknown words in phrase-based statistical machine translation for languages of rich morphology. In *Proc. of SLTU*.

Yunbo Cao and Hang Li. 2002. Base noun phrase translation using web data and the EM algorithm. In *Proc. of COLING*.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2008. Communicating unknown words in machine translation. In *Proc. of LREC*.

Pascale N. Fung and Percy Cheung. 2004. Mining ver-non-parallal corpora: parallel sentence and lexicon extraction via bootstrapping and EM. In *Proc. of EMNLP*.

Hany Hassan and Jeffrey Sorensen. 2005. An integrated approach for Arabic-English named entity translation. In *Proc. of the ACL Workshop on Computational Approaches to Semitic Languages*.

Chu-Ren Huang, Ru-Yng Chang, and Shiang-Bin Lee. 2004. Sinica BOW (Bilingual Ontological Wordnet): integration of bilingual WordNet and SUMO. In *Proc. of LREC*.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proc. of EACL*.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proc. of EACL*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL/HLT*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

Philipp Koehn, 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 *Proceedings of the ACL 2007 Demo and Poster*.

Zhifei Li and David Yarowsky. 2008. Unsupervised translation induction for Chinese abbreviations using monolingual corpora. In *Proc. of ACL*.

Wei-Yun Ma and Keh-Jiann Chen. 2003. Introduction to CKIP Chinese word segmentation system for the first international Chinese word segmentation bakeoff. In *Proc. of the ACL Workshop on Chinese Language Processing*.

Dragos S. Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4).

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proc. of EMNLP*.

Shachar Mirkin, Lucia Specia, Nicola Cancedda, Ido Dagan, Marc Dymetman, and Idan Szpektor. 2009. Source-language entailment modeling for translating unknown terms. In *Proc. of ACL/IJCNLP*.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Kishore Papineni, Salim Roukos, ToddWard, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP*.

David Vilar, Jan-T. Peter, and Hermann Ney. 2007. Can we translate letters?. In *Proc. of ACL workshop on Statistical Machine Translation*.

Mei Yang and Katrin Kirchhoff. 2006. Phrase-based backoff models for machine translation of highly inflected languages. In *Proc. of EACL*.