A Term Translation System Using Hierarchical Phrases and Morphemes

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

[†]Computer Science, The University of Tokyo
[‡]School of Computer Science, University of Manchester
National Centre for Text Mining (NaCTeM)
{wxc, tsujii}@is.s.u-tokyo.ac.jp

1 Introduction

This paper describes a cascaded translation system for translating technical terms. Our motivation is to 1) translate new technical terms and out-of-vocabulary (OOV) words, and 2) generalize the translation ability of existing bilingual lexicons. Our cascaded translation system makes use of hierarchical phrases to capture the reordering patterns. In addition, the system includes a transliteration model which takes English syllable and Japanese katakana as the basic translation units. For a randomly selected test set, our system achieved an accuracy of 94.7%. For the OOV terms, our system achieved an accuracy of 21.3%. Furthermore, we testified our system is competitive to a state-of-the-art online translation system when translating Chinese sentences into English.

2 System Framework

2.1 Training

Figure 1 shows the training framework of our system. Similar with traditional phrase-based translation systems (Koehn et al., 2007), the training process includes three steps, *preprocessing*, *rule extracting*, and *tuning*. Starting from original parallel corpora, we lexical analyze and normalize the bilingual terms. For example, we segment the Japanese terms into word sequences and tokenize the English terms by English punctuation such as ',' and '.'. The English letters are set into lowercase as well.

During rule extracting, we train three kinds of translation rules: a flat phrase translation table, a hierarchical phrase translation table, and a morphological translation table. The former two kinds of tables are extracted from phrase-aligned parallel corpora, where the word alignments are obtained by applying GIZA++ (Och and Ney, 2003) and *grow-diag-final-and* balancing strategy (Koehn et al., 2007) on the tokenized parallel corpora. We use Moses (Koehn et al., 2007) to generate the flat phrase translation table. The extracting approach described in (Chiang, 2005) is re-implemented by us to generate the hierarchical phrase translation table. Note that we keep these two kinds of tables to be *open*. That is, we allow additional bilingual lexicons to be added to the flat translation table and additional manual hierarchical rules to be appended to the hierarchical translation table. The bidirectional translation probabilities and bidirectional lexical weights of these additional rules are greedily set to be 1.

The third translation table is in morpheme and syllable level. We make use of Morfessor[1] (Creutz and Lagus, 2007), an unsupervised language-independent morphological analyzer, to generate the morpheme sequences for English. As for Japanese, we use the character or katakana sequences. We again use Moses to generate a flat morpheme phrase based translation table.

In addition, we take the English word and Japanese katakana pairs extracted from the tokenized parallel corpora as the parallel corpus for training a transliteration model. This time, we use the following heuristic rules described in (Jiang et al., 2007) to split a English word into syllables:

- a, e, i, o, u are defined as vowels. y is defined as a vowel when it is not followed by a vowel. All other letters are defined as consonants;
- Duplicate the nasals m and n whenever they

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

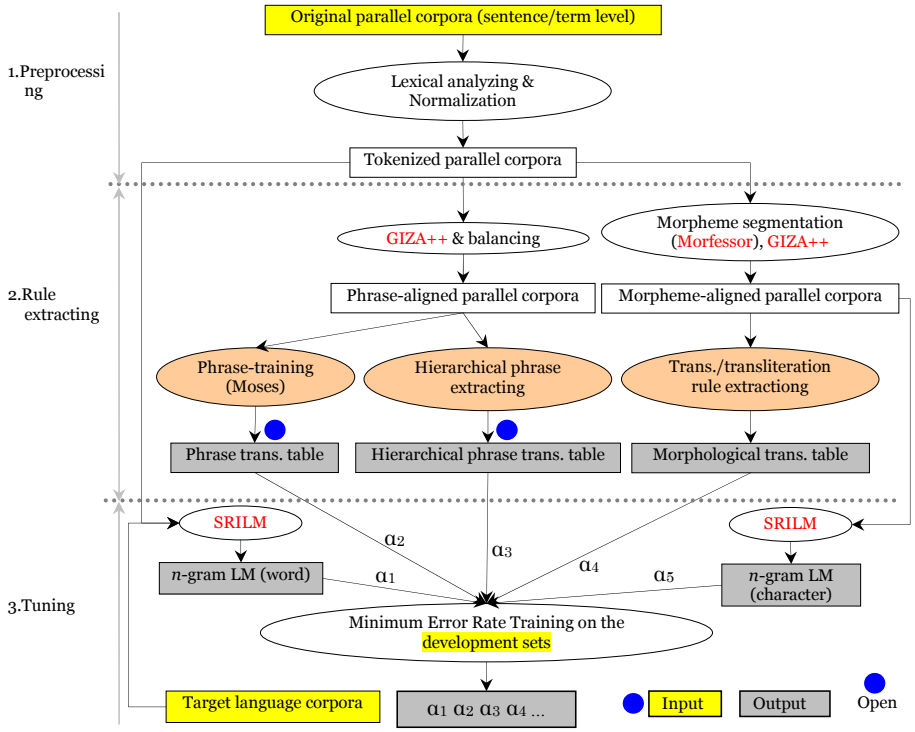

Figure 1: System training framework.

are surrounded by vowels. When they appear behind a vowel, they will be combined with the vowel to form a new vowel;

- Consecutive consonants are separated, and consecutive vowels are treated as a single vowel;

- A consonant and the subsequent vowel are treated as a syllable. Each isolated vowel or consonant is regarded as an individual syllable.

For the examples shown in Figure 2, *trihexyphenidyl* is split into *t ri he xy p hen ni dy l*; emphtonography is split into *ton no g ra p hy*; *cysteinyl* is split into *cy s tein ny l*; *leukotriene* is split into *leu ko t rien ne*, and *receptor* is split into *re ce p to r*. From Figure 2, we observe that one English syllable possibly align to one or two Japanese katakana. Thus, we again make use of GIZA++ to automatically generate the alignments between English syllable and Japanese katakana. Consequently, a syllable sequence to katakana sequence translation table is extracted.

Finally, we tune the weights of the translation features by use minimum error rate training (Och, 2003) on additional development sets. The features include the bidirectional translation probabilities and the bidirectional lexical weights of the three kinds of translation tables, number of words and number of phrases in the final translation string, n-gram word language model score, n-gram character language model score, number of hierarchical rules used, and number of morpheme rules used. Since the three kinds of tables are extracted in different ways, we further make use of *derivation-level combination* (Liu et al., 2009) for mixing different types of translation rules within one derivation.

2.2 Decoding

Figure 3 shows the basic idea of our decoding algorithm. The major idea is to translation a given term in a cascaded way: from hierarchical phrase level to flat phrase level and then to morpheme/syllable levels. Following (Chiang, 2007), we use the +LM CKY decoding algorithm and cube-pruning to generate the n-best lists.

At the first step, we retrieve the flat phrase translation table to seek the translations for each n-gram in the source terms. When meeting an OOV word,

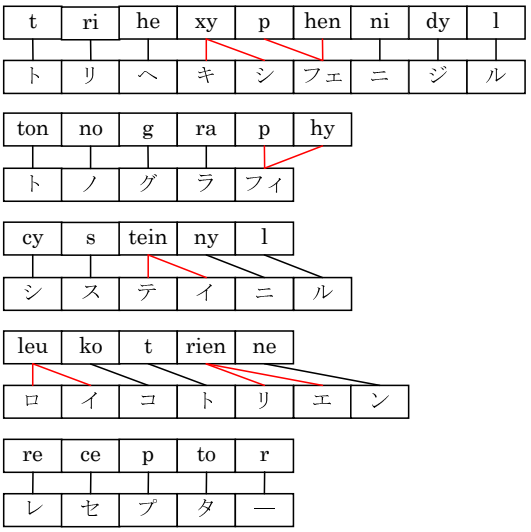

Figure 2: Examples of transliteration alignments between English syllable and Japanese katakana.

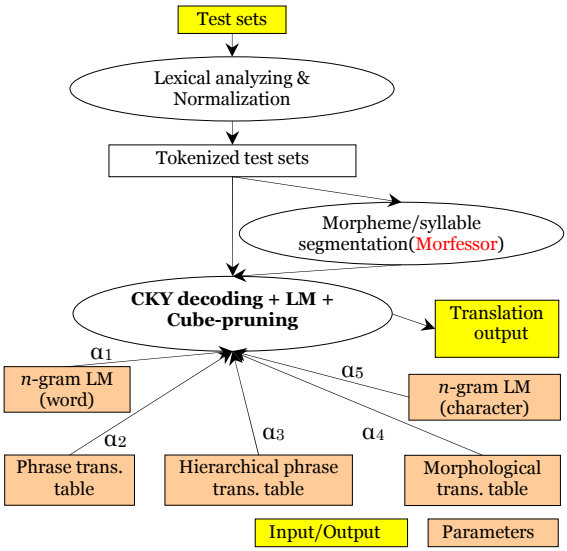

Figure 3: Decoding algorithm.

	# lines	# EN words	# JP words
03 En2Jp	149,583	249,895	415,635
09 Jp2En	150,517	251,311	418,629

Table 1: Statistics of the LSD dictionary.

	BLEU-2	WER	PER	EMR
Random	0.9488	0.0344	0.0298	0.947
OOV	0.4197	0.5970	0.5562	0.213

Table 2: Term level English-to-Japanese translation result.

we retrieve the morphological translation table and generate a translation candidate list for it. Then, we make use of hierarchical rules to combine the existing translation candidates together. The word order during combination is guided by the hierarchical rules. In case of no hierarchical rules available, we use the glue rules used in (Chiang, 2007) to combine the translation candidates in a left-to-right monotonic way.

3 Experiments

We testify our translation system on two translation tasks, English-to-Japanese term translation and Chinese-to-English sentence translation.

3.1 Term translation

We use the Life Science Dictionary (LSD)[2] as our corpus for English-to-Japanese term translation. Table 1 shows the statistics of the dictionary. In the English side, the terms contain averagely 1.67 words and the words contain averagely 8.8 letters. In the Japanese side, the terms contain averagely 2.78 words and the words contain averagely 2.1 Japanese characters/katakana. We perform two experiments which use different kinds of development and test sets. In the first experiment, we randomly selected 2,000 bilingual terms as the development set and another 2,000 as the test set. The remaining bilingual terms is taken as the training data. In the second experiment, we only select terms that contain words that appear only once in the LSD dictionary. We selected 1,000 entries as the development set and another 1,000 entries as the test set. The remaining bilingual terms is taken as the training data. For each experiment, we train a 2-gram word model and a 2-gram character model on the Japanese terms in the training data. We set the beam size to 10 during cube-pruning.

Table 2 shows the translation result. We evaluate the final translations by four metrics, BLEU-2 (Papineni et al., 2002), word error rate (WER), position-independent word error rate (PER), and exact matching rate (EMR). For the randomly selected test set, our system achieved an accuracy of 94.7% which is very impressive. For the OOV terms, our system

[2]http://lsd.pharm.kyoto-u.ac.jp/ja/index.html

NIST-C2E	Google	Ours (Decoding time)
2003	0.3439	0.3760** (0.404)
2004	0.3423	0.3526** (0.283)
2005	0.3494	0.3507 (0.320)
2006	0.3284	0.3314 (0.260)
2008	0.2813**	0.2604 (0.216)

Table 3: Sentence level Chinese-to-English translation results. The tests were performed on 16:24 21th Jan 2011. ** = $p < 0.01$.

achieved an accuracy of 21.3%.

3.2 Sentence translation

Besides term level translation, we further testified our translation system in sentence level. We use the parallel data available for the NIST 2008 constrained track of Chinese-to-English translation task as the training data, which contains 5.1M parallel sentences, 128M Chinese words and 147M English words after pre-processing. We take the NIST 2003 test set as the only development set and NIST test sets from 2004 to 2008 as our test sets. We trained a 5-gram LM on the Xinhua portion of LDC English gigaword corpus version 3 (LDC2007T07). The beam size of 20 is used during cube-pruning. For simplicity, we only make use of flat/hierarchical phrases for translation.

Table 3 shows the translation results and the decoding time (second per sentence) of our system. As a baseline system for comparison, we use a state-of-the-art online translation system, Google[3]. For the NIST 2003, 2004 test sets, our system is significantly better ($p < 0.01$) than the baseline. Our system is competitive and slightly better than the baseline in terms of NIST 2005 and 2006 test sets. However, the baseline is significantly better ($p < 0.01$) than our system on the NIST 2008 test set. Through these comparisons, we can draw the conclusion that our system is competitive to the baseline system in terms of Chinese-to-English translation.

4 Conclusion

We have described the training and decoding processes of a cascaded translation system for translating technical terms. Our cascaded translation system makes use of hierarchical phrases to capture the reordering problem. In addition, the system includes a transliteration model which can take English syllable and Japanese katakana as the basic translation units. For a randomly selected test set, our system achieved an accuracy of 94.7%. For the OOV terms, our system achieved an accuracy of 21.3%. Furthermore, our system is competitive to a state-of-the-art online translation system for Chinese-to-English sentence translation.

[3]http://translate.google.co.jp/

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), and Microsoft Research Asia Machine Translation Theme.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*.

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

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1).

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named entity translation with web mining and transliteration. In *Proceedings of IJCAI 2007*, pages 1629–1634, Hyderabad, India.

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 *Proc. of ACL Demo*.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009. Joint decoding with multiple translation models. In *Proc. of ACL-IJCNLP*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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

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