Improving Word Alignment Using Linguistic Code Switching Data

Fei Huang* and **Alexander Yates**
Temple University
Computer and Information Sciences
324 Wachman Hall
Philadelphia, PA 19122
{fei.huang,yates}@temple.edu

Abstract

Linguist Code Switching (LCS) is a situation where two or more languages show up in the context of a single conversation. For example, in English-Chinese code switching, there might be a sentence like "我们15分钟后有个meeting (We will have a meeting in 15 minutes)". Traditional machine translation (MT) systems treat LCS data as noise, or just as regular sentences. However, if LCS data is processed intelligently, it can provide a useful signal for training word alignment and MT models. Moreover, LCS data is from non-news sources which can enhance the diversity of training data for MT. In this paper, we first extract constraints from this code switching data and then incorporate them into a word alignment model training procedure. We also show that by using the code switching data, we can jointly train a word alignment model and a language model using co-training. Our techniques for incorporating LCS data improve by 2.64 in BLEU score over a baseline MT system trained using only standard sentence-aligned corpora.

1 Introduction

Many language users are competent in multiple languages, and they often use elements of multiple languages in conversations with other speakers with competence in the same set of languages. For example, native Mandarin speakers who also speak English might use English words in a Chinese sentence, like "你知道这个问题的solution吗？(Do you know the solution to this problem ?)". This phenomenon of mixing languages within a single utterance is known as Linguistic Code Switching (LCS). Examples of these utterances are common in communities of speakers with a shared competency in multiple languages, such as Web forums for Chinese emigrés to the United States. For example, more than 50% of the sentences we collected from a Web forum (MITBBS.com) contains both Chinese and English.

Traditional word alignment models take a sentence-level aligned corpus as input and generate word-level alignments for each pair of parallel sentences. Automatically-gathered LCS data typically contains no sentence-level alignments, but it still has some advantages for training word alignment models and machine translation (MT) systems which are worth exploring. First, because it contains multiple languages in the same sentence and still has a valid meaning, it will tell the relationship between the words from different languages to some extent. Second, most LCS data is formed during people's daily conversation, and thus it contains a diversity of topics that people care about, such as home furnishings, cars, entertainment, *etc*, that may not show up in standard parallel corpora. Moreover, LCS data is easily accessible from Web communities, such as MITBBS.com, Sina Weibo, Twitter, etc.

However, like most unedited natural language text on the Web, LCS data contains symbols like emotions, grammar and spelling mistakes, slang and strongly idiomatic usage, and a variety of other phenomena that are difficult to handle. LCS data with different language pairs may also need special handling. For instance, Sinha and Thakur (2005) focus on words in mixed English and Hindi texts where a single word contains elements from both languages; they propose techniques for translating such words into both pure English and pure Hindi. Our study focuses on Chinese-English LCS, where this is rarely a problem,

*The author is working at Raytheon BBN Technologies now

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 1–9,
Gothenburg, Sweden, April 26-30 2014.

but for other language pairs, Sinha and Thakur's techniques may be required as preprocessing steps. Primarily, though, LCS data requires special-purpose algorithms to use it for word alignment, since it contains no explicit alignment labels.

In this paper, we investigate two approaches to using LCS data for machine translation. The first approach focuses exclusively on word alignment, and uses patterns extracted from LCS data to guide the EM training procedure for word alignment over a standard sentence-aligned parallel corpus. We focus on two types of patterns in the LCS data: first, English words are almost never correct translations for any Chinese word in the same LCS utterance. Second, for sentences that are mostly Chinese but with some English words, if we propose substitutes for the English words using a Chinese language model, those substitutes are often good translations of the English words. We incorporate these patterns into EM training via the posterior regularization framework (Ganchev et al., 2010).

Our second approach treats the alignment and language model as two different and complementary views of the data. We apply the co-training paradigm for semi-supervised learning to incorporate the LCS data into the training procedures for the alignment model and the language model. From the translation table of the alignment model, the training procedure finds candidate translations of the English words in the LCS data, and uses those to supplement the language model training data. From the language model, the training procedure identifies Chinese words that complete the Chinese sentence with high probability, and it uses the English word paired with these completion words as additional training points for translation probabilities. These models are trained repeatedly until they converge to similar predictions on the LCS data. In combination with a larger phrase-based MT system (Koehn et al., 2003), these two training procedures yield an MT system that achieves a BLEU score of 31.79 on an English-to-Chinese translation task, an improvement of 2.64 in BLEU score over a baseline MT system trained on only our parallel corpora.

The rest of this paper is organized as follows. The next section presents related work. Section 3 gives an overview of word alignment. Sections 4 and 5 detail our two algorithms. Section 6 presents our experiments and discusses results, and Section 7 concludes and discusses future work.

2 Related Work

There has been a lot of research on LCS from the theoretical and socio-linguistic communities (Nilep, 2006; De Fina, 2007). Computational research on LCS has studied how to identify the boundaries of an individual language within LCS data, or how to predict when an utterance will switch to another language (Chan et al., 2004; Solorio and Liu, 2008). Manandise and Gdaniec (2011) analyzed the effect on machine translation quality of LCS of Spanish-English and showed that LCS degrades the performance of the syntactic parser. Sinha and Thakur (2005) translate mixed Hindi and English (Hinglish) to pure Hindi and pure English by using two morphological analyzers from both Hindi and English. The difficulty in their problem is that Hindi and English are often mixed into a single word which uses only the English alphabet; approaches based only on the character set cannot tell these words apart from English words. Our current study is for a language pair (English-Chinese) where the words are easy to tell apart, but for MT using code-switching data for other language pairs (such as Hindi-English), we can leverage some of the techniques from their work to separate the tokens into source and target.

Like our proposed methods, other researchers have used co-training before for MT (Callison-Burch and Osborne, 2003). They use target strings in multiple languages as different views on translation. However, in our work, we treat the alignment model and language model as different views of LCS data.

In addition to co-training, various other semi-supervised approaches for MT and word alignment have been proposed, but these have relied on sentence alignments among multiple languages, rather than LCS data. Kay (2000) proposes using multiple target documents as a way of informing subsequent machine translations. Kumar et al. (2007) described a technique for word alignment in a multi-parallel sentence-aligned corpus and showed that this technique can be used to obtain higher quality bilingual word alignments. Other work like (Eisele, 2006) took the issue one step further that they used bilingual translation systems

which share one or more common pivot languages to build systems which non-parallel corpus is used. Unlike the data in these techniques, LCS data requires no manual alignment effort and is freely available in large quantities.

Another line of research has attempted to improve word alignment models by incorporating manually-labeled word alignments in addition to sentence alignments. Callison-Burch et al. (2004) tried to give a higher weight on manually labeled data compared to the automatic alignments. Fraser and Marcu (2006) used a log-linear model with features from IBM models. They alternated the traditional Expectation Maximization algorithm which is applied on a large parallel corpus with a discriminative step aimed at increasing word-alignment quality on a small, manually word-aligned corpus. Ambati et al.(2010) tried to manually correct the alignments which are informative during the unsupervised training and applied them to an active learning model. However, labeled word alignment data is expensive to produce. Our approach is complementary, in that we use mixed data that has no word alignments, but still able to learn constraints on word alignments.

Our techniques make use of posterior regularization (PR) framework (Ganchev et al., 2010), which has previously been used for MT (Graca et al., 2008), but with very different constraints on EM training and different goals. (Graca et al., 2008) use PR to enforce the constraint that one word should not translate to many words, and that if a word s translates to a word t in one MT system, then a model for translation in the reverse direction should translate t to s. Both of these constraints apply to sentence-aligned training data directly, and complement the constraints that we extract from LCS data.

3 Statistical Word Alignment

Statistical word alignment (Brown et al., 1994) is the task identifying which words are translations of each other in a bilingual sentence corpus. It is primarily used for machine translation. The input to an alignment system is a sentence-level aligned bilingual corpus, which consists of pairs of sentences in two languages. One language is denoted as the target language, and the other language as the source language.

We now introduce the baseline model for word alignment and how we can incorporate the LCS data to improve the model. IBM Model 1 (Brown et al., 1994) and the HMM alignment model (Vogel et al., 1996) are cascaded to form the baseline model for alignment. These two models have a similar formulation $\mathcal{L} = P(t, a|s) = P(a)\prod_j P(t_j|s_{a_j})$ with a different distortion probability $P(a)$. s and t denote the source and target sentences. a is the alignment, and a_j is the index of the source language word that generates the target language word at position j. The HMM model assumes the alignments have a first-order Markov dependency, so that $P(a) = \prod_j P(a_j|a_j - a_{j-1})$. IBM Model 1 ignores the word position and uses a uniform distribution, so $P(a) = \prod_j P(a_j)$ where $P(a_j) = \frac{1}{|t|}$, where $|t|$ is the length of t.

Expectation Maximization (Dempster et al., 1977) is typically used to train the alignment model. It tries to maximize the marginal likelihood of the sentence-level aligned pairs. For the HMM alignment model, the forward-backward algorithm can be used the optimize the posterior probability of the hidden alignment a.

4 Learning Constraints for Word Alignments from LCS Data

We observed that most LCS sentences are predominantly in one language, which we call the *majority* language, with just a small number of words from another language, which we call the *minority* language. The grammar of each sentence appears to mirror the structure of the majority language. Speakers appear to be substituting primarily content words from the minority language, especially nouns and verbs, without changing the structure of the majority language. In this section, we explain two types of constraints we extract from the LCS data that can be helpful for guiding the training of a word alignment model, and we describe how we incorporate those constraints into a full training procedure.

4.1 Preventing bad alignments

After inspecting sentences in our LCS data, we found that the words from the target language occurring in the sentence are highly likely not to be the translation of the remaining source word. Figure 1 shows an example LCS sentence where the speaker has replaced the Chinese word “要求” with the corresponding English word “request”.

Figure 1: The upper sentence is the original LCS sentence. The bottom ones are its translation in pure Chinese and English. Underlined words are the original words in the LCS sentence.

In most LCS utterances, the minority language replaces or substitutes for words in the majority language, and thus it does not serve as a translation of any majority-language words in the sentence. If we can enforce that a word alignment model avoids pairing words that appear in the same LCS sentence, we can significantly narrow down the possible choices of the translation candidates during word alignment training.

Formally, let t^{LCS} be the set of target (Chinese) words and s^{LCS} be the source (English) words in the same sentence of the LCS data. According to our observation, each s_j^{LCS} in s^{LCS} should not be aligned with any word t_i^{LCS} in t^{LCS}. We call every target-source word pair (t_i^{LCS}, s_j^{LCS}) from LCS data a **blocked alignment**. For a set of word alignments $\text{WA} = \{(s_w, t_w)\}$ produced by a word alignment model, define

$$\phi_{\text{BA}} = \sum_{(s_w,t_w)\in \text{WA}} \mathbf{1}[(s_w, t_w) \in \text{BA}] \quad (1)$$

where BA is the set of blocked alignments extracted from the LCS data. We want to minimize ϕ_{BA}. Figure 2 shows a graphical illustration of this constraint.

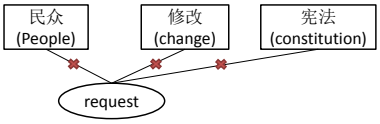

Figure 2: Illustration of the blocked alignment constraint.

4.2 Encouraging alignments with substitutes proposed by a language model

Another perspective of using the LCS data is that if we can find some target word set $t^{similar}$ from the target language which shares similar contexts as the source word s_j^{LCS} in the LCS data, then we can encourage s_j^{LCS} to be aligned with the each word $t_m^{similar}$ in $t^{similar}$. Figure 3 shows example phrases (“民众建议修改”, “民众要求修改”, “民众拒绝修改” etc) that appear in a Chinese language model and which share the same left context and right context as the word “request.” Our second objective is to encourage minority language words like “request” to align with possible substitutes from the majority language’s language model. If we see any of “建议, 要求, 拒绝” in the parallel corpus, we should encourage the word “request” to be aligned with them. We call this target-source word pair $(t_m^{similar}, s_j^{LCS})$ an **encouraged alignment**.

Formally, we define

$$\phi_{\text{EA}} = |C| - \sum_{(s_w,t_w)\in \text{WA}} \mathbf{1}[(s_w, t_w) \in \text{EA}] \quad (2)$$

where $|C|$ is the size of the parallel corpus and EA is the encouraged alignment set. We define this expression in such a way that if the optimization procedure minimizes it, it will increase the number of encouraged alignments.

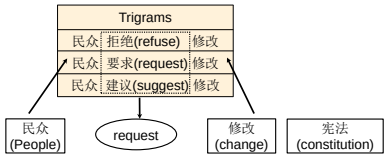

Figure 3: Illustration of the encouraged alignment constraint. The dotted rectangle shows the candidate translations of the English word from the tri-gram output from the language model

Algorithm 1 shows the algorithm of calculating $t^{similar}$. $(t_l^{LCS}, s_j^{LCS}, t_r^{LCS})$ is a (target, source, target)word tuple contained in the LCS data. l and r denote the left and right target words to the source word. We use the language model output from the target language. For each pair of contexts t_l and t_r for the source word, we find the exact match of this pair in the ngram. Then we extract the middle word as the candidates for $t^{similar}$. Here, we only use 3 grams in our experiments, but it is possible to extend this to 5grams, which might lead to further improvements. The EA constraint

Algorithm 1: finding $t^{similar}$

1: **Input:** s^{LCS},t^{LCS}, language model LM
2: Set $t^{similar}$={}
3: Extract the 3 grams $(t_l, t_m, t_r) \in gram_3$ from LM
4: set $S = \{\}$
5: For j from 1 to size($gram_3$)
    if $(t_l^j, t_r^j) \in S$
        add t_m^j into $C_{t_l^j, t_r^j}$
    else
        put (t_l^j, t_r^j) into S
        set $C_{t_l^j, t_r^j} = \{\}$
6: Extract tuple $(t_l^{LCS}, s_j^{LCS}, t_r^{LCS})$
    if $(t_l^{LCS}, t_r^{LCS}) \in S$
        add $C_{t_l^{LCS}, t_r^{LCS}}$ into $t^{similar}$
7: **Output:** $t^{similar}$

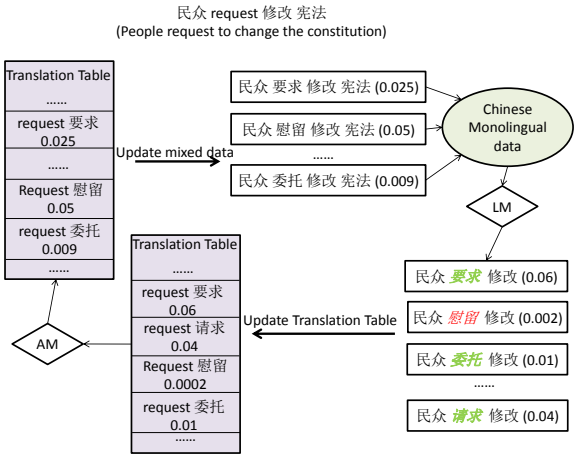

Figure 4: The framework of co-training in word alignment. AM represents alignment model and LM represents language model. Green italic words are the encouraged translation and red italic words are the discouraged translation.

is similar to a bilingual dictionary. However, in the bilingual dictionary, each source word might have several target translations (senses), so it might be ambiguous. The candidate translations used in EA are from language model (3 grams in this paper, but it can be extended to 5 grams), which will always match the contexts. Additionally, the bilingual dictionary contains the standard English/Chinese word pairs. But the LCS data is generated from people's daily conversation; it reflects usage in a variety of domains, including colloquial and figurative usages that may not appear in a dictionary.

4.3 Constrained parameter estimation

We incorporate ϕ_{BA} and ϕ_{EA} into the EM training procedure for the alignment model using posterior regularization (PR) (Ganchev et al., 2010). Formally, let x be the sentence pairs s and t. During the E step, instead of using the posterior $p(a|x)$ to calculate the expected counts, the PR framework tries to find a distribution $q(a)$ which is close to $p(a|x)$, but which also minimizes the properties $\phi(\mathbf{a}, \mathbf{x})$:

$$\min_{q,\xi} [\mathbf{KL}(q(\mathbf{a})||\mathbf{p}(\mathbf{a}|\mathbf{x}, \theta)) + \sigma||\xi||] \quad (3)$$

$$\text{s.t. } \mathbf{E}_{\mathbf{a}\sim q}[\phi(\mathbf{a}, \mathbf{x})] \leq \xi \quad (4)$$

where KL is the Kullback-Leibler divergence, σ is a free parameter indicating how important the constraints are compared with the marginal log likelihood and ξ is a small violation allowed in the optimization. To impose multiple constraints, we define a norm $||\xi||_A = \sqrt{(\xi^t A \xi)}$, where A is a diagonal matrix whose diagonal entries A_{ii} are free parameters that provide weights on the different constraints. Since we only have two constraints here from LCS data, $A = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix}$ where α controls the relative importance of the two constraints.

To make the optimization task in the E-step more tractable, PR transforms it to a dual problem:

$$\max_{\lambda \geq 0, \|\lambda\|_* \leq \sigma} -\log \sum_{\mathbf{a}} p(\mathbf{a}|\mathbf{x}, \theta) \exp\{-\lambda \cdot \phi(\mathbf{a}, \mathbf{x})\}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|_A$. The gradient of this dual objective is $-\mathbf{E}_q[\phi(\mathbf{a}, \mathbf{x})]$. A projected subgradient descent algorithm is used to perform the optimization.

5 Co-training using the LCS data

The above approaches alter the translation and distortion probabilities in the alignment model. However, they leave the language model unchanged. We next investigate a technique that uses LCS data to re-estimate parameters for the language model as well as the alignment model simultaneously. Co-training (Blum and Mitchell, 1998) is a semi-supervised learning technique that requires two different views of the data. It assumes that each example can be described using two different feature sets which are conditionally independent. Also, each feature set of the data should be sufficient to make accurate prediction.

The schema fits perfectly into our problem. We can treat the alignment model and the language model as two different views of the LCS data.

We use the same example "民众request 修改宪法" to show how co-training works, shown in Figure 4. From the translation table generated by the alignment model, we can get a set of candidate translations of "request", such as "'要求","请求",etc. We can find the candidate with the highest probability as the translation. Similarly, from the language model, we can extract all the ngrams containing " 民众" and "修改" as the left and right words and pick the words in the middle such as " 建议, 要求, 拒绝" etc as the candidate translations. We can then use the candidate with the highest probability as the translation for "request". Thus both models can predict translations for the English (minority language) in this example. Each model's predictions can be used as supplemental training data for the other model.

Algorithm 2 shows the co-training algorithm for word alignment. At each iteration, a language model and an alignment model are trained. The language model is trained on a Chinese-only corpus plus a corpus of probabilistic LCS sentences where the source words are replaced with target candidates from the alignment model. The alignment model is retrained using a translation table which is updated according to the output word pairs from the language model output and the LCS data. In order to take the sentence probability into consideration, we modify the language model training procedure: when it counts the number of times each ngram appears, instead of adding 1, it adds the probability from the translation model for ngrams in the LCS data that contain predicted translations.

Algorithm 2: Co-training for word alignment and language modeling

1: **Input:** parallel data X_p, LCS data X_{LCS}, language model training data X_l
2: Initialize translation table tb for IBM1 model
3: For iteration from 1 to MAX
  $tb \leftarrow$ Train-IBM(X_p)
  $tb' \leftarrow$ Train-HMM($X_p|tb$)
4: For each sentence x_i in X_{LCS}:
    For each source word s_j in x_i:
      1) find the translation t_j of s_j with with probability p_j from tb'
      2) replace s_j with t_j and update sentence's probability $p^s = p^s * p_j$
    $X_l^{new} \leftarrow X_l \cup x_i$
5: LM $\leftarrow$ Train-LM(X_l^{new})
6: Extract the tri-gram $gram_3$ from LM
7: For each sentence x_i in X_{LCS}:
    run Algorithm 1: finding $t^{similar}$
8: update tb' using (t_m, s_j) where $t_m \in t^{similar}$ and $s_j \in x_i$
9: **End For**
10: **Output:** word alignment for X_p and LM

6 Experiments and Results

6.1 Experimental Setup

We evaluated our LCS-driven training algorithms on an English-to-Chinese translation task. We use Moses (Koehn et al., 2003), a phrase-based translation system that learns from bilingual sentence-aligned corpora as the MT system. We supplement the baseline word alignment model in Moses with our LCS data, constrained training procedure, and co-training algorithm as well as IBM 3 model. Because IBM 3 model is a fertility based model which might also alleviate some of the problems caused by LCS data. To clarify, we use IBM1 model and HMM models in succession for the baseline. We trained the IBM1 model first and used the resulting parameters as the initial parameter values to train HMM model. Parameters for the final MT system are tuned with Minimum Error Rate Training (MERT) (Och, 2003). The tuning set for MERT is the NIST MT06 data set, which includes 1664 sentences. We test the system on NIST MT02 (878 sentences). To evaluate the word alignment results, we manually aligned 250 sentences from NIST MT02 data set. For simplicity, we only have two types of labels for evaluating word alignments: either two words are aligned together or not. (Previous evaluation metrics also consider a third label for "possible" alignments.) Out of the word-aligned data, we use 100 sentences as a development set and the rest as our testing set.

Our MT training corpus contains 2,636,692 sentence pairs from two parallel corpora: Hong Kong News (LDC2004T08) and Chinese English News Magazine Parallel Text (LDC2005T10). We use the Stanford Chinese segmenter to segment the Chinese data. We use a ngram model package called SRILM (Stolcke, 2002) to train

the language model. Because our modified ngram counts contain factions, we used Witten-Bell smoothing(Witten and Bell, 1991) which supports fractional counts. The 3-gram language model is trained on the Xinhua section of the Chinese Gigaword corpus (LDC2003T09) as well as the Chinese side of the parallel corpora. We also removed the sentences in MT02 from the Gigaword corpus if there is any to avoid the biases.

We gather the LCS data from "MITBBS.com," a popular forum for Chinese people living in the United States. This forum is separated by discussion topic, and includes topics such as "Travel", "News", and "Living style". We extract data from 29 different topics. To clean up the LCS data, we get rid of HTML mark-up, and we remove patterns that are commonly repeated in forums, like "Re:" (for "reply" posts) and "[转载]" (for "repost"). We change all English letters written in Chinese font into English font. We stem the English words in both the parallel training data and the LCS data. After the cleaning step, we have 245,470 sentences in the LCS data. 120,922 of them actually contain both Chinese and English in the same sentence. 101,302 of them contain only Chinese, and we add these into the language model training data. We discard the sentences that only contain English.

6.2 Word Alignment Results

In order to incorporate the two constraints during the Posterior Regularization, we need to tune the parameters σ which controls the weights between the constraints and the marginal likelihood and α which controls the relative importance between two constraints on development data. We varied σ from 0.1 to 1000 and varied α over the set $\{0.01, 0.1, 1, 10, 100\}$. After testing the 25 different combinations of σ and α on the development data, we find that the setting with $\sigma = 100$ and $\alpha = 0.1$ achieves the best performance. During PR training, we trained the model 20 iterations for the dual optimization and 5 iterations for the modified EM.

Table 1 shows the word alignment results. We can see that incorporating the LCS data into our alignment model improves the performance. Our best co-training+PR$^+$ system outperforms the baseline by 8 points. Figure 5 shows an example of how BA is extracted from LCS data can help the word alignment performance. The upper figure shows that alignment by the baseline system. We can see that the word "badminton" is aligned incorrectly with word "陶菲克(Taufik)" . However, in the LCS data, we see that " 陶菲克(Taufik)" and "badminton" appear in the same sentence "陶菲克的badminton太厉害了(Taufik plays badminton so well)" and by adding the blocked constraint into the alignment model, it correctly learns that " 陶菲克(Taufik)" should be aligned with something else, and it finds "Taufik" at end. Table 2 shows some of the translations of "badminton" before and after incorporating the LCS data. We can see that it contains some wrong translations like "乒乓球室(pingpong room)","陶菲克(Taufik)"etc using baseline model. After using the LCS data as constraints and the co-training framework, these wrong alignments are eliminated and the translation "羽球(another way of expressing badminton)" get a higher probability. We found that IBM 3 model can also correct this specific case. However, our co-training+PR$^+$ system still outperforms it by 6 points.

System	F1
Baseline	0.68
IBM 3	0.70
PR+BA	0.71
PR+EA	0.70
PR$^+$	0.73
co-training	0.74
co-training+PR$^+$	0.76

Table 1: Word alignment results (PR$^+$ means PR+BA+EA).

Figure 6 shows an example of how EA is extracted from LCS data can help the word alignment. The solid lines show the alignment by the baseline model and we can see that the word "compiled" is not aligned with any Chinese word. After using the LCS data and the language model, we find that "集纳(compile)" shows up in the same context "书(book) 起来(up)"as "compile" along with "装订(staple)" and "订(staple)", therefore "(compile, 集纳)" will be an encouraged alignment. After adding the EA constraint, the model learns that "compile" should be aligned with "集纳".

6.3 Phrase-based machine translation

In this section, we investigated whether improved alignments can improve MT performance. We

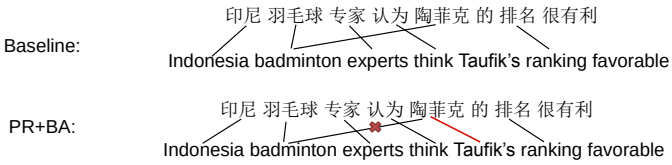

Figure 5: After incorporating the BA constraint from the LCS data, the word "Taufik(陶菲克)" is aligned correctly.

Baseline		PR+co-training	
Translation	Probability	Translation	Probability
羽毛球(badminton)	0.500	羽毛球(badminton)	0.500
乒乓球(pingpong)室(room)	0.500	羽球(two of the three characters in badminton)	0.430
打(play)羽毛(feather)	0.250	打(play)羽毛(feather)	0.326
羽毛球(shuttlecock)头(head)	0.125	羽毛球(shuttlecock)头(head)	0.105
...	...	...	...
陶菲克(Taufik)	0.005	网球拍(racket)	0.002

Table 2: Translation tables of "badminton" before and after incorporation of LCS data.

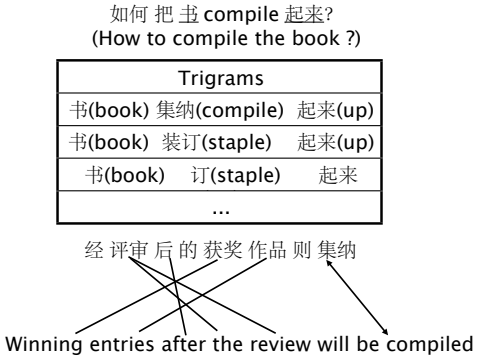

Figure 6: After incorporating the EA constraint from the LCS data, the word "compiled(集纳)" is aligned correctly.

use different word alignment models' outputs as the first step for Moses and keep the rest of Moses system the same. We incorporate Moses's eight standard features as well as the lexicalized reordering model. We also use the grow-diag-final and alignment symmetrization heuristic.

Table 3 shows the machine translation results. We can see that 3 techniques we proposed for word alignment all improve the machine translation result over the baseline system as well as the IBM 3 model. However, although co-training has a bigger improvement on the word alignment compared with PR$^+$, it actually has a lower BLEU score. This phenomenon shows that the improvement in the word alignment does not necessarily lead to the improvement on machine translation. After combining the co-training and the PR$^+$ together, co-training+PR$^+$ improved slightly over PR$^+$ for MT.

System	BLEU score
Baseline	29.15
IBM 3	30.24
PR$^+$	31.59*
co-training	31.04*
co-training+PR$^+$	31.79*

Table 3: Machine translation results. All entries marked with an asterisk are better than the baseline with 95% statistical significance computed using paired bootstrap resampling (Koehn, 2004).

7 Conclusion and Future Work

In this paper, we explored two different ways to use LCS data in a MT system: 1) PR framework to incorporate with Blocked Alignment and Encouraged Alignment constraints. 2) A semi-supervised co-training procedure. Both techniques improve the performance of word alignment and MT over the baseline. Our techniques are currently limited to sentences where the LCS data contains very short (usually one word) phrases from a minority language. An important line of investigation for generalizing these approaches is to consider techniques that cover longer phrases in the minority language; this can help add more of the LCS data into training.

Acknowledgements

This work was supported in part by NSF awards 1065397 and 1218692.

References

S.and Carbonell J. Ambati, V.and Vogel. 2010. Active semi-supervised learning for improving word alignment. In *In Proceedings of the Active Learning for NLP Workshop, NAACL*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Annual Conference on Computational Learning Theaory*.

P. F. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1994. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chris Callison-Burch and Miles Osborne. 2003. Co-training for statistical machine translation. In *In Proceedings of the 6th Annual CLUK Research Colloquium*.

Chris Callison-Burch, David Talbot, and Miles Osborne. 2004. Statistical machine translation with word- and sentence-aligned parallel corpora. In *In Proceedings of ACL*.

J. Y. C. Chan, P. C. Ching, and H. M. LEE, T.and Meng. 2004. Detection of language boundary in code-switching utterances by bi-phone probabilities. In *In Proceedings of the International Symposium on Chinese Spoken Language Processing*.

A De Fina. 2007. Code-switching and the construction of ethnic identity in a community of practice. In *Language in Society*, volume 36.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. In *Royal Statistical Society, Ser*, volume 39.

Andreas Eisele. 2006. Parallel corpora and phrase-based statistical machine translation for new language pairs via multiple intermediaries. In *International Conference on Language Resources and Evaluation*.

Alex Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *In Proceedings of ACL*.

Kuzman Ganchev, J. Graca, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. In *Journal of Machine Learning Research*, volume 11.

J. Graca, K. Ganchev, and B. Taskar. 2008. Expectation maximization and posterior constraints. In *NIPS*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL-HLT*.

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

Shankar Kumar, Franz Josef Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *EMNLP*.

Esme Manandise and Claudia Gdaniec. 2011. Morphology to the rescue redux: Resolving borrowings and code-mixing in machine translation. In *SFCM*.

C. Nilep. 2006. Code switching in sociocultural linguistics. In *Colorado Research in Linguistics*, volume 19.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*.

R.M.K. Sinha and A. Thakur. 2005. Machine translation of bi-lingual hindi-english (hinglish) text. In *In Proceedings of the 10th Conference on Machine Translation*.

T. Solorio and Y. Liu. 2008. Learning to predict code-switching points. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A. Stolcke. 2002. An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904.

S. Vogel, H. Ney, and C. Tillmann. 1996. Hmm-based word alignment in statistical translation. In *In Proc.COLING*.

Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabil- ities of novel events in adaptive text compression. In *IEEE Transactions on Information Theory*, volume 4, pages 1085–1094.