Learning Translation Boundaries for Phrase-Based Decoding

Deyi Xiong, Min Zhang, Haizhou Li
Human Language Technology
Institute for Infocomm Research
1 Fusionopolis Way, #21-01 Connexis, Singapore 138632.
{dyxiong, mzhang, hli}@i2r.a-star.edu.sg

Abstract

Constrained decoding is of great importance not only for speed but also for translation quality. Previous efforts explore soft syntactic constraints which are based on constituent boundaries deduced from parse trees of the source language. We present a new framework to establish soft constraints based on a more natural alternative: translation boundary rather than constituent boundary. We propose simple classifiers to learn translation boundaries for any source sentences. The classifiers are trained directly on word-aligned corpus without using any additional resources. We report the accuracy of our translation boundary classifiers. We show that using constraints based on translation boundaries predicted by our classifiers achieves significant improvements over the baseline on large-scale Chinese-to-English translation experiments. The new constraints also significantly outperform constituent boundary based syntactic constrains.

1 Introduction

It has been known that phrase-based decoding (phrase segmentation/translation/reordering (Chiang, 2005)) should be constrained to some extent not only for transferring the NP-hard problem (Knight, 1999) into a tractable one in practice but also for improving translation quality. For example, Xiong et al. (2008) find that translation quality can be significantly improved by either prohibiting reorderings around punctuation or restricting reorderings within a 15-word window.

Recently, more linguistically motivated constraints are introduced to improve phrase-based decoding. (Cherry, 2008) and (Marton and Resnik, 2008) introduce syntactic constraints into the standard phrase-based decoding (Koehn et al., 2003) and hierarchical phrase-based decoding (Chiang, 2005) respectively by using a counting feature which accumulates whenever hypotheses violate syntactic boundaries of source-side parse trees. (Xiong et al., 2009) further presents a bracketing model to include thousands of context-sensitive syntactic constraints. All of these approaches achieve their improvements by guiding the phrase-based decoder to prefer translations which respect source-side parse trees.

One major problem with such constituent boundary based constraints is that syntactic structures of the source language do not necessarily reflect translation structures where the source and target language correspond to each other. In this paper, we investigate building classifiers that directly address the problem of *translation boundary*, rather than extracting constituent boundary from source-side parsers built for a different purpose. A translation boundary is a position in the source sequence which begins or ends a *translation zone* [1] spanning multiple source words. In a translation zone, the source phrase is translated as a unit. Reorderings which cross translation zones are not desirable.

Inspired by (Roark and Hollingshead, 2008) which introduces classifiers to decide if a word can begin/end a multi-word constituent, we build two discriminative classifiers to tag each word in the source sequence with a binary class label. The first classifier decides if a word can begin a multi-source-word translation zone; the second classifier decides if a word can end a multi-source-word translation

[1] We will give a formal definition of translation zone in Section 2.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 136–144,
Los Angeles, California, June 2010.

zone. Given a partial translation covering source sequence (i, j) with start word c_i and end word c_j [2], this translation can be penalized if the first classifier decides that the start word c_i can not be a beginning translation boundary or the second classifier decides that the end word c_j can not be an ending translation boundary. In such a way, we can guide the decoder to boost hypotheses that respect translation boundaries and therefore the common translation structure shared by the source and target language, rather than the syntactic structure of the source language.

We report the accuracy of such classifiers by comparing their outputs with "gold" translation boundaries obtained from reference translations on the development set. We integrate translation boundary based constraints into phrase-based decoding and display that they improve translation quality significantly in large-scale experiments. Furthermore, we confirm that they also significantly outperform constituent boundary based syntactic constraints.

2 Beginning and Ending Translation Zones

To better understand the particular task that we address in this paper, we study the distribution of classes of translation boundaries in real-world data. First, we introduce some notations. Given a source sentence $c_1...c_n$, we will say that a word c_i ($1 < i < n$) is in the class B_y if there is a translation zone τ spanning $c_i...c_j$ for some $j > i$; and $c_i \in B_n$ otherwise. Similarly, we will say that a word c_j is in the class E_y if there is a translation zone spanning $c_i...c_j$ for some $j > i$; and $c_j \in E_n$ otherwise.

Here, a translation zone τ is a pair of aligned source phrase and target phrase

$$\tau = (c_i^j, e_p^q)$$

where τ must be consistent with the word alignment M

$$\forall (u, v) \in M, i \leq u \leq j \leftrightarrow p \leq v \leq q$$

By this, we require that no words inside the source phrase c_i^j are aligned to words outside the target phrase e_p^q and that no words outside the source phrase are aligned to words inside the target phrase.

[2] In this paper, we use c to denote the source language and e the target language.

Item	Count (M)	P (%)
Sentences	3.8	–
Words	96.9	–
Words $\in B_y$	22.7	23.4
Words $\in E_y$	41.0	42.3
Words $\notin B_y$ and $\notin E_y$	33.2	34.3

Table 1: Statistics on word classes from our bilingual training data. All numbers are calculated on the source side. P means the percentage.

This means, in other words, that the source phrase c_i^j is mapped as a unit onto the target phrase e_p^q.

When defining the B_y and E_y class, we also require that the source phrase c_i^j in the translation zone must contain multiple words ($j > i$). Our interest is the question of whether a sequence of consecutive source words can be translated as a unit (i.e. whether there is a translation zone covering these source words). For a single-word source phrase, if it can be translated separately, it is always translated as a unit in the context of phrase-based decoding. Therefore this question does not exist.

Note that the first word c_1 and the last word c_n are unambiguous in terms of whether they begin or end a translation zone. The first word c_1 must begin a translation zone spanning the whole source sentence. The last word c_n must end a translation zone spanning the whole source sentence. Therefore, our classifiers only need to predict the other $n-2$ words for a source sentence of length n.

Table 1 shows statistics of word classes from our training data which contain nearly 100M words in approximately 4M sentences. Among these words, only 22.7M words can begin a translation zone which covers multiple source words. 41M words can end a translation zone spanning multiple source words, which accounts for more than 42% in all words. We still have more than 33M words, accounting for 34.3%, which neither begin nor end a multi-source-word translation zone. Apparently, translations that begin/end on words $\in B_y$/$\in E_y$ are preferable to those which begin/end on other words.

Yet another interesting study is to compare translation boundaries with constituent boundaries deduced from source-side parse trees. In doing so, we can know further how well constituent boundary

Classification Task	Avg. Accuracy (%)
B_y/B_n	46.9
E_y/E_n	52.2

Table 2: Average classification accuracy on the development set when we treat constituent boundary deducer (according to source-side parse trees) as a translation boundary classifier.

based syntactic constraints can improve translation quality. We pair the source sentences of our development set with each of the reference translations and include the created sentence pairs in our bilingual training corpus. Then we obtain word alignments on the new corpus (see Section 5.1 for the details of learning word alignments). From the word alignments we obtain translation boundaries (see details in the next section). We parse the source sentences of our development set and obtain constituent boundaries from parse trees.

To make a clear comparison with our translation boundary classifiers (see Section 3.3), we treat constituent boundaries deduced from source-side parse trees as output from beginning/ending boundary classifiers: the constituent beginning boundary corresponds to B_y; the constituent ending boundary corresponds to E_y. We have four reference translations for each source sentence. Therefore we have four translation boundary sets, each of which is produced from word alignments between source sentences and one reference translation set. Each of the four translation boundary sets will be used as a gold standard. We calculate classification accuracy for our constituent boundary deducer on each gold standard and average them finally.

Table 2 shows the accuracy results. The average accuracies on the four gold standard sets are very low, especially for the B_y/B_n classification task. In section 3.3, we will show that our translation boundary classifiers achieve higher accuracy than that of constituent boundary deducer. This suggests that pure constituent boundary based constraints are not the best choice to constrain phrase-based decoding.

3 Learning Translation Boundaries

In this section, we investigate building classifiers to predict translation boundaries. First, we elaborate the acquisition of training instances from word alignments. Second, we build two classifiers with simple features on the obtained training instances. Finally, we evaluate our classifiers on the development set using the "gold" translation boundaries obtained from reference translations.

3.1 Obtaining Translation Boundaries from Word Alignments

We can easily obtain constituent boundaries from parse trees. Similarly, if we have a tree covering both source and target sentence, we can easily get translation boundaries from this tree. Fortunately, we can build such a tree directly from word alignments. We use (Zhang et al., 2008)'s shift-reduce algorithm (SRA) to decompose word alignments into hierarchical trees.

Given an arbitrary word-level alignment as an input, SRA is able to output a tree representation of the word alignment (a.k.a **decomposition tree**). Each node of the tree is a translation zone as we defined in the Section 2. Therefore the first word on the source side of each multi-source-word node is a beginning translation boundary ($\in B_y$); the last word on the source side of each multi-source-word node is an ending translation boundary ($\in E_y$).

Figure 1a shows an example of many-to-many alignment, where the source language is Chinese and the target language is English. Each word is indexed with their occurring position from left to right. Figure 1b is the tree representation of the word alignment after hierarchical analysis using SRA. We use $([i,j],[p,q])$ to denote a tree node, where i,j and p,q are the beginning and ending index in the source and target language, respectively. By checking nodes which cover multiple source words, we can easily decide that the source words {过去, 五, 因故} are in the class B_y and any other words are in the class B_n if we want to train a B_y/B_n classifier with class labels $\{B_y, B_n\}$. Similarly, the source words {次, 飞行, 都, 失败} are in the class E_y and any other words are in the class E_n when we train a E_y/E_n classifier with class labels $\{E_y, E_n\}$.

By using SRA on each word-aligned bilingual sentence, as described above, we can tag each source word with two sets of class labels: $\{B_y, B_n\}$ and $\{E_y, E_n\}$. The tagged source sentences will be used to train our two translation boundary classifiers.

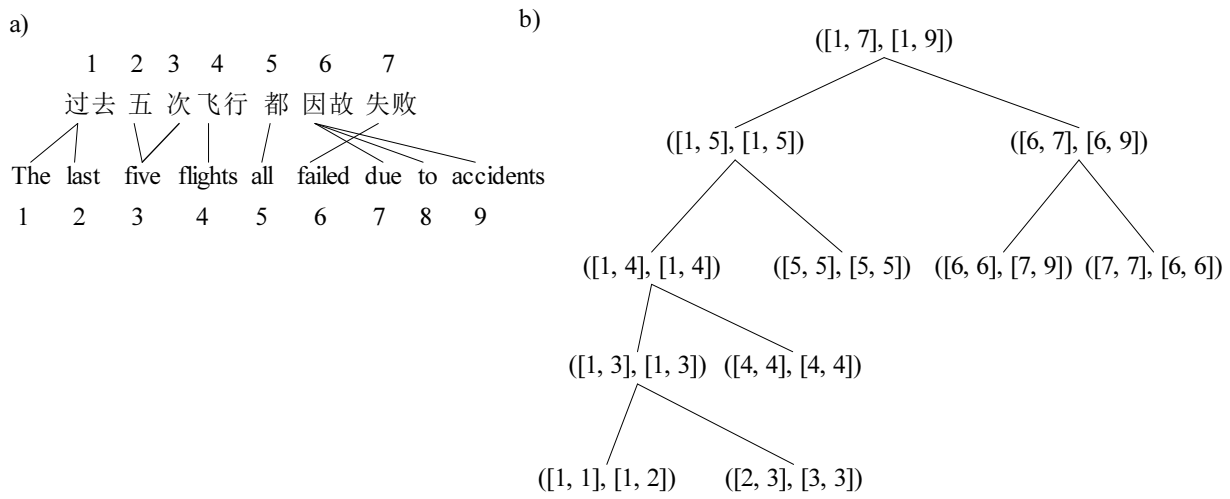

Figure 1: An example of many-to-many word alignment and its tree representation produced by (Zhang et al., 2008)'s shift-reduce algorithm.

3.2 Building Translation Boundary Classifiers

We build two discriminative classifiers based on Maximum Entropy Markov Models (MEMM) (McCallum et al., 2000). One classifier is to predict the word class $\zeta \in \{B_y, B_n\}$ for each source word. The other is to predict the word class $\zeta \in \{E_y, E_n\}$. These two classifiers are separately trained using training instances obtained from our word-aligned training data as demonstrated in the last section.

We use features from surrounding words, including 2 before and 2 after the current word position $(c_{-2}, c_{-1}, c_{+1}, c_{+2})$. We also use class features to train models with Markov order 1 (including class feature $\zeta_{c_{-1}}$), and Markov order 2 (including class features $\zeta_{c_{-1}}, \zeta_{c_{-2}}$).

3.3 Evaluating Translation Boundary Classifiers

How well can we perform these binary classification tasks using the classifiers described above? Can we obtain better translation boundary predictions than extracting constituent boundary from source-side parse trees? To investigate these questions, we evaluate our MEMM based classifiers. We trained them on our 100M-word word-aligned corpus. We ran the two trained classifiers on the development set separately to obtain the B_y/B_n words and E_y/E_n words. Then we built our four gold standards using four reference translation sets as described in Section 2. The average classification accuracy results are shown in Table 3.

	Avg. Accuracy (%)	
Classification Task	MEMM 1	MEMM 2
B_y/B_n	71.7	70.2
E_y/E_n	59.2	58.8

Table 3: Average classification accuracy on the development set for our MEMM based translation boundary classifiers with various Markov orders.

Comparing Table 3 with Table 2, we find that our MEMM based classifiers significantly outperform constituent boundary deducer in predicting translation boundaries, especially in the B_y/B_n classification task, where our MEMM based B_y/B_n classifier (Markov order 1) achieves a relative increase of 52.9% in accuracy over the constituent boundary deducer. In the E_y/E_n classification task, our classifiers also perform much better than constituent boundary deducer.

Then are our MEMM based translation boundary classifiers good enough? The accuracies are still low although they are higher than those of constituent boundary deducer. One reason why we have low accuracies is that our gold standard based evaluation is not established on real gold standards. In other words, we don't have gold standards in terms of translation boundary since different translations

Classification Task	Avg. Accuracy (%)
B_y/B_n	80.6
E_y/E_n	75.7

Table 4: Average classification accuracy on the development set when treating each reference translation set as a boundary classifier.

generate very different translation boundaries. We can measure these differences in reference translations using the same evaluation metric (classification accuracy). We treat each reference translation set as a translation boundary classifier while the other three reference translation sets as gold standards. We calculate the classification accuracy for the current reference translation set and finally average all four accuracies. Table 4 presents the results.

Comparing Table 4 with Table 3, we can see that the accuracy of our translation boundary classification approach is not that low when considering vast divergences of reference translations. The question now becomes, how can classifier output be used to constrain phrase-based decoding, and what is the impact on the system performance of using such constraints.

4 Integrating Translation Boundaries into Decoding

By running the two trained classifiers on the source sentence separately, we obtain two classified word sets: B_y/B_n words, and E_y/E_n words. We can prohibit any translations or reorderings spanning $c_i...c_j$ ($j > i$) where $c_i \notin B_y$ according to the first classifier or $c_j \notin E_y$ according to the second classifier. In such a way, we integrate translation boundaries into phrase-based decoding as hard constraints, which, however, is at the risk of producing no translation covering the whole source sentence.

Alternatively, we introduce soft constraints based on translation boundary that our classifiers predict, similar to constituent boundary based soft constraints in (Cherry, 2008) and (Marton and Resnik, 2008). We add a new feature to the decoder's log-linear model: translation boundary violation counting feature. This counting feature accumulates whenever hypotheses have a partial translation spanning $c_i...c_j$ ($j > i$) where $c_i \notin B_y$ or $c_j \notin E_y$. The weight λ_v of this feature is tuned via minimal error rate training (MERT) (Och, 2003) with other feature weights.

Unlike hard constraints, which simply prevent any hypotheses from violating translation boundaries, soft constraints allow violations of translation boundaries but with a penalty of $\exp(-\lambda_v C_v)$ where C_v is the violation count. By using soft constraints, we can enable the model to prefer hypotheses which are consistent with translation boundaries.

LDC ID	Description
LDC2004E12	United Nations
LDC2004T08	Hong Kong News
LDC2005T10	Sinorama Magazine
LDC2003E14	FBIS
LDC2002E18	Xinhua News V1 beta
LDC2005T06	Chinese News Translation
LDC2003E07	Chinese Treebank
LDC2004T07	Multiple Translation Chinese

Table 5: Training corpora.

5 Experiment

Our baseline system is a phrase-based system using BTGs (Wu, 1997), which includes a content-dependent reordering model discriminatively trained using reordering examples (Xiong et al., 2006). We carried out various experiments to evaluate the impact of integrating translation boundary based soft constraints into decoding on the system performance on the Chinese-to-English translation task of the NIST MT-05 using large scale training data.

5.1 Experimental Setup

Our training corpora are listed in Table 5. The whole corpora consist of 96.9M Chinese words and 109.5M English words in 3.8M sentence pairs. We ran GIZA++ (Och and Ney, 2000) on the parallel corpora in both directions and then applied the "grow-diag-final" refinement rule (Koehn et al., 2005) to obtain many-to-many word alignments. From the word-aligned corpora, we extracted bilingual phrases and trained our translation model.

We used all corpora in Table 5 except for the United Nations corpus to train our MaxEnt based reordering model (Xiong et al., 2006), which con-

sist of 33.3M Chinese words and 35.8M English words. We built a four-gram language model using the SRILM toolkit (Stolcke, 2002), which was trained on Xinhua section of the English Gigaword corpus (181.1M words).

To train our translation boundary classifiers, we extract training instances from the whole word-aligned corpora, from which we obtain 96.9M training instances for the B_y/B_n and E_y/E_n classifier. We ran the off-the-shelf MaxEnt toolkit (Zhang, 2004) to tune classifier feature weights with Gaussian prior set to 1 to avoid overfitting.

We used the NIST MT-03 evaluation test data as our development set (919 sentences in total, 27.1 words per sentence). The NIST MT-05 test set includes 1082 sentences with an average of 27.4 words per sentence. Both the reference corpus for the NIST MT-03 set and the reference corpus for the NIST MT-05 set contain 4 translations per source sentence. To compare with constituent boundary based constraints, we parsed source sentences of both the development and test sets using a Chinese parser (Xiong et al., 2005) which was trained on the Penn Chinese Treebank with an F_1-score of 79.4%.

Our evaluation metric is case-insensitive BLEU-4 (Papineni et al., 2002) using the shortest reference sentence length for the brevity penalty. Statistical significance in BLEU score differences was tested by paired bootstrap re-sampling (Koehn, 2004).

5.2 Using Translation Boundaries from Reference Translations

The most direct way to investigate the impact on the system performance of using translation boundaries is to integrate "right" translation boundaries into decoding which are directly obtained from reference translations. For both the development set and test set, we have four reference translation sets, which are named ref1, ref2, ref3 and ref4, respectively. For the development set, we used translation boundaries obtained from ref1. Based on these boundaries, we built our translation boundary violation counting feature and tuned its feature weight with other features using MERT. When we obtained the best feature weights λs, we evaluated on the test set using translation boundaries produced from ref1, ref2, ref3 and ref4 of the test set respectively.

Table 6 shows the results. We clearly see that using "right" translation boundaries to build soft constraints significantly improve the performance measured by BLEU score. The best result comes from ref4, which achieves an absolute increase of 1.16 BLEU points over the baseline. We believe that the best result here only indicates the lower bound of potential improvement when using right translation boundaries. If we have consistent translation boundaries on the development and test set (for example, we have the same 4 translators build reference translations for both the development and test set), the performance improvement will be higher.

System	BLEU-4 (%)
Base	33.05
Ref1	33.99*
Ref2	34.17*
Ref3	33.93*
Ref4	34.21*

Table 6: Results of using translation boundaries obtained from reference translations. *: significantly better than baseline ($p < 0.01$).

5.3 Using Automatically Learned Translation Boundaries

The success of using translation boundaries from reference translations inspires us to pursue translation boundaries predicted by our MEMM based classifiers. We ran our MEMM1 (Markov order 1) and MEMM2 (Markov order 2) B_y/B_n and E_y/E_n classifiers on both the development and test set. Based on translation boundaries output by MEMM1 and MEMM2 classifiers, we built our translation boundary violation feature and tuned it on the development set. The evaluation results on the test set are shown in Table 7.

From Table 7 we observe that using soft constraints based on translation boundaries from both our MEMM 1 and MEMM 2 significantly outperform the baseline. Impressively, when using outputs from MEMM 2, we achieve an absolute improvement of almost 1 BLEU point over the baseline. This result is also very close to the best result of using translation boundaries from reference translations.

To compare with constituent boundary based syntactic constraints, we also carried out experiments using two kinds of such constraints. One is the

System	BLEU-4 (%)
Base	33.05
Condeducer	33.18
XP+	33.58*
BestRef	34.21*+
MEMM 1	33.70*
MEMM 2	34.04*+

Table 7: Results of using automatically learned translation boundaries. Condeducer means using pure constituent boundary based soft constraint. XP+ is another constituent boundary based soft constraint but with distinction among special constituent types (Marton and Resnik, 2008). BestRef is the best result using reference translation boundaries in Table 6. MEMM 1 and MEMM 2 are our MEMM based translation boundary classifiers with Markov order 1 and 2. *: significantly better than baseline ($p < 0.01$). +: significantly better than XP+ ($p < 0.01$).

Condeducer which uses pure constituent boundary based syntactic constraint: any partial translations which cross any constituent boundaries will be penalized. The other is the XP+ from (Marton and Resnik, 2008) which only penalizes hypotheses which violate the boundaries of a constituent with a label from {NP, VP, CP, IP, PP, ADVP, QP, LCP, DNP}. The XP+ is the best syntactic constraint among all constraints that Marton and Resnik (2008) use for Chinese-to-English translation.

Still in Table 7, we find that both syntactic constraint Condeducer and XP+ are better than the baseline. But only XP+ is able to obtain significant improvement. Both our MEMM 1 and MEMM 2 outperform Condeducer. MEMM 2 achieves significant improvement over XP+ by approximately 0.5 BLEU points. This comparison suggests that translation boundary is a better option than constituent boundary when we build constraints to restrict phrase-based decoding.

5.4 One Classifier vs. Two Classifiers

Revisiting the classification task in this paper, we can also consider it as a sequence labeling task where the first source word of a translation zone is labeled "B", the last source word of the translation zone is labeled "E", and other words are labeled "O". To complete such a sequence labeling task, we built only one classifier which is still based on MEMM (with Markov order 2) with the same features as described in Section 3.2. We built soft constraints based on the outputs of this classifier and evaluated them on the test set. The case-insensitive BLEU score is 33.62, which is lower than the performance of using two separate classifiers (34.04).

We calculated the accuracy for class "B" by mapping "B" to B_y and "E" and "O" to B_n. The result is 67.9%. Similarly, we obtained the accuracy of class "E", which is as low as 48.6%. These two accuracies are much lower than those of using two separate classifiers, especially the accuracy of "E". This suggests that the B_y and E_y are not interrelated tightly. It is better to learn them separately with two classifiers.

Another advantage of using two separate classifiers is that we can explore more constraints. A word c_k can be possibly labeled as B_y by the first classifier and E_y by the second classifier. Therefore we can build soft constraints on span (c_i, c_k) ($c_i \in B_y, c_k \in E_y$) and span (c_k, c_j) ($c_k \in B_y, c_j \in E_y$). This is impossible if we use only one classifier since each word can have only one class label. We can build only one constraint on span (c_i, c_k) or span (c_k, c_j).

6 Related Work

Various approaches incorporate constraints into phrase-based decoding in a soft or hard manner. Our introduction has already briefly mentioned (Cherry, 2008) and (Marton and Resnik, 2008), which utilize source-side parse tree boundary violation counting feature to build soft constraints for phrase-based decoding, and (Xiong et al., 2009), which calculates a score to indicate to what extent a source phrase can be translated as a unit using a bracketing model with richer syntactic features. More previously, (Chiang, 2005) rewards hypotheses whenever they exactly match constituent boundaries of parse trees on the source side.

In addition, hard linguistic constraints are also explored. (Wu and Ng, 1995) employs syntactic bracketing information to constrain search in order to improve speed and accuracy. (Collins et al., 2005) and (Wang et al., 2007) use hard syntactic constraints to perform reorderings according to source-side parse trees. (Xiong et al., 2008) prohibit any swappings

which violate punctuation based constraints.

Non-linguistic constraints are also widely used in phrase-based decoding. The IBM and ITG constraints (Zens et al., 2004) are used to restrict reorderings in practical phrase-based systems.

(Berger et al., 1996) introduces the concept of *rift* into a machine translation system, which is similar to our definition of translation boundary. They also use a maximum entropy model to predict whether a source position is a rift based on features only from source sentences. Our work differs from (Berger et al., 1996) in three major respects.

1) We distinguish a segment boundary into two categories: beginning and ending boundary due to their different distributions (see Table 1). However, Berger et al. ignore this difference.

2) We train two classifiers to predict beginning and ending boundary respectively while Berger et al. build only one classifier. Our experiments show that two separate classifiers outperform one classifier.

3) The last difference is how segment boundaries are integrated into a machine translation system. Berger et al. use predicted rifts to divide a long source sentence into a series of smaller segments, which are then translated sequentially in order to increase decoding speed (Brown et al., 1992; Berger et al., 1996). This can be considered as a hard integration, which may undermine translation accuracy given wrongly predicted rifts. We integrate predicted translation boundaries into phrase-based decoding in a soft manner, which improves translation accuracy in terms of BLEU score.

7 Conclusion and Future Work

In this paper, we have presented a simple approach to learn translation boundaries on source sentences. The learned translation boundaries are used to constrain phrase-based decoding in a soft manner. The whole approach has several properties.

- First, it is based on a simple classification task that can achieve considerably high accuracy when taking translation divergences into account using simple models and features.
- Second, the classifier output can be straightforwardly used to constrain phrase-based decoder.
- Finally, we have empirically shown that, to build soft constraints for phrase-based decoding, translation boundary predicted by our classifier is a better choice than constituent boundary deduced from source-side parse tree.

Future work in this direction will involve trying different methods to define more informative translation boundaries, such as a boundary to begin/end a swapping. We would also like to investigate new methods to incorporate automatically learned translation boundaries more efficiently into decoding in an attempt to further improve search in both speed and accuracy.

References

Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39-71.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, and Surya Mohanty. 1992. Dividing and Conquering Long Sentences in a Translation System. In *Proceedings of the workshop on Speech and Natural Language, Human Language Technology*.

Colin Cherry. 2008. Cohesive Phrase-based Decoding for Statistical Machine Translation. In *Proceedings of ACL.*

David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of ACL*, pages 263–270.

Michael Collins, Philipp Koehn and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *Proceedings of ACL.*

Kevin Knight. 1999. Decoding Complexity in Word Replacement Translation Models. In *Computational Linguistics*, 25(4):607 - 615.

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

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*.

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 *Proceedings of IWSLT*.

Yuval Marton and Philip Resnik. 2008. Soft Syntactic Constraints for Hierarchical Phrase-Based Translation. In *Proceedings of ACL*.

Andrew McCallum, Dayne Freitag and Fernando Pereira 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning 2000*.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL 2000*.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL 2003*.

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatically Evaluation of Machine Translation. In *Proceedings of ACL 2002*.

Brian Roark and Kristy Hollingshead. 2008. Classifying Chart Cells for Quadratic Complexity Context-Free Inference. In *Proceedings of COLING 2008*.

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 901-904.

Chao Wang, Michael Collins and Philipp Koehn 2007. Chinese Syntactic Reordering for Statistical Machine Translation. In *Proceedings of EMNLP*.

Dekai Wu and Cindy Ng. 1995. Using Brackets to Improve Search for Statistical Machine Translation In *Proceedings of PACLIC-IO, Pacific Asia Conference on Language, Information and Computation*.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377-403.

Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, Yueliang Qian. 2005. Parsing the Penn Chinese Treebank with Semantic Knowledge. In *Proceedings of IJCNLP*, Jeju Island, Korea.

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, Ai Ti Aw, Haitao Mi, Qun Liu and Shouxun Lin. 2008. Refinements in BTG-based Statistical Machine Translation. In *Proceedings of IJCNLP 2008*.

Deyi Xiong, Min Zhang, Ai Ti Aw, and Haizhou Li. 2009. A Syntax-Driven Bracketing Model for Phrase-Based Translation. In *Proceedings of ACL-IJCNLP 2009*.

Richard Zens, Hermann Ney, Taro Watanabe and Eiichiro Sumita 2004. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *Proceedings of COLING*.

Hao Zhang, Daniel Gildea, and David Chiang. 2008. Extracting Synchronous Grammars Rules from Word-Level Alignments in Linear Time. In *Proceeding of COLING 2008*.

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