Using Features from a Bilingual Alignment Model in Transliteration Mining

Takaaki Fukunishi
Doshisha University
dtk0706@mail4.doshisha.ac.jp

Andrew Finch
NICT
andrew.finch@nict.go.jp

Seiichi Yamamoto
Doshisha University
seyamamo@mail.doshisha.ac.jp

Eiichiro Sumita
NICT
eiichiro.sumita@nict.go.jp

Abstract

In this paper we present a novel method for selecting transliteration word pairs from a set of candidate word pairs when mining for training data. Our method relies on a Bayesian technique that simultaneously co-segments and force-aligns the bilingual segments. The Bayesian model strongly rewards the re-use of features already present in its model, resulting in a very compact and efficient model. Our idea relies on the assumption that genuine transliteration pairs can be derived by using bilingual sequence pairs already present in the model, or at worst by introducing a very short unobserved pair into the derivation. We assume that incorrect pairs are likely to have larger contiguous segments that are costly to force-align with our model. We use features derived from the co-segmentation (alignment) of the candidate pair in combination with other heuristic features to train a classifier to label whether or not the candidate pair is a genuine transliteration pair. To evaluate our approach we used the all data-tracks from the 2010 Named-entity Workshop (NEWS2010). Our results show that the new features we propose are powerfully predictive, enabling our approach to achieve levels of performance on this task that are comparable to the state of the art.

1 Introduction

For some language pairs, especially those that use the same or very similar character sets, named entities are commonly unchanged in the process of translation between the languages. For example the term ‘Michael Jackson’ is used as is in the English, German and Italian languages. However, in languages that do not share the same writing system, such expressions are transcribed into the respective native writing system, usually in such a manner as to preserve the phonetics as far as possible. So for example, in Japanese the name would be transcribed into the katakana alphabet as マイケル・ジャクソン (MA-I-KE-RU・ JI-YA-KU-SO-N). The form in parentheses is a romanized (rōmaji) form of the preceding Japanese character sequence in Japanese script (katakana), where each roman character or character pair corresponds to a single character in the Japanese writing system, and furthermore corresponds very closely to the English phonetics of the character sequence. We will come back to this correspondence in the next section. This process of transcription from one language into another, usually based on phonetics, is known as transliteration.

Transliteration mining is the process of obtaining lists of bilingual word pairs (we will refer to these as *transliteration pairs*) automatically, that is pairs of words that are transliterations of each other in parallel or comparable corpora. The mined word pairs have many applications, for example as data for training a transliteration generation system, for the enhancement of the bilingual dictionary of a machine translation system to improve lexical coverage, and in query term translation for cross-language information retrieval.

2 Previous Work

The field of transliteration mining is currently being actively researched and there is a wealth of previous research (Brill et al., 2001; Lee and Chang, 2003a; Bilac and Tanaka, 2005; Tsuji and Kageura,

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 49–57,
Chiang Mai, Thailand, November 12, 2011.

2006; Oh and Isahara, 2006; Jiampojamarn et al., 2010; Darwish, 2010; Khapra et al., 2010; Nabende, 2010; Noeman and Madkour, 2010), and recently a shared task in the 2010 ACL Named Entities Workshop (NEWS2010) (Kumaran and Li, 2010).

One common strategy to determine cross-lingual phonetic similarity between words is to transcribe them into the roman alphabet and then use character level similarity measures to compare them, for example normalized edit distance (Jiampojamarn et al., 2010). In practice this seems to be an effective technique; in the previous example, it is easy to see that the romanized string 'maikeru jiyakuson' will be reasonably close in terms of edit distance to the English 'michael jackson', but very likely to be distanced from other English strings that it is not a transliteration of.

A large advantage of these approaches is that they can often be developed without the need to collect a training corpus. On the other hand, a potential drawback of these methods is that they are language dependent in nature, simply because they rely on a language specific romanization scheme. Furthermore, performance will depend on the particular romanization scheme chosen, and often there are several to choose from, in addition to bespoke romanization schemes that might be devised for this task (for example, deleting diacritics and performing character substitutions in European languages (Jiampojamarn et al., 2010)). In Japanese, for example, there are three main competing systems for romanizing Japanese kana characters: the Hepburn, Kunrei-shiki Rōmaji, and Nihon-shiki Rōmaji, romanization systems.

One way to eliminate this language dependency is to build a transliteration generation system to transduce a transliterated string into the other language, and then use a heuristic operating at the character level to measure the string-similarity between the two character sequences. This approach is taken by (Noeman and Madkour, 2010) who use an FST to generate a set of candidate transliterations and an FSA to accept those that can be used to form transliteration pairs. The approach is also used in the generation-based models of (Jiampojamarn et al., 2010), where forward and backward generated transliterations are compared by edit distance against the corresponding strings in the other languages; a score consisting of weighted edit distances of these comparisons in both directions was used to classify the candidate transliteration pair. Other examples of the use of this approach include: (Lee and Chang, 2003b; Tsuji and Kageura, 2006).

A second advantage of approaches that do not require a system for phonetically transcribing a language is that these approaches can handle non-phonetic transcriptions if necessary. For example, the words 'personal computer' would in Japanese be transcribed into 'PA-SO-KO-N', a contraction of the original word pair. The transcription of Japanese kanji into their rōmaji readings is another example commonly encountered in real-world Japanese named entity translation.

The approach we take in this paper is a direct approach that does not rely on an intermediate representation, but rather a direct grapheme-to-grapheme mapping between the languages. We use a generative model directly to assess whether two strings constitute a transliteration pair and avoid the necessity to explicitly generate strings in either language. This type of approach was taken by (Lee and Chang, 2003b), who use a noisy channel model to assess transliteration pair candidates. Our approach differs from theirs in the Bayesian model that we employ. Bayesian models such as the one we use have been successfully applied to transliteration generation (Finch and Sumita, 2010; Huang et al., 2011) and offer several benefits; primarily the technique has the ability to train models whilst avoiding over-fitting the data, and can typically construct compact models that have only a small number of well-chosen parameters. Our system further differs from theirs in that our underlying generative transliteration model is based on the joint source-channel model (Li et al., 2004), and is symmetric with respect to source and target language.

In the next section we will briefly describe the Dirichlet process model that drives the co-segmentation process that underpins our technique. We then present the methodology we use to exploit features from samples taken from this training process to determine whether two words constitute a transliteration pair. Next we describe the set of experiments we performed to investigate the effectiveness of our system on data from all the NEWS2010 shared tasks on transliteration mining, and also on a similar English-Japanese corpus that we constructed, and present our results in the following section. Finally, we conclude and offer some directions for future research.

Throughout the paper we use the following acronyms as shorthand for the various languages:

Ar=Arabic, En=English, Ch=Chinese, Hi=Hindi, Ja=Japanese, Ru=Russian, Ta=Tamil.

3 Using Features from Alignment

Our alignment model is based on a Dirichet process model: a stochastic process defined over a set S (in our case, the set of all possible bilingual sequence-pairs) whose sample path is a probability distribution on S. For brevity we provide only a brief description of the alignment model; for a full description, the reader is referred to (Finch and Sumita, 2010).

3.1 Dirichlet Process Model

Intuitively, the Dirichlet process model has two basic components: a model for generating an outcome that has already been generated at least once before, and a second model that assigns a probability to an outcome that has not yet been produced. To encourage the re-use of model parameters, the probability of generating a novel bilingual sequence-pair is considerably lower then the probability of generating a previously observed sequence pair. The probability distribution over these bilingual sequence-pairs (including an infinite number of unseen pairs) is learned directly from unlabeled data by Bayesian inference of the hidden co-segmentation of the corpus.

More formally, the underlying stochastic process for the generation of a corpus composed of bilingual phrase pairs γ is usually written in the following from:

$$\begin{aligned} G|_{\alpha,G_0} &\sim DP(\alpha, G_0) \\ (\mathbf{s}_k, \mathbf{t}_k)|G &\sim G \end{aligned} \tag{1}$$

G is a discrete probability distribution over the all bilingual sequence-pairs according to a *Dirichlet process prior* with *base measure* G_0 and concentration parameter α. The concentration parameter $\alpha > 0$ controls the variance of G; intuitively, the larger α is, the more similar G_0 will be to G. For the *base measure* G_0 that controls the generation of novel sequence-pairs, we use the joint spelling model described in (Finch and Sumita, 2010), that assigns exponentially smaller probabilities with increasing source/target sequence length.

3.1.1 The Generative Model

The generative model is given in Equation 2 below. The equation assignes a probability to the k^{th} bilingual sequence-pair $(\mathbf{s}_k, \mathbf{t}_k)$ in a derivation of the corpus, given all of the other sequence-pairs observed so far $(\mathbf{s}_{-k}, \mathbf{t}_{-k})$. Here $-k$ is read as: "up to but not including k".

$$p((\mathbf{s}_k, \mathbf{t}_k))|(\mathbf{s}_{-k}, \mathbf{t}_{-k})) = \frac{N((\mathbf{s}_k, \mathbf{t}_k)) + \alpha G_0((\mathbf{s}_k, \mathbf{t}_k))}{N + \alpha} \tag{2}$$

In this equation, N is the total number of bilingual sequence-pairs generated so far, $N((\mathbf{s}_k, \mathbf{t}_k))$ is the number of times the sequence-pair $(\mathbf{s}_k, \mathbf{t}_k)$ has occurred in the history.

3.2 Alignment

By repeatedly scoring bilingual sequence pairs with the probability from Equation 2, the algorithm is able to co-segment and align source and target grapheme sequences through an iterative process of Bayesian inference using Gibbs sampling. The training procedure is based on an extension of the forward filtering backward sampling algorithm (Mochihashi et al., 2009) which is too complex to describe in full here, but is covered in detail in (Finch and Sumita, 2010).

An example of an aligned grapheme sequence pair, the output of running this Dirichlet process model on the bilingual data, illustrated in Figure 1. Given such an alignment of source and target grapheme sequences, it is possible to perform generation by monotonic concatenation of grapheme sequence pairs to form words, as in the joint-source channel models of (Li et al., 2004). The probability of generating a bilingual word pair is given by the product of the probabilities of the bilingual grapheme sequence-pairs that generate it. Our idea is built on the assumption that the better able our model is to generate a bilingual word pair, the more likely it is that the word pair is a transliteration pair that we would like to mine. We use the Dirichlet process model to co-segment and align the data, extract features from this segmentation (explained in the next section) and use them to train a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) to classify them as correct or incorrect transliteration pairs.

3.3 Feature Set

Figure 1 shows the bilingual segmentation and alignment together with the scores for each segment for the candidate pair ANDORIYUU (in Japanese) and 'andrew' in English. The scores in

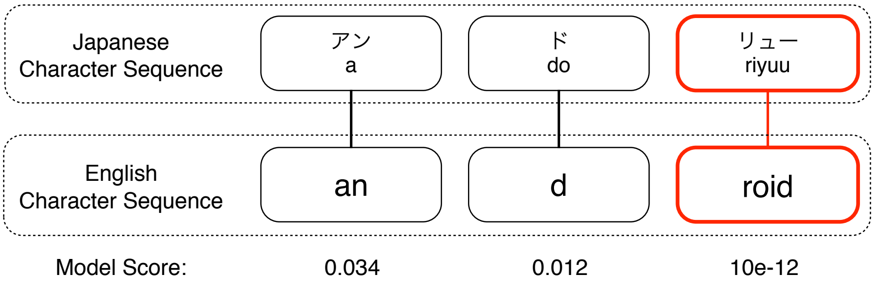

Figure 1: A co-segmentation of the transliteration word-pair candidate ‘andoriyuu’ (Japansese transliteration of the English ‘andrew’) and ‘android’ in English. The figure shows the co-segmentation together with the probabilities of each segment. It can be seen that the segments ‘an’ (Japanese), ‘an’ (English), and ‘do’ (Japanese) and ‘d’ (English) both receive high probabilities from the model, whereas the segment ‘riyuu’ (Japanese) and ‘roid’ in the English receives a very low probability from the model because source and target grapheme sequences are long, and this pairing has not been observed in the corpus.

$f1$	$f2$	$f3$	$f4$
$\frac{logprob}{numsegs}$	$\frac{\|t\|}{\|s\|}$	$\frac{\|s_{bad}\|+\|t_{bad}\|}{\|s\|+\|t\|}$	$minprob$

Table 1: The feature set used by the SVM to classify candidate transliteration pairs.

the figure for each of the bilingual character sequence pairs arise directly from applying Equation 2. In this example the candidate pair is not a transliteration pair, but nonetheless the pair comes quite close to being a transliteration pair because they share a common substring as a prefix. It would be possible to use any of a number of features derived from the alignment and the corresponding score. For example, using the log-probability itself would be possible, but it is strongly determined by sequence length, and therefore not directly comparable across lengths without modification.

The Bayesian model is able to align the corresponding parts of these two words using bilingual sequence pairs that have been observed a number of times in the training corpus. The non-corresponding subsequences of these two words will not have been observed in the data and the Bayesian model therefore must introduce a costly new feature into its model to generate them. In our model, the cost of introducing a new feature increases exponentially with the lengths of the source and target components (see (Finch and Sumita, 2010)). The features (described in detail below) we will use in our experiments are based on two basic hypotheses. The first is that the alignment scores for bad candidate pairs are likely to be lower than scores for good candidate pairs of the same length.

Our second hypothesis is based on the process of forced alignment which co-segments the candidate pair piece by piece. Unobserved pieces typically have extremely low probability and are therefore very costly to introduce into the segmentation hypotheses. As a consequence the model will be driven to generate as much as possible of the sequence pair by re-using the higher probability pieces that have already been observed. Our assumption is that the proportion of the sequence pair that cannot be generated using model features observed in the data will be a good indicator as to whether or not the pair is a correct transliteration pair.

We used a total of 4 features in our SVM classifier, these are shown in Table 1. Feature $f1$ is based on the first of the two assumptions above. Feature $f2$ is a simple length-based heuristic which was expected to be generally useful. Feature $f3$ is designed to capture the idea underpinning the second of the two hypotheses above, that is: what proportion of the candidate pair cannot modeled directly by the features learned by the Dirichlet process model. $f4$ focuses on the score of the weakest part of the derivation.

In Table 1, $logprob$ is the log probability of the sampled derivation of the two grapheme sequences, according to our generative model. $numsegs$ is the number of bilingual segments used in this derivation. $minprob$ is the log probability of the segment with the lowest probability in the derivation. $|s|$ and $|t|$ are the lengths (in graphemes) of the source and target words respectively. $|s_{bad}| + |t_{bad}|$ is the total number of graphemes in both source and target, that are in *bad segments*. Here by *bad segment* we mean a bilingual segment that has not been observed in the training corpus and thus is only receiving a contribution from the base measure component of our Dirichlet process model (a *bad segment* is illustrated in Figure 1 as the rightmost segment in the sequence).

4 Experimental Evaluation

4.1 Corpora

For our experiments we used data from all tracks of the NEWS 2010 Named Entity Workshop (Kumaran et al., 2010b; Kumaran et al., 2010a; Kumaran and Li, 2010). A complete description of this shared task is given in (Kumaran et al., 2010b) and the results for all of the 15 systems evaluated is presented in (Kumaran et al., 2010a).

Our experiments were not part of the official NEWS2010 shared task, but used the same data sets. The training data for this track consisted of title-pairs of interlanguage links between wikipedia articles. These titles are noisy in the sense that they can be sequences of words, only some or even none of which may be transliterations of each other. The proportion of correct transliteration pairs to incorrect pairs in the training data was unknown. In addition, 1000 ‘seed’ pairs of clean data were provided. The seed pairs contained only one word for each language and all were positive examples of transliteration pairs; no negative examples were included in the seed data.

For evaluation, the participants were expected to mine transliteration pairs from the full training set. A set of approximately 1000 interlanguage links (each giving rise to 0, 1 or more transliteration pairs) was randomly sampled from the training data, and not disclosed to the participants. In our experiments we used the same data and the same precision/recall/f-score evaluation metrics that were used in the official runs for the NEWS2010 workshop (refer to (Kumaran et al., 2010a) for full details).

4.2 The Mining Process

A flowchart illustrating the end-to-end process that was used in our experiments to mine transliteration pairs is shown in Figure 2. As can be seen from the figure, the process starts with the Bayesian alignment of the large corpus of noisy title-pairs.

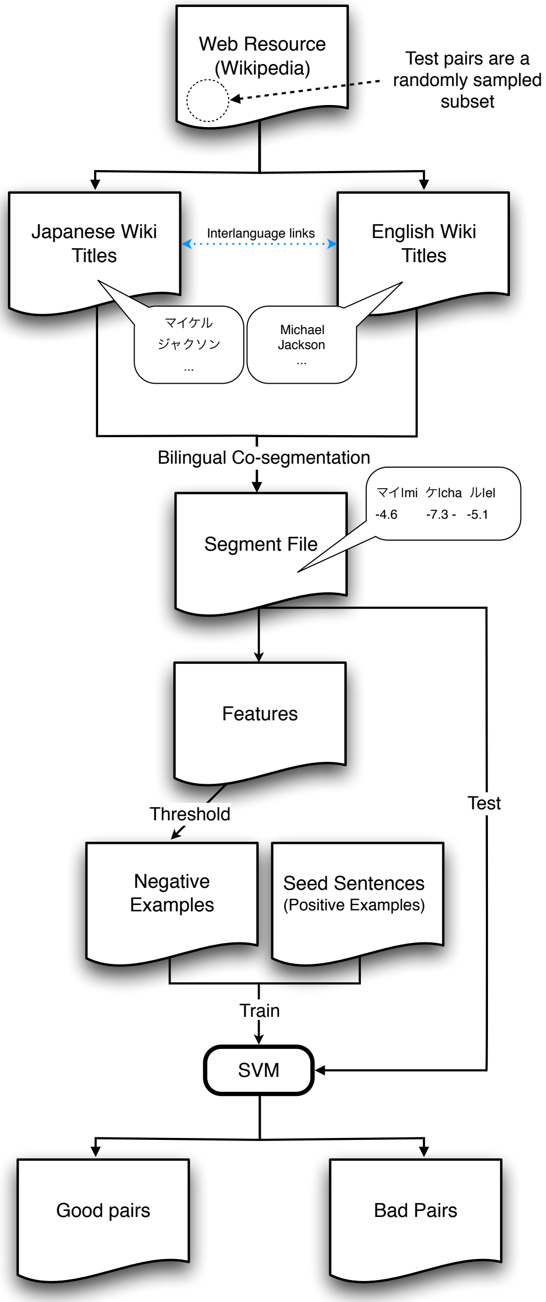

Figure 2: The mining process used in our experiments.

4.3 Negative Examples

No negative examples were provided for this task. (Jiampojamarn et al., 2010) overcame this issue by generating their own set of negative examples. We propose a novel approach that creates a set of negative examples by exploiting the natural clustering that is induced by the features derived from our Bayesian model (see Figure 3). This is described in the following section. We later compare this ap-

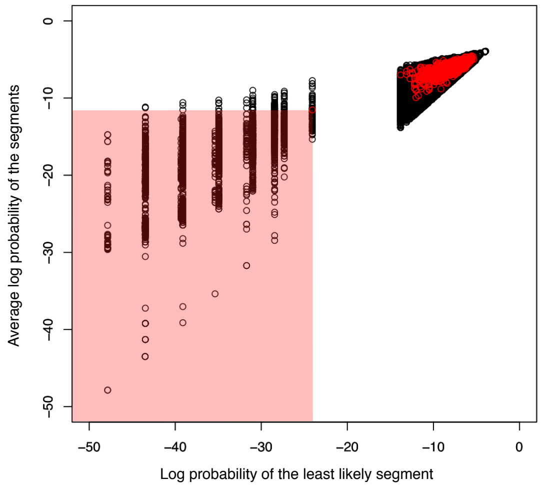

Figure 3: A scatter plot of two features derived from the model scores of the training data set for the English-Russian task. Negative examples were selected from the shaded area.

proach to two other strategies based on those employed in (Jiampojamarn et al., 2010) in the experimental section.

4.3.1 Model-based selection

Figure 3 shows a scatter plot of two plausible features over the En-Ru training data set. The first feature (vertical axis) is the arithmetic mean of the log-probabilities of each of the segments. This averaging allows sequences of differing lengths to be compared. The second feature (horizontal axis) is the log-probability of the least probable segment in the sequence. As can be seen from the plot, the second feature in particular partitions the data set quite cleanly into two clusters, 99.9% of the seed data (plotted on the graph in a lighter shade (red)) lie in the upper right-hand cluster.

We select negative examples, by means of thresholds on these features. The thresholds used to gather negative examples were set using the seed data by choosing the lowest values of any seed data points as the thresholds. This process is illustrated visually in Figure 3; the negative samples being extracted from the lower-left cluster (in the shaded area of the graph). The thresholds used for all language pairs are given in Table 2, together with the number of negative examples that were collected. We used these negative samples together with the provided seed sentences (known to be positive examples) to train an SVM classifier [1].

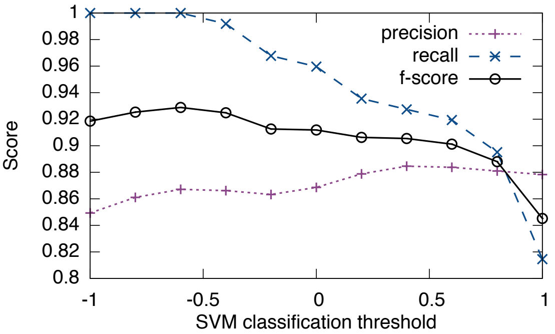

Figure 4: A graph showing the trade-off between precision and recall and its effect on the F-score for the English-Russian task.

4.3.2 Other approaches

Following (Jiampojamarn et al., 2010) we investigated two other methods of generating negative examples. These methods create a large set of incorrect candidates by pairing each source sequence in the seed data, with every target sequence except the correct target. In the first method of selecting negative examples, this large set of candidates is reduced to a smaller set by filtering out those candidates in which the source and target sequences are not phonetically similar. Phonetic similarity being measured as using the longest common subsequence ratio (LCSR) of the romanized forms. In our experiments we adjusted this threshold so that the same number of negative samples were generated in each case (10,000 samples). This approach generates negative examples that are similar to the positive examples, and it can be argued this is advantageous for training a discriminator.

The second approach simply takes a random sample from the large set of candidates. This approach generates samples that more closely approximate the similarity of examples in the real data. Results using each of these methods and also our model-based approach are shown in Figure 5.

4.4 Results

Figure 5 presents the results of our main experiment. Since the mixture of positive and negative examples in the test data is not known *a priori*, we provide results from our system for a range of values of the classification threshold on the output of the SVM. This gives precision/recall curves for each of the strategies for generating negative examples: our proposed approach, the approach based on LCSR, and the approach based on ran-

[1] In these experiments we used the publicly available SVM-lite classifier http://svmlight.joachims.org

	En-Ar	En-Ch	En-Hi	En-Ja	En-Ru	En-Ta
Average	-11.35	-21.05	-7.55	-12.44	-7.9	-7.552
Minimum	-38.34	-42.11	-34.09	-38.28	-13.8	-34.09
Number	831	2061	890	160	9000	450

Table 2: Thresholds on each of the two features used (Average and Minimum segment probability) to obtain the negative examples for each language pair, together with the number of negative examples extracted at these thresholds.

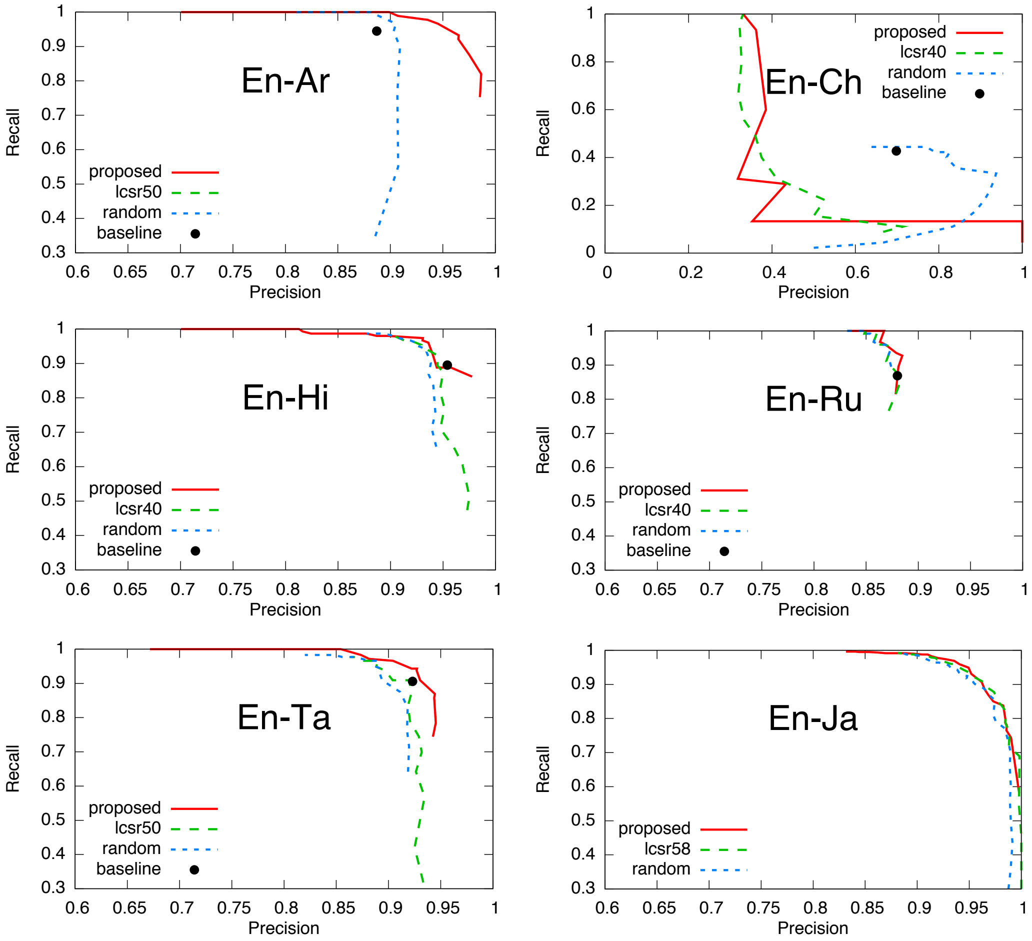

Figure 5: The precision and recall of our proposed method for all language pairs.

dom sampling. The precision/recall/f-score tradeoff for En-Ru is shown in Figure 4. For the baseline we plot a point for the precision and recall of the top-ranked system in the NEWS2010 transliteration shared task to represent the current state of the art. The graph on the bottom right shows similar results on a new English-Japanese task that we constructed in a similar manner to the NEWS workshop tasks.

It is clear that the results on English to Chinese are anomalous. The results on this task were very dependent on the strategy for choosing negative examples and only the random sampling technique was effective. The English-Chinese task differs from the other language pairs in two important respects. Firstly, in the data as supplied for the task there is no segmentation information on the Chinese side, other languages contained word boundaries. We would not expect this to pose problems for our technique which performs unsupervised segmentation of both source and target during the alignment process. The second respect in which this language pair differs is that the grapheme vocabulary size is much larger for Chinese than for the other languages. We believe this is the cause of the anomalous result, and that the larger vocabulary size requires a larger amount of training data to build models that can function effectively. Choosing similar examples, by using the prosed technique or the technique based on LCSR, will reduce the variety of kanji seen in the negative examples, and this could handicap the models where the data size is too small.

On all the other language pairs, our proposed strategy for selecting negative examples performs as well as, or better than the other strategies. Of the other two strategies, the method based on LCSR is generally the the better approach. Moreover, our results show that our system is able to offer performance comparable to the state-of-the-art baseline systems on these language pairs. For the English-Arabic and English-Tamil tasks in particular, our strategy for selecting negative examples offers higher scores in terms of both precision and recall than the other strategies. Our approach typically makes errors on sequence pairs that are genuine but contain novel sub-sequences of graphemes for which our model has no corresponding sequence pair. Feature $f3$ in our model was designed to address this issue by balancing evidence from the lengths of the 'bad' segments in the pairs against evidence from the lengths of the 'good'. The idea being that an unobserved sequence pair within a much larger context of observed sequence pairs is likely to be a correct but novel alignment, rather than an incorrect alignment. Nonetheless some errors of this type remain, but the frequency of type of error can be expected to decrease with training set size.

We created a new task for our experiments based on English-Japanese data. Text from the titles of Wikipedia inter-language links was used as the data to be mined, and we used a set of English-Katakana pairs from the publicly available EDict dictionary [2] to create the seed data. 4000 pairs of interlanguage links were used, 1000 of which were hand-annotated as correct or incorrect transliteration pairs and used as test data. 1000 seed pairs were selected randomly from the bilingual dictionary. The precision and recall curves for the En-Ja task are shown in Figure 5. The results show that mining Japanese can be performed reasonably easily, relative to the language pairs used in the NEWS2010 tasks. All techniques for choosing negative examples were effective here; our proposed approach and the LCSR approach slightly outperforming random sampling. The English-Japanese precision/recall indicate that the automatic mining of English-Japanese transliteration pairs should be fruitful. We believe it would be possible to mine English-Japanese pairs at high-levels of precision and recall. In our experiments, for example, close to 100% precision can be achieved whilst still maintaining 70% recall.

5 Conclusion

In this paper we have presented a novel approach to identifying transliteration word pairs for transliteration mining based on features derived from a Bayesian process that simultaneously co-segments and force-aligns grapheme sequences within the words. Our approach is simple and symmetrical with respect to the two languages involved, and will operate on grapheme sequences in the native scripts of the languages involved. It is not dependent on the existence of a method for romanizing either language. Furthermore, our method performs automatic co-segmentation of both source and target sequences, eliminating any requirement for language specific segmentation schemes.

We evaluated our approach on all of the transliteration mining tracks of the NEWS2010 Named Entity Workshop shared task. Our system in spite of its simplicity, achieved performance comparable to the state of the art systems on this task, indicating the features derived from the Bayesian forced alignment are strongly predictive in classifying transliteration pairs. This paper also contributes a new set of results on an English-Japanese data set we constructed in a similar manner to the NEWS workshop datasets. Our results indicate that mining English-Japanese transliteration pairs should be possible at high levels of precision and recall using the techniques proposed in this paper.

In future research we would like to extend the scope of our work to integrate it into a broader framework to be used for mining named entity pairs (including but not limited to transliteration pairs) that will be used to improve a named entity translation system, and integrate this into an end-to-end machine translation system. In addition we intend to enhance the Bayesian model used to align the grapheme sequences.

[2] http://www.csse.monash.edu.au/~jwb/edict_doc.html

References

Slaven Bilac and Hozumi Tanaka. 2005. Extracting transliteration pairs from comparable corpora. In *In Proceedings of the Annual Meeting of the Natural Language Processing Society*, Japan.

Eric Brill, Gary Kacmarcik, and Chris Brockett. 2001. Automatically harvesting katakana-english term pairs from search engine query logs.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. In *Machine Learning*, pages 273–297.

Kareem Darwish. 2010. Transliteration mining with phonetic conflation and iterative training. In *Proceedings of the 2010 Named Entities Workshop*, pages 53–56, Uppsala, Sweden, July. Association for Computational Linguistics.

Andrew Finch and Eiichiro Sumita. 2010. A Bayesian Model of Bilingual Segmentation for Transliteration. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 259–266.

Yun Huang, Min Zhang, and Chew Lim Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In *ACL (Short Papers)*, pages 534–539.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47, Uppsala, Sweden, July. Association for Computational Linguistics.

Mitesh Khapra, Raghavendra Udupa, A. Kumaran, and Pushpak Bhattacharyya. 2010. Pr + rq □ pq: Transliteration mining using bridge language.

A Kumaran and Haizhou Li, editors. 2010. *Proceedings of the 2010 Named Entities Workshop*. Association for Computational Linguistics, Uppsala, Sweden, July.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010a. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28, Uppsala, Sweden, July. Association for Computational Linguistics.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010b. Whitepaper of news 2010 shared task on transliteration mining. In *Proceedings of the 2010 Named Entities Workshop*, pages 29–38, Uppsala, Sweden, July. Association for Computational Linguistics.

Chun-Jen Lee and Jason S. Chang. 2003a. Acquisition of english-chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 96–103, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chun-Jen Lee and Jason S. Chang. 2003b. Acquisition of english-chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond - Volume 3*, HLT-NAACL-PARALLEL '03, pages 96–103, Stroudsburg, PA, USA. Association for Computational Linguistics.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 159, Morristown, NJ, USA. Association for Computational Linguistics.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 100–108, Morristown, NJ, USA. Association for Computational Linguistics.

Peter Nabende. 2010. Mining transliterations from wikipedia using pair hmms. In *Proceedings of the 2010 Named Entities Workshop*, pages 76–80, Uppsala, Sweden, July. Association for Computational Linguistics.

Sara Noeman and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework. In *Proceedings of the 2010 Named Entities Workshop*, pages 57–61, Uppsala, Sweden, July. Association for Computational Linguistics.

Jong-Hoon Oh and Hitoshi Isahara. 2006. Mining the web for transliteration lexicons: Joint-validation approach. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 254–261, Washington, DC, USA. IEEE Computer Society.

Keita Tsuji and Kyo Kageura. 2006. Automatic generation of japanese□english bilingual thesauri based on bilingual corpora. *J. Am. Soc. Inf. Sci. Technol.*, 57:891–906, May.