Learning Bilingual Linguistic Reordering Model for Statistical Machine Translation

Han-Bin Chen, Jian-Cheng Wu and Jason S. Chang
Department of Computer Science
National Tsing Hua University
101, Guangfu Road, Hsinchu, Taiwan
{hanbin,d928322,jschang}@cs.nthu.edu.tw

Abstract

In this paper, we propose a method for learning reordering model for BTG-based statistical machine translation (SMT). The model focuses on linguistic features from bilingual phrases. Our method involves extracting reordering examples as well as features such as part-of-speech and word class from aligned parallel sentences. The features are classified with special considerations of phrase lengths. We then use these features to train the maximum entropy (ME) reordering model. With the model, we performed Chinese-to-English translation tasks. Experimental results show that our bilingual linguistic model outperforms the state-of-the-art phrase-based and BTG-based SMT systems by improvements of 2.41 and 1.31 BLEU points respectively.

1 Introduction

Bracketing Transduction Grammar (BTG) is a special case of Synchronous Context Free Grammar (SCFG), with binary branching rules that are either straight or inverted. BTG is widely adopted in SMT systems, because of its good trade-off between efficiency and expressiveness (Wu, 1996). In BTG, the ratio of legal alignments and all possible alignment in a translation pair drops drastically especially for long sentences, yet it still covers most of the syntactic diversities between two languages.

It is common to utilize phrase translation in BTG systems. For example in (Xiong et al., 2006), source sentences are segmented into phrases. Each sequences of consecutive phrases, mapping to cells in a CKY matrix, are then translated through a bilingual phrase table and scored as implemented in (Koehn et al., 2005; Chiang, 2005). In other words, their system shares the same phrase table with standard phrase-based SMT systems.

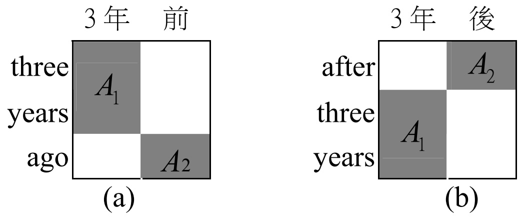

Figure 1: Two reordering examples, with straight rule applied in (a), and inverted rule in (b).

On the other hand, there are various proposed BTG reordering models to predict correct orientations between neighboring blocks (bilingual phrases). In Figure 1, for example, the role of reordering model is to predict correct orientations of neighboring blocks A_1 and A_2. In flat model (Wu, 1996; Zens et al., 2004; Kumar and Byrne, 2005), reordering probabilities are assigned uniformly during decoding, and can be tuned depending on different language pairs. It is clear, however, that this kind of model would suffer when the dominant rule is wrongly applied.

Predicting orientations in BTG depending on context information can be achieved with lexical features. For example, Xiong et al. (2006) proposed MEBTG, based on maximum entropy (ME) classification with words as features. In MEBTG, first words of blocks are considered as the features, which are then used to train a ME model

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 254–262,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

for predicting orientations of neighboring blocks. Xiong et al. (2008b) proposed a linguistically annotated BTG (LABTG), in which linguistic features such as POS and syntactic labels from source-side parse trees are used. Both MEBTG and LABTG achieved significant improvements over phrase-based Pharaoh (Koehn, 2004) and Moses (Koehn et al., 2007) respectively, on Chinese-to-English translation tasks.

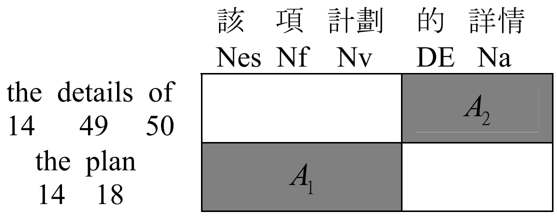

Figure 2: An inversion reordering example, with POS below source words, and class numbers below target words.

However, current BTG-based reordering methods have been limited by the features used. Information might not be sufficient or representative, if only the first (or tail) words are used as features. For example, in Figure 2, consider target first-word features extracted from an inverted reordering example (Xiong et al., 2006) in MEBTG, in which first words on two blocks are both "the". This kind of feature set is too common and not representative enough to predict the correct orientation. Intuitively, one solution is to extend the feature set by considering both boundary words, forming a more complete boundary description. However, this method is still based on lexicalized features, which causes data sparseness problem and fails to generalize. In Figure 2, for example, the orientation should basically be the same, when the source/target words "計畫/plan" from block A_1 is replaced by other similar nouns and translations (e.g. "plans", "events" or "meetings"). However, such features would be treated as unseen by the current ME model, since the training data can not possibly cover all such similar cases.

In this paper we present an improved reordering model based on BTG, with bilingual linguistic features from neighboring blocks. To avoid data sparseness problem, both source and target words are classified; we perform part-of-speech (POS) tagging on source language, and word classification on target one, as shown in Figure 2. Additionally, features are extracted and classified depending on lengths of blocks in order to obtain a more informed model.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the model used in our BTG-based SMT systems. Section 4 formally describes our bilingual linguistic reordering model. Section 5 and Section 6 explain the implementation of our systems. We show the experimental results in Section 7 and make the conclusion in Section 8.

2 Related Work

In statistical machine translation, reordering model is concerned with predicting correct orders of target language sentence given a source language one and translation pairs. For example, in phrase-based SMT systems (Koehn et al., 2003; Koehn, 2004), distortion model is used, in which reordering probabilities depend on relative positions of target side phrases between adjacent blocks. However, distortion model can not model long-distance reordering, due to the lack of context information, thus is difficult to predict correct orders under different circumstances. Therefore, while phrase-based SMT moves from words to phrases as the basic unit of translation, implying effective local reordering within phrases, it suffers when determining phrase reordering, especially when phrases are longer than three words (Koehn et al., 2003).

There have been much effort made to improve reordering model in SMT. For example, researchers have been studying CKY parsing over the last decade, which considers translations and orientations of two neighboring block according to grammar rules or context information. In hierarchical phrase-based systems (Chiang, 2005), for example, SCFG rules are automatically learned from aligned bilingual corpus, and are applied in CKY style decoding.

As an another application of CKY parsing technique is BTG-based SMT. Xiong et al. (2006) and Xiong et al. (2008a) developed MEBTG systems, in which first or tail words from reordering examples are used as features to train ME-based reordering models.

Similarly, Zhang et al. (2007) proposed a model similar to BTG, which uses first/tail words of phrases, and syntactic labels (e.g. NP and VP) from source parse trees as features. In their work, however, inverted rules are allowed to apply only when source phrases are syntactic; for non-syntactic ones, blocks are combined straight with a constant score.

More recently, Xiong et al. (2008b) proposed LABTG, which incorporates linguistic knowledge by adding features such as syntactic labels and POS from source trees to improve their MEBTG. Different from Zhang's work, their model do not restrict non-syntactic phrases, and applies inverted rules on any pair of neighboring blocks.

Although POS information is used in LABTG and Zhang's work, their models are syntax-oriented, since they focus on syntactic labels. Boundary POS is considered in LABTG only when source phrases are not syntactic phrases.

In contrast to the previous works, we present a reordering model for BTG that uses bilingual information including class-level features of POS and word classes. Moreover, our model is dedicated to boundary features and considers different combinations of phrase lengths, rather than only first/tail words. In addition, current state-of-the-art Chinese parsers, including the one used in LABTG (Xiong et al., 2005), lag beyond in inaccuracy, compared with English parsers (Klein and Manning, 2003; Petrov and Klein 2007). In our work, we only use more reliable information such as Chinese word segmentation and POS tagging (Ma and Chen, 2003).

3 The Model

Following Wu (1996) and Xiong et al. (2006), we implement BTG-based SMT as our system, in which three rules are applied during decoding:

$$A \rightarrow [A_1 \quad A_2] \tag{1}$$

$$A \rightarrow \langle A_1 \quad A_2 \rangle \tag{2}$$

$$A \rightarrow x / y \tag{3}$$

where A_1 and A_2 are blocks in source order. Straight rule (1) and inverted rule (2) are reordering rules. They are applied for predicting target-side order when combining two blocks, and form the reordering model with the distributions

$$\mathrm{P}_{\mathrm{reo}}(A_1, A_2, order)^{\lambda_{reo}}$$

where $order \in \{straight,\ inverted\}$.

In MEBTG, a ME reordering model is trained using features extracted from reordering examples of aligned parallel corpus. First words on neighboring blocks are used as features. In reordering example (a), for example, the feature set is

{"S1L=three", "S2L=ago", "T1L=3", "T2L=前"}

where "S1" and "T1" denote source and target phrases from the block A_1.

Rule (3) is lexical translation rule, which translates source phrase x into target phrase y. We use the same feature functions as typical phrase-based SMT systems (Koehn et al., 2005):

$$\mathrm{P}_{\mathrm{trans}}(x \mid y) = p(x \mid y)^{\lambda_1} \cdot p(y \mid x)^{\lambda_2} \cdot p_{lw}(x \mid y)^{\lambda_3} \\ \cdot p_{lw}(y \mid x)^{\lambda_4} \cdot \mathrm{e}^{\lambda_5} \cdot \mathrm{e}^{|y| \lambda_6}$$

where $p_{lw}(x \mid y)^{\lambda_3} \cdot p_{lw}(y \mid x)^{\lambda_4}$, e^{λ_5} and $\mathrm{e}^{|y| \lambda_6}$ are lexical translation probabilities in both directions, phrase penalty and word penalty.

During decoding, the blocks are produced by applying either one of two reordering rules on two smaller blocks, or applying lexical rule (3) on some source phrase. Therefore, the score of a block A is defined as

$$\mathrm{P}(A) = \mathrm{P}(A_1) \cdot \mathrm{P}(A_2) \\ \cdot \Delta\mathrm{P}_{\mathrm{lm}}(A_1, A_2)^{\lambda lm} \cdot \mathrm{P}_{\mathrm{reo}}(A_1, A_2, order)^{\lambda reo}$$

or

$$\mathrm{P}(A) = \mathrm{P}_{\mathrm{lm}}(A)^{\lambda lm} \cdot \mathrm{P}_{\mathrm{trans}}(x \mid y)$$

where $\mathrm{P}_{\mathrm{lm}}(A)^{\lambda lm}$ and $\Delta\mathrm{P}_{\mathrm{lm}}(A_1, A_2)^{\lambda lm}$ are respectively the usual and incremental score of language model.

To tune all lambda weights above, we perform minimum error rate training (Och, 2003) on the development set described in Section 7.

Let B be the set of all blocks with source side sentence C. Then the best translation of C is the target side of the block $\overline{A}$, where

$$\overline{\mathrm{A}} = \underset{A \in B}{\mathrm{argmax}}\ \mathrm{P}(A)$$

4 Bilingual Linguistic Model

In this section, we formally describe the problem we want to address and the proposed method.

4.1 Problem Statement

We focus on extracting features representative of the two neighboring blocks being considered for reordering by the decoder, as described in Section 3. We define $S(A)$ and $T(A)$ as the information on source and target side of a block A. For two neighboring blocks A_1 and A_2, the set of features extracted from information of them is denoted as feature set function $F(S(A_1), S(A_2), T(A_1), S(A_2))$. In Figure 1 (b), for example, $S(A_1)$ and $T(A_1)$ are simply the both sides sentences "3 年" and "three years", and $F(S(A_1), S(A_2), T(A_1), S(A_2))$ is

{"S1L=three", "S2L=after", "T1L=3", "T2L=後"}

where "S1L" denotes the first source word on the block A_1, and "T2L" denotes the first target word on the block A_2.

Given the adjacent blocks A_1 and A_2, our goal includes (1) adding more linguistic and representative information to A_1 and A_2 and (2) finding a feature set function F' based on added linguistic information in order to train a more linguistically motivated and effective model.

4.2 Word Classification

As described in Section 1, designing a more complete feature set causes data sparseness problem, if we use lexical features. One natural solution is using POS and word class features.

In our model, we perform Chinese POS tagging on source language. In Xiong et al. (2008b) and Zhang et al. (2007), Chinese parsers with Penn Chinese Treebank (Xue et al., 2005) style are used to derive source parse trees, from which source-side features such as POS are extracted. However, due to the relatively low accuracy of current Chinese parsers compared with English ones, we instead use CKIP Chinese word segmentation system (Ma and Chen, 2003) in order to derive Chinese tags with high accuracy. Moreover, compared with the Treebank Chinese tagset, the CKIP tagset provides more fine-grained tags, including many tags with semantic information (e.g., Nc for place nouns, Nd for time nouns), and verb transitivity and subcategorization (e.g., VA for intransitive verbs, VC for transitive verbs, VK for verbs that take a clause as object).

On the other hand, using the POS features in combination with the lexical features in target language will cause another sparseness problem in the phrase table, since one source phrase would map to multiple target ones with different POS sequences.

As an alternative, we use mkcls toolkit (Och, 1999), which uses maximum-likelihood principle to perform classification on target side. After classification, the toolkit produces a many-to-one mapping between English tokens and class numbers. Therefore, there is no ambiguity of word class in target phrases and word class features can be used independently to avoid data sparseness problem and the phrase table remains unchanged.

As mentioned in Section 1, features based on words are not representative enough in some cases, and tend to cause sparseness problem. By classifying words we are able to linguistically generalize the features, and hence predict the rules more robustly. In Figure 2, for example, the target words are converted to corresponding classes, and form the more complete boundary feature set

{"T1L=14", "T1R=18", "T2L=14", "T2R=50"} (4)

In the feature set (4), #14 is the class containing "the", #18 is the class containing "plans", and #50 is the class containing "of." Note that we add last-word features "T1R=18" and "T2R=50". As mentioned in Section 1, the word "plan" from block A_1 is replaceable with similar nouns. This extends to other nominal word classes to realize the general rule of inverting "the ... NOUN" and "the ... of".

It is hard to achieve this kind of generality using only lexicalized feature. With word classification, we gather feature sets with similar concepts from the training data. Table 1 shows the word classes can be used effectively to cope with data sparseness. For example, the feature set (4) occurs 309 times in our training data, and only 2 of them are straight, with the remaining 307 inverted examples, implying that similar features based on word classes lead to similar orientation. Additional examples of similar feature sets with different word classes are shown in Table 1.

class X	T1R = X	straight/inverted
9	graph, government	2/488
18	plans, events	2/307
20	bikes, motors	0/694
48	day, month, year	4/510

Table 1: List of feature sets in the form of {"T1L=14", "T1R=X", "T2L=14", "T2R=50"}.

4.3 Feature with Length Consideration

Boundary features using both the first and last words provide more detailed descriptions of neighboring blocks. However, we should take the special case blocks with length 1 into consideration. For example, consider two features sets from straight and inverted reordering examples (a) and (b) in Figure 3. There are two identical source features in both feature set, since first words on block A_1 and last words on block A_2 are the same:

{"S1L=P","S2R=Na"} $\subseteq F(S(A_1),S(A_2),T(A_1), S(A_2))$

Therefore, without distinguishing the special case, the features would represent quite different cases with the same feature, possibly leading to failure to predict orientations of two blocks.

We propose a method to alleviate the problem of features with considerations of lengths of two adjacent phrases by classifying both the both source and target phrase pairs into one of four classes: M, L, R and B, corresponding to different combinations of phrase lengths.

Suppose we are given two neighboring blocks A_1 and A_2, with source phrases P_1 and P_2 respectively. Then the feature set from source side is classified into one of the classes as follows. We give examples of feature set for each class according to Figure 4.

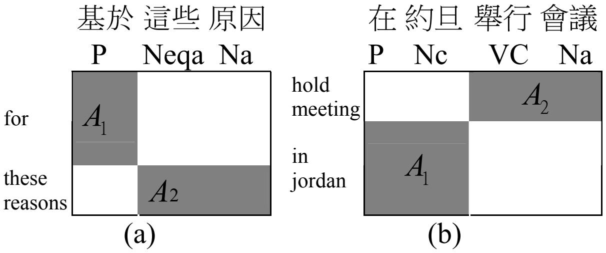

Figure 3: Two reordering examples with ambiguous features on source side.

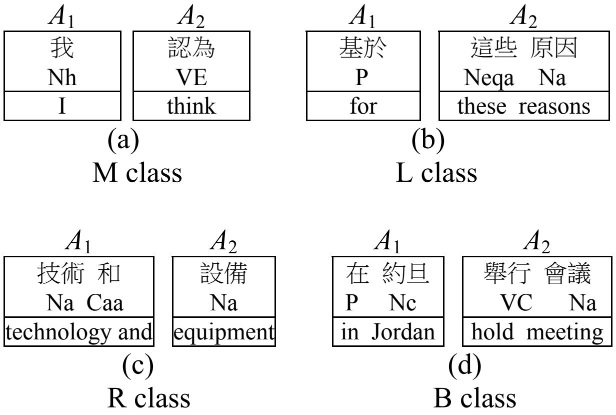

Figure 4: Examples of different length combinations, mapping to four classes.

1. M class. The lengths of P_1 and P_2 are both 1. In Figure 4 (a), for example, the feature set is

   {"M1=Nh", "M2=VE"}

2. L class. The length of P_1 is 1, and the length of P_2 is greater than 1. In Figure 4 (b), for example, the feature set is

   {"L1=P", "L2=Neqa", "L3=Na"}

3. R class. The length of P_1 is greater than 1, and the length of P_2 is 1. In Figure 4 (c), for example, the feature set is

   {"R1=Na", "R2=Caa", "R3=Na"}

4. B class. The lengths of P_1 and P_2 are both greater than 1. In Figure 4 (d), for example, the feature set is

   {"B1=P", "B2=Nc", "B3=VC", "B4=Na"}

We use the same scheme to classify the two target phrases. Since both source and target words are classified as described in Section 4.2, the feature sets are more representative and tend to lead to consistent prediction of orientation. Additionally, the length-based features are easy to fit into memory, in contrast to lexical features in MEBTG.

To summarize, we extract features based on word lengths, target-language word classes, and fine-grained, semantic oriented parts of speech. To illustrate, we use the neighboring blocks from Figure 2 to show an example of complete bilingual linguistic feature set:

{"S.B1=Nes", "S.B2=Nv", "S.B3=DE", "S.B4=Na", "T.B1=14", "T.B2=18", "T.B3=14", "T.B4=50"}

where "S." and "T." denote source and target sides.

In the next section, we describe the process of preparing the feature data and training an ME model. In Section 7, we perform evaluations of this ME-based reordering model against standard phrase-based SMT and previous work based on ME and BTG.

5 Training

In order to train the translation and reordering model, we first set up Moses SMT system (Koehn et al., 2007). We obtain aligned parallel sentences and the phrase table after the training of Moses, which includes running GIZA++ (Och and Ney, 2003), grow-diagonal-final symmetrization and phrase extraction (Koehn et al., 2005). Our system shares the same translation model with Moses, since we directly use the phrase table to apply translation rules (3).

On the other side, we use the aligned parallel sentences to train our reordering model, which includes classifying words, extracting bilingual phrase samples with orientation information, and training an ME model for predicting orientation.

To perform word classification, the source sentences are tagged and segmented before the Moses training. As for target side, we ran the Moses scripts to classify target language words using the mkcls toolkit before running GIZA++. Therefore, we directly use its classification result, which generate 50 classes with 2 optimization runs on the target sentences.

To extract the reordering examples, we choose sentence pairs with top 50% alignment scores provided by GIZA++, in order to fit into memory. Then the extraction is performed on these aligned sentence pairs, together with POS tags and word classes, using basically the algorithm presented in Xiong et al. (2006). However, we enumerate all reordering examples, rather than only extract the smallest straight and largest inverted examples. Finally, we use the toolkit by Zhang (2004) to train the ME model with extracted reordering examples.

6 Decoding

We develop a bottom-up CKY style decoder in our system, similar to Chiang (2005). For a Chinese sentence *C*, the decoder finds its best translation on the block with entire *C* on source side. The decoder first applies translation rules (3) on cells in a CKY matrix. Each cell denotes a sequence of source phrases, and contains all of the blocks with possible translations. The longest length of source phrase to be applied translations rules is restricted to 7 words, in accordance with the default settings of Moses training scripts.

To reduce the search space, we apply threshold pruning and histogram pruning, in which the block scoring worse than 10^{-2} times the best block in the same cell or scoring worse than top 40 highest scores would be pruned. These pruning techniques are common in SMT systems. We also apply recombination, which distinguish blocks in a cell only by 3 leftmost and rightmost target words, as suggested in (Xiong et al., 2006).

7 Experiments and Results

We perform Chinese-to-English translation task on NIST MT-06 test set, and use Moses and MEBTG as our competitors.

The bilingual training data containing 2.2M sentences pairs from Hong Kong Parallel Text (LDC2004T08) and Xinhua News Agency (LDC2007T09), with length shorter than 60, is used to train the translation and reordering model. The source sentences are tagged and segmented with CKIP Chinese word segmentation system (Ma and Chen, 2003).

About 35M reordering examples are extracted from top 1.1M sentence pairs with higher alignment scores. We generate 171K features for lexicalized model used in MEBTG system, and 1.41K features for our proposed reordering model.

For our language model, we use Xinhua news from English Gigaword Third Edition (LDC2007T07) to build a trigram model with SRILM toolkit (Stolcke, 2002).

Our development set for running minimum error rate training is NIST MT-08 test set, with sentence lengths no more than 20. We report the experimental results on NIST MT-06 test set. Our evaluation metric is BLEU (Papineni et al., 2002) with case-insensitive matching from unigram to four-gram.

System	BLEU-4
Moses(distortion)	22.55
Moses(lexicalized)	23.42
MEBTG	23.65
WC+LC	24.96

Table 2: Performances of various systems.

The overall result of our experiment is shown in Table 2. The lexicalized MEBTG system proposed by Xiong et al. (2006) uses first words on adjacent blocks as lexical features, and outperforms phrase-based Moses with default distortion model and enhanced lexicalized model, by 1.1 and 0.23 BLEU points respectively. This suggests lexicalized Moses and MEBTG with context information outperforms distance-based distortion model. Besides, MEBTG with structure constraints has better global reordering estimation than unstructured Moses, while incorporating their local reordering ability by using phrase tables.

The proposed reordering model trained with word classification (WC) and length consideration (LC) described in Section 4 outperforms MEBTG by 1.31 point. This suggests our proposed model not only reduces the model size by using 1% fewer features than MEBTG, but also improves the translation quality.

We also evaluate the impacts of WC and LC separately and show the results in Table 3-5. Table 3 shows the result of MEBTG with word classified features. While classified MEBTG only improves 0.14 points over original lexicalized one, it drastically reduces the feature size. This implies WC alleviates data sparseness by generalizing the observed features.

Table 4 compares different length considerations, including boundary model demonstrated in Section 4.2, and the proposed LC in Section 4.3. Although boundary model describes features better than using only first words, which we will show later, it suffers from data sparseness with twice feature size of MEBTG. The LC model has the largest feature size but performs best among three systems, suggesting the effectiveness of our LC.

In Table 5 we show the impacts of WC and LC together. Note that all the systems with WC significantly reduce the size of features compared to lexicalized ones.

System	Feature size	BLEU-4
MEBTG	171K	23.65
WC+MEBTG	0.24K	23.79

Table 3: Performances of lexicalized and word classified MEBTG.

System	Feature size	BLEU-4
MEBTG	171K	23.65
Boundary	349K	23.42
LC	780K	23.86

Table 4: Performances of BTG systems with different representativeness.

System	Feature size	BLEU-4
MEBTG	171K	23.65
WC+MEBTG	0.24K	23.79
WC+Bounary	0.48K	24.29
WC+LC	1.41K	24.96

Table 5: Different representativeness with word classification.

While boundary model is worse than first-word MEBTG in Table 4, it outperforms the latter when both are performed WC. We obtain the best result that outperforms the baseline MEBTG by more than 1 point when we apply WC and LC together.

Our experimental results show that we are able to ameliorate the sparseness problem by classifying words, and produce more representative features by considering phrase length. Moreover, they are both important, in that we are unable to outperform our competitors by a large margin unless we combine both WC and LC. In conclusion, while designing more representative features of reordering model in SMT, we have to find solutions to generalize them.

8 Conclusion and Future Works

We have proposed a bilingual linguistic reordering model to improve current BTG-based SMT systems, based on two drawbacks of previously proposed reordering model, which are sparseness and representative problem.

First, to solve the sparseness problem in previously proposed lexicalized model, we perform word classification on both sides.

Secondly, we present a more representative feature extraction method. This involves considering length combinations of adjacent phrases.

The experimental results of Chinese-to-English task show that our model outperforms baseline phrase-based and BTG systems.

We will investigate more linguistic ways to classify words in future work, especially on target language. For example, using word hierarchical structures in WordNet (Fellbaum, 1998) system provides more linguistic and semantic information than statistically-motivated classification tools.

Acknowledgements

This work was supported by National Science Council of Taiwan grant NSC 95-2221-E-007-182-MY3.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL 2005*, pp. 263−270.

Christiane Fellbaum, editor. 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT/NAACL 2003*.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrased-Based Statistical Machine Translation Models. In *Proceedings of AMTA 2004.*

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation*.

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 2007, Demonstration Session*.

Dan Klein and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. In *Proceedings of ACL 2003*.

Shankar Kumar and William Byrne. 2005. *Local phrase reordering models for statistical machine translation*. In *Proceedings of HLT-EMNLP 2005*.

Wei-Yun Ma and Keh-Jiann Chen. 2003. Introduction to CKIP Chinese Word Segmentation System for the First International Chinese Word Segmentation Bakeoff. In *Proceedings of ACL, Second SIGHAN Workshop on Chinese Language Processing*, pp168-171.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *EACL '99: Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 71–76, Bergen, Norway, June.

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

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

Kishore Papineni, 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 ACL*, pages 311–318.

Slav Petrov and Dan Klein. 2007. Improved Inference-for Unlexicalized Parsing. In *Proceedings of HLT-NAACL 2007.*

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Dekai Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In *Proceedings of ACL 1996*.

Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the Penn Chinese treebank with semantic knowledge. In *Proceedings of IJCNLP 2005*, pages 70-81.

Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006*.

Deyi Xiong, Min Zhang, Aiti Aw, Haitao Mi, Qun Liu, and Shouxun Liu. 2008a. Refinements in BTG-based statistical machine translation. In *Proceedings of IJCNLP 2008*, pp. 505-512.

Deyi Xiong, Min Zhang, Ai Ti Aw, and Haizhou Li. 2008b. Linguistically Annotated BTG for Statistical Machine Translation. In *Proceedings of COLING 2008*.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.

R. Zens, H. Ney, T. Watanabe, and E. Sumita. 2004. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *Proceedings of CoLing 2004*, Geneva, Switzerland, pp. 205-211.

Le Zhang. 2004. Maximum Entropy Modeling Toolkit for Python and C++. Available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

Dongdong Zhang, Mu Li, Chi-Ho Li and Ming Zhou. 2007. Phrase Reordering Model Integrating Syntactic Knowledge for SMT. In *Proceedings of EMNLP-CoNLL 2007*.