4-3 A Transliteration System Based on Bayesian Alignment and its Human Evaluation within a Machine Translation System

Andrew Finch and YASUDA Keiji

This paper reports on contributions in two areas. Firstly, we present a novel Bayesian model for unsupervised bilingual character sequence alignment of corpora for transliteration. The system is based on a Dirichlet process model trained using Bayesian inference through blocked Gibbs sampling implemented using an efficient forward filtering/backward sampling dynamic programming algorithm. The Bayesian approach is able to overcome the overfitting problem inherent in maximum likelihood training. We demonstrate the effectiveness of our Bayesian alignment by using it to build models for phrase-based statistical machine transliteration (SMT) systems. We compare our alignment technique to the commonly used GIZA++ word alignment process, and also to the state-of-the-art m2m bilingual aligner by using their alignments to train transliteration generation systems. In both cases the model resulting from our Bayesian alignment was considerably smaller than competitive technique, and in addition gave an increase in transliteration generation quality. Our second contribution is to conduct a large-scale real-world evaluation of the effectiveness of integrating an automatic transliteration system with a machine translation system. A human evaluation is usually preferable to an automatic evaluation, and in the case of this evaluation especially so, since the common machine translation evaluation methods are often being biassed towards translations in terms of their length rather than the information they convey. We evaluate our transliteration system on data collected in field experiments conducted all over Japan. Our results conclusively show that using a transliteration system can improve machine translation quality when translating unknown words.

Keywords
Transliteration, Human evaluation, Machine translation, Dirichlet process model, Bayesian alignment

1 Introduction

Machine translation has to handle wide variety of proper nouns to achieve high translation performance. However, it is impossible to discover all of these words from limited quantities of bilingual training data, or even to manually maintain a bilingual dictionary because of the huge number of distinct proper nouns in the real world including new words and expressions being created day by day. Therefore, machine translation is a promising area for the application of transliteration technology. In this paper we present a Bayesian technique for building transliteration models that avoids the overfitting issues commonly encountered with non-Bayesian techniques. We evaluate this approach automatically with respect to prior work, but more importantly we evaluate the worth of our approach in a real-world translation system.

Automatic machine translation metrics, for example the BLEU score [1], and the NIST score [2] often have a natural bias towards shorter translations. That is to say, it is often a better strategy for a machine translation system to output nothing or a very short erroneous translation than to output a long, incorrect translation. Attempts to correct this length bias are included within these metrics, but unfortunately they are not completely effective and some degree of bias remains. As evidence of this phenomenon, it is common practice in competitive machine translation evaluation campaigns for the systems to delete untranslated unknown words from their machine translation output, rather than keep them or attempt to transliterate them (for example, [3]). For this reason it is important to study the effect of introducing transliteration into a machine translation system through a human evaluation experiment, even though such experiments are expensive in terms of human effort.

Throughout this paper we will refer to graphemes, grapheme sequences and grapheme sequence pairs. By grapheme, we mean a single unicode character, for example ‘a’ in English, ‘ア’ in Japanese. Grapheme sequences are arbitrary sequences of these graphemes, and grapheme sequence pairs are 2-tuples of grapheme sequences, each element in the tuple being a grapheme sequence in a given language; for example: (‘hello’,‘ハロー’). We now move on to motivate the Bayesian alignment scheme using in training our transliteration model.

1.1 Motivation

It is possible to couch the problem of transliteration as a problem of direct grapheme to grapheme transduction that proceeds in a monotone order. Recently systems based on phrase-based statistical machine translation technology [4]-[6] and the joint source-channel model [7] are being actively researched and have achieved state-of-the-art performance on this task. This type of approach makes no linguistic assumptions about the data and no intermediate phonetic representation is required, because the transduction is directly from grapheme to grapheme. The advantages of this type of approach are that the only training corpus required is a set of bilingual word pairs, and the approach can be applied directly to a wide range of language pairs without the need to develop a set of linguistically-motivated heuristics specific to the languages involved. In this paper we focus on translation methodologies that use grapheme sequence pairs as their basis for generation, namely transliteration using phrase-based statistical machine translation (PBSMT) techniques, and the joint-source channel approach.

At the core of the PBSMT approaches is the phrase-table. In transliteration, this is a set of bilingual grapheme sequence pairs that are concatenated to generate the transliteration. The creation of a phrase-table during a typical training procedure for a PBSMT system consists of the following steps:

1. Word alignment using GIZA++ [8]
2. Phrase-pair extraction using heuristics (for example *grow-diag-final-and* from the MOSES [9] toolkit)

This approach works very well in practice, but a more elegant solution would be to arrive at a set of bilingual grapheme sequence-pairs (we use this term to describe the analogue of the phrase-pair at the grapheme level) in one step, from a generative model. Unfortunately, when traditional methods that use the Expectation and Maximization (EM) algorithm to maximize likelihood are applied to the task, they produce solutions that can grossly over-fit the data. As an extreme example, the most likely alignment of a corpus into sequence-pairs, assuming no limits on sequence-pair size would be the entire corpus as a single bilingual sequence-pair, holding all the probability mass.

GIZA++ mitigates this problem by aligning the words in a one-to-many fashion. The single word on one side of the alignment acts as a constraint on the size of the bilingual pairs. A similar approach can be taken in transliteration, where a single character in one language is permitted to align to multiple characters of the other, but not vice versa. This approach is reasonable for English-Chinese transliteration [7][10], where one Chinese character can be assumed to map to several English characters.

In GIZA++ this one-to-many alignment is done twice: from both source-to-target and also from target-to-source. A table of word-to-word alignments is then constructed from (typically the intersection) both of these alignments. Additional word alignments that are not in the intersection are added based on evidence and heuristics, and finally all possible phrase-pairs are extracted from the table of alignments that are consistent with the table.

In [11][12] many-to-many alignment is performed directly using maximum likelihood training, but evidence trimming heuristics that exclude part of the available training data are required to prevent the models from overfitting the data. [13] have successfully applied a similar Bayesian technique to grammar induction and [14][15] have developed tractable Bayesian methods for the more complex task of bilingual phrase pair extraction for SMT, which involves reordering. [16] tackle the overfitting problem in phrasal alignment by using a leave-one-out approach using a strategy that despite being a different paradigm, shares many of the characteristics of our approach. [17] have also developed a Bayesian adaptor grammar approach to alignment for transliteraion.

In this paper we extend existing monolingual word segmentation models ([18][19]) to bilingual alignment, and provide a simple yet elegant way to directly align a bilingual training corpus in a many-to-many fashion without overfitting, using a Bayesian model in the manner of [20].

This paper is organized into 2 parts: in the first part we describe the Bayesian alignment technique, and present experiments to explore its characteristics and evaluate its performance relative to other state-of-the-art approaches. In the second part we present a human evaluation of the usefulness of using transliterations to handle out-of-vocabulary words in an industrial machine translation system. More specifically, in Section **2** we describe the Bayesian model used in our transliteration system. Here we give an overview of the Dirichlet process model, the Chinese Restaurant process and explain how our model relates to these two representations. In Section **3** we describe the blocked Gibbs sampling technique used to train the model. In Section **4** we describe the experiments we performed to evaluate our model with respect to the GIZA++ alignment technique commonly used in phrase-extraction for machine translation. Section **5** presents experiments and analysis of our alignment technique when compared to the m2m aligner, a state-of-the-art many-to-many sequence alignment tool. Section **6** contains our experiments to use human evaluation to assess the effectiveness of using transliterations in machine translation output. Finally, in Section **7** we conclude and suggested promising avenues for future research.

2 Methodology

Recently in the natural language processing field Bayesian models have been proposed to tackle a variety of problems, and have been found to be particularly effective in word segmentation [18][19]. The model we use in this paper is a unigram Dirichlet process model. Using this approach to perform bilingual alignment for the general case of machine translation with re-ordering would be a challenging undertaking, however for transliteration where the sequence lengths are short and under the assumption that there is no reordering, it is feasible to tackle the bilingual alignment problem directly without the need for specialized optimization or annealing (we do use a block sampling algorithm, and a dynamic programming algorithm).

The next section introduces the joint-source channel model. We use this model to define the generative process of the Dirichlet process model outline in Subsection **2.2**.

2.1 Joint source-channel model

Let us assume we are given a bilingual corpus consisting of a source sequence $\mathbf{s}_1^M = <s_1, s_2, \ldots, s_M>$ and a target sequence $\mathbf{t}_1^N = <t_1, t_2, \ldots, t_N>$ We distinguish sequences of characters from single characters by using a boldface font with an overbar.

We adopt the joint source-channel model of [7] as the underlying generative model, and we make the additional assumption that the segments are independent of each other (our approach can easily be extended to model these dependencies at the expense of some additional complexity, see [19]). Under this model, the corpus is generated through the concatenation of *bilingual sequence-pairs* (we will use this term repeated throughout this paper to refer to corresponding sequences of source and target graphemes, as defined below).

A bilingual sequence-pair is a tuple $(\mathbf{s}, \mathbf{t})$ consisting of a sequence of source graphemes together with a sequence of target graphemes $(\mathbf{s}, \mathbf{t}) = (<s_1, s_2, \ldots s_i>, <t_1, t_2, \ldots, t_j>)$.

The corpus probability is simply the probability of all possible derivations of the corpus given the set of bilingual sequence-pairs and their probabilities.

$$\begin{aligned} p(\mathbf{s}_1^M, \mathbf{t}_1^N) &= p(s_1, s_2, \ldots, s_M, t_1, t_2, \ldots, t_N) \\ &= \sum_{\gamma \in \Gamma} p(\gamma) \end{aligned} \tag{1}$$

where $\gamma = ((\mathbf{s}_1, \mathbf{t}_1), \ldots, (\mathbf{s}_k, \mathbf{t}_k), \ldots, (\mathbf{s}_K, \mathbf{t}_K))$ is a derivation of the corpus characterized by its alignment, and Γ is the set of all derivations (alignments) of the corpus.

The probability of a single derivation is given by the product of its component bilingual sequence-pairs:

$$p(\gamma) = \prod_{k=1}^{K} p((\mathbf{s}_k, \mathbf{t}_k)) \tag{2}$$

The corpus for our experiments is segmented into bilingual word-pairs. We therefore constrain our model such that both source and target character sequences of each bilingual sequence-pair in the derivation of the corpus are not allowed to cross a word segmentation boundary. Equation 2 can therefore be arranged as a product of word-pair w derivations of the sequence of all word-pairsw $\mathcal{W}$ in the corpus.

$$p(\gamma) = \prod_{w \in \mathcal{W}} \prod_{(\mathbf{s}_k, \mathbf{t}_k) \in \gamma_w} p((\mathbf{s}_k, \mathbf{t}_k)) \tag{3}$$

where γ_w is a derivation of bilingual word-pair w.

2.2 Dirichlet process model

A Dirichlet process is 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.

The Dirichlet process model we use in our approach is a simple model that resembles the cache models used in language modeling [21]. Intuitively, the 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. Ideally, to encourage the re-use of model parameters, the probability of generating a novel bilingual sequence-pair should be considerably lower then the probability of generating a previously observed sequence pair. This is a characteristic of the Dirichlet process model we use and furthermore, the model has a preference to generate new sequence-pairs early on in the process, but is much less likely to do so later on. In this way, as the cache becomes more and more reliable and complete, so the model prefers to use it rather than generate novel sequence-pairs. The probability distribution over these bilingual sequence-pairs (including an infinite number of unseen pairs) can be learned directly from unlabeled data by Bayesian inference of the hidden alignment of the corpus. The ability of the model to assign a probability to any unseen sequence-pair gives the technique the ability to score candidate training data.

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

$$
\begin{aligned}
G|_{\alpha,G_0} &\sim DP(\alpha, G_0) \\
(\mathbf{s}_k, \mathbf{t}_k)|G &\sim G
\end{aligned}
\tag{4}
$$

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.

2.2.1 The Chinese restaurant process

Unfortunately it is not possible to estimate G directly, since there are an infinite number of possible bilingual sequence-pairs, so instead we integrate over its possible values. To do this we cast the bilingual sequence-pair generation process as an instance of the Chinese Restaurant Process (CRP) [22]. According to this representation, every bilingual sequence-pair corresponds to the dish served at its table in a potentially infinite set of tables in a Chinese restaurant. The number of customers seated at each table represents the cumulative count of the bilingual sequence-pair. A new customer to the restaurant can take a seat at an occupied table with a probability proportional to the number of customers at that table, and must eat that table's dish, or can take a seat at an unoccupied table with a probability proportional to a constant, in which case they must eat a dish (a bilingual sequence-pair) chosen by the chef (in this analogy the chef's choice is in accordance with the base distribution G_0).

2.2.2 The base measure

For the *base measure* that controls the generation of novel sequence-pairs, we use a joint spelling model that assigns probability to new sequence-pairs according to the following joint distribution:

$$
\begin{aligned}
G_0((\mathbf{s},\mathbf{t})) &= p(|\mathbf{s}|)p(\mathbf{s}||\mathbf{s}|) \times p(|\mathbf{t}|)p(\mathbf{t}||\mathbf{t}|) \\
&\simeq \frac{\lambda_s^{|\mathbf{s}|}}{|\mathbf{s}|!} e^{-\lambda_s} v_s^{-|\mathbf{s}|} \times \frac{\lambda_t^{|\mathbf{t}|}}{|\mathbf{t}|!} e^{-\lambda_t} v_t^{-|\mathbf{t}|}
\end{aligned}
\tag{5}
$$

where $|\mathbf{s}|$ and $|\mathbf{t}|$ are the length in characters of the source and target sides of the bilingual sequence-pair; v_s and v_t are that vocabulary (alphabet) sizes of the source and target languages respectively; and λ_s and λ_t are the expected lengths of source and target.

According to this model, source and target sequences are generated independently: in each case the sequence length is chosen from a Poisson distribution, and then the sequence itself is generated given the length. Note that this model is able to assign a probability to arbitrary bilingual sequence-pairs of any length in source and target sequence, but favors shorter sequences in both.

More sophisticated methods of defining the base measure are possible, for example [14, 15] use the IBM model 1 likelihood of one phrase conditioned on the other in the base model to encourage the formation of bilingual pairs that follow the word alignments in the corpus. This idea can be transferred to the character level and used in our approach, but remains as future work at this point.

Following [18] we assign the parameters λ_s, λ_t and α, the values 2, 2 and 0.3 respectively. Ideally these parameters should be learned from the data, however in our experiments the settings were sufficient to give a useful alignment of the training corpus. Moreover, the system proved to be insensitive to changes in these parameters in a set of pilot experiments, converging to very similar final iteration samples for a range of parameter settings.

2.2.3 The generative model

The generative model is given in Equation 6 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 in the history so far $(\mathbf{s}_{-k}, \mathbf{t}_{-k})$. Here *-k* is read as: "up to but not including *k*".

$$
p((\mathbf{s}_k,\mathbf{t}_k))|(\mathbf{s}_{-k},\mathbf{t}_{-k})) = \frac{N((\mathbf{s}_k,\mathbf{t}_k)) + \alpha G_0((\mathbf{s}_k,\mathbf{t}_k))}{N+\alpha}
\tag{6}
$$

In this equation, N is the total number of bilingual sequence-pairs generated so far (the number of customers 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 (the number of people seated at its table). G_0 and α are the base measure and concentration parameter as before.

3 Bayesian inference

3.1 Gibbs sampling

We used a blocked version of a Gibbs sampler for training. In [21] they report issues with mixing in the sampler that were overcome using annealing. In [19] this issue was overcome by using a blocked sampler together with a dynamic programming approach. Our algorithm is similar to that of [19], and we found our sampler converged rapidly without annealing (see Fig. 5). The number of iterations was set by hand after observing the convergence behavior of the algorithm in pilot experiments. We used a value of 30 iterations through the corpus in all our experiments.

The sampling algorithm is shown in Fig. 1 and the iterative component proceeds as follows. Firstly the training set of bilingual word-pairs is permuted randomly, and a bilingual word-pair is sampled from this permutation without replacement. Secondly, a probability distribution over all possible alignments of the chosen bilingual word-pair is calculated by obtaining probabilities with respect to a model that does not include the bilingual word-pair, its previous alignment information and respective counts. Due to the short sequence lengths involved in transliteration, it is possible to use a brute force approach to calculate this distribution, however for efficiency we extended the forward filtering/backward sampling (FFBS) dynamic programming algorithm of [19] to deal with bilingual alignment. We implemented this algorithm graphically as explained below.

We use a alignment graph (shown in Fig. 2) to guide the process. This directed graph is a compact representation of all possible ways in which to align a bilingual pair. Each node represents a set of partial alignment hypotheses of the whole sequence that share the same sequences of source and target tokens, and each arc represents the bilingual phrase pair used to transition from the tail of the arc to the head. In the figure the arcs are labelled with the log-probability of this sequence-pair (given by the model in Equation 6), therefore the log-probability of a full alignment hypothesis is given by the sum of the arc labels on the respective path from the source node '<s>' to the sink node 'abba'. The most probable alignment is indicated with bold arcs in the figure and corresponds to the alignment 'a b ba', this is reasonable since both 'a/A' and 'ba/BA' are associated with their phonetic equivalents in Japanese, and the Japanese 'TSU' indicates that the consonant immediately to the right is to be repeated. The least probable alignment in the graph is given by 'abb/A a/TSU-BA'. The log-probabilities in the graph are real values taken from the third iteration of the training, and here the most probable alignment is already by far the most likely.

Nodes in the graph can have multiple in- and out-degree. Two nodes are combined

```
Input: Random initial corpus alignment
Output: Unsupervised alignment of the corpus according to the model
foreach iter=1 to NumIterations do
    foreach bilingual word-pair $w \in randperm(\mathcal{W})$ do
        foreach alignment $\gamma_i$ of $w$ do
            Compute probability $p(\gamma_i|h)$
            where $h$ is the set of data (excluding $w$) and its hidden alignment
        end
        Sample an alignment $\gamma_i$ from the distribution $p(\gamma_i|h)$
        Update counts
    end
end
```

Fig.1 The blocked Gibbs sampling algorithm

when the unaligned part of the bilingual sequence pair is the same for both, giving rise to a compact, efficient representation.

The FFBS algorithm operates directly on the alignment graph, and has two steps. The *forward filtering* step, calculates for each node in the graph, the probability of the subgraph (including the node itself) to the left of the node, back to the source node. This probability α, is stored in the node itself (these α's are shown in Fig. 2). This process proceeds recursively in a depth-first post-order traversal of the graph, starting at the sink node. Nodes for which the probability has been calculated are marked as done, ensuring α gets calculated only once for the node.

The *backward sampling* step samples a derivation of the bilingual word pair according to the probability distribution over all possible alignments. This is done easily using the α values stored in the graph by the forward filtering process. The backward sampling also proceeds recursively from the sink node. For each incoming arc, the probability of including that arc in the sample is given by the product of the arc probability and the α value at the tail of the arc. This value is calculated for each incoming arc, and one arc from the set is sampled according to the probability distribution over the arcs. The sampling procedure is called recursively on the tail of the sampled arc until the source node of the graph is reached. The sequence of arcs traversed defines the sampled derivation of the bilingual pair for the current iteration of the training process, and this sample is in accordance with the probability distribution over all derivations with respect to the model.

3.2 Sequence-pair extraction

During the phrase-table generation process of a typical phrase-based SMT system, GIZA++ is run twice to generate alignments at the word level, from source-to-target and from target-to-source. Following this step, the *grow-diag-final-and* procedure is used to extract *all phrases* consistent with the word alignments arising from the two GIZA++ runs. When building a phrase-table from the alignment achieved at final iteration of our Gibbs sampling procedure, we use a much simpler heuristic that is in the same spirit to derive a larger

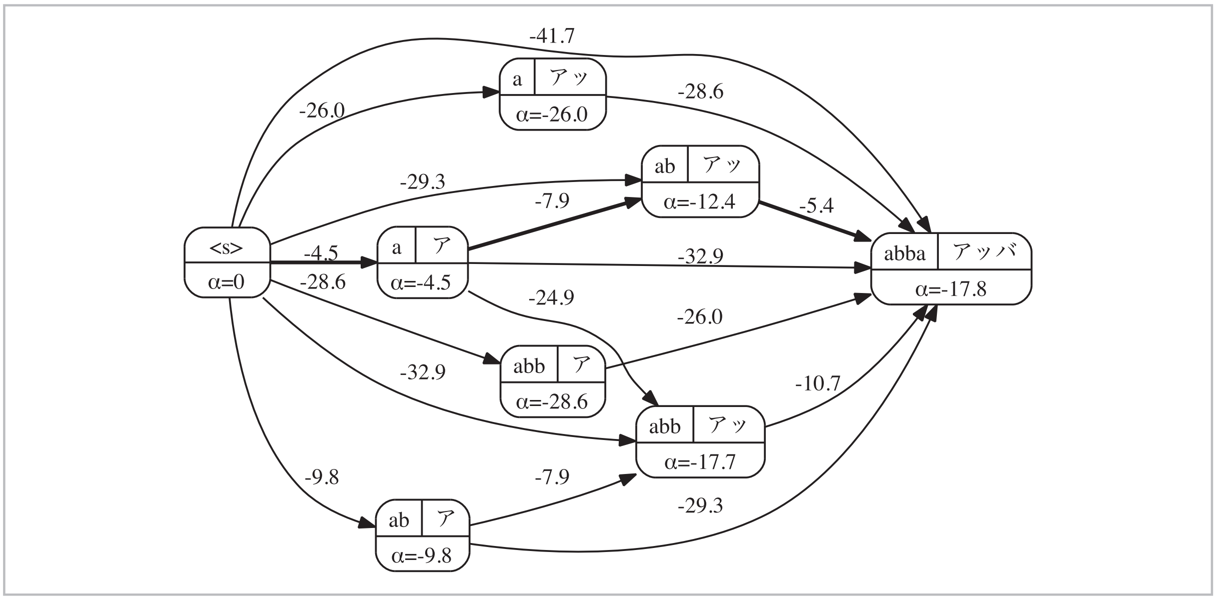

Fig.2 *A graph representing all possible alignments of the character sequences "abba" in English and "A-TSU-BA" in Japanese.*

The α labels on the nodes represent the log-probability of subgraph (including the node itself) to the left of the node. The labels on the arcs are the log-probabilities of bilingual phrase pairs used to transition from tail-to-head, and are given by the model of Equation 6.

set of phrases consistent with the initial alignment. Our experiments show that this is a necessary step that considerably improves system performance.

The algorithm we use for phrasal extraction from the aligned corpus is as follows: within a single bilingual word-pair, agglomerate all contiguous bilingual sequence-pairs in all possible ways, but limit the size of the agglomerated source and target phrases to match the *maximum phrase length* parameter used to train the SMT system (this was set to 7 in our experiments). This is not strictly necessary, but we performed this step to keep the phrase-table generated from our Bayesian alignment comparable to that generated by the baseline system. The algorithm used to perform the sequence-pair agglomeration is outlined in Fig. 3. An example word pair together with its alignment and the set of agglomerated phrases that can be extracted from it is illustrated in Fig. 4.

```
INPUT:     a sequence of bilingual sequence pairs: ARRAY-of-sequence-pairs derivation
OUTPUT:    a set of all pairs formed by agglomerating the pairs in derivation

SET-of-sequence-pairs      agglomerations     = empty
ARRAY-of-sequence-pairs    chunk              = empty

FOR chunk in all subsequences of derivation
{
    sequence-pair pair = concatenation of sequence pairs in chunk

    IF (pair has source sequence length <= MAX_SOURCE_SEQUENCE_LENGTH AND
        pair has target sequence length   <= MAX_TARGET_SEQUENCE_LENGTH)
    {
        INSERT pair into agglomerations
    }
}

PRINT all_agglomerations
```

Fig.3 *The sequence-pair agglomeration algorithm*

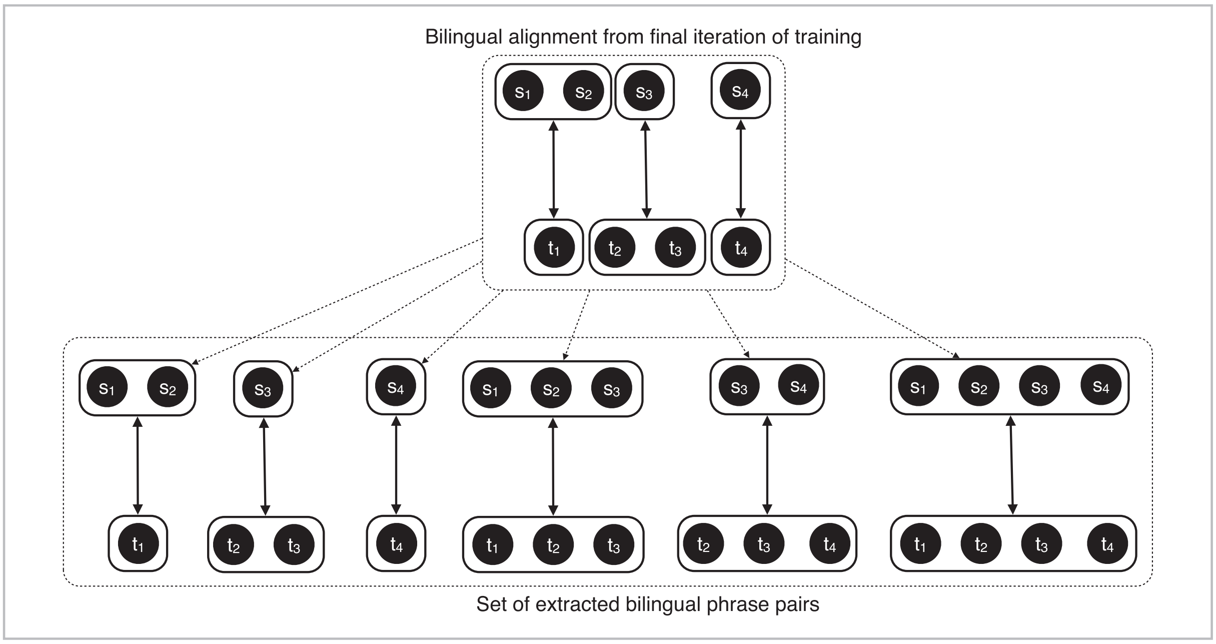
Bilingual alignment from final iteration of training
s1 s2 s3 s4
t1 t2 t3 t4
s1 s2
t1
s3
t2 t3
s4
t4
s1 s2 s3
t1 t2 t3
s3 s4
t2 t3 t4
s1 s2 s3 s4
t1 t2 t3 t4
Set of extracted bilingual phrase pairs

Fig.4 *The sequence-pair extraction process for a single bilingual word pair, using sequence-pair agglomeration*

4 Comparing to GIZA++

4.1 Baseline system

For our experiments we use the phrase-based machine translation techniques introduced by [9], integrating our models within a log-linear framework [23]. Word alignment was performed using GIZA++ [8] and sequence-pair extraction using the MOSES [9] tools. The decoder used was an in-house phrase-based machine translation decoder OCTAVIAN that for these experiments was configured to operate according to the same principles as the publicly available MOSES [9] SMT decoder.

In these experiments 5-gram language models built with Witten-Bell smoothing were used. The system was trained in a standard manner, using a minimum error-rate training (MERT) procedure [24] with respect to the BLEU score on the held-out development data to optimize the log-linear model weights.

Rama and Gali [25], evaluated several techniques for sequence-pair extraction for transliteration and found the *grow-diag-final-and* heuristic to be the most effective, we therefore adopt this method in the baseline system our experiments.

4.1.1 Decoding constraints

The experiments reported in this paper were conducted using a beam width of 100, with no stack thresholding, and a strictly monotone decoding process.

4.2 Experimental data

Our training data consisted of 27,993 bilingual single word-pairs that were used in the NEWS2010 workshop transliteration shared task. The development data consisted of 3,606 bilingual word-pairs drawn from the same sample. The evaluation data consisted of a further 1,935 bilingual word-pairs not contained in the other two data sets. The corpus statistics for the three corpora are given in Table 1.

We used the data to train a phrase-based SMT system to perform transliteration from English to Japanese. We trained our Dirichlet process model on the same parallel data set, and extracted transliteration phrase-tables from the alignment of the corpus at the final iteration (iteration 30).

4.3 Training procedure

For the Gibbs sampling, we chose to start the sampling from a random alignment of the corpus. That is, for each bilingual word-pair in the corpus, a single alignment was sampled from a uniform distribution over all possible alignments of the pair. We believe that it might be advantageous, and certainly more efficient to start the sampling from a more intelligent starting point, for example one derived from a pre-processing pass of GIZA++. However, the training was able to arrive at a good alignment (by visual inspection) of the training corpus, its usefulness being borne out by the experimental results in the next section.

4.4 Evaluation procedure

The results presented in this paper are given in terms of official evaluation metrics used in the NEWS2010 transliteration generation shared task [26]. In our results, ACC refers to the top-1 accuracy score, that measures the percentage of the time the top hypothesis from the system exactly matches the reference.

Table 1 *Statistics of the English-Japanese bilingual corpora*

Corpus	word-pairs	Characters		Avg. Word Len.	
		En	Ja	En	Ja
Training	27993	188941	131275	6.75	4.69
Development	3606	24066	16651	6.67	4.62
Evaluation	1935	11863	8199	6.13	4.24

F-score measures the distance of the best hypothesis from the reference transliteration; the reader is referred to the workshop white-paper [26] for more details. For brevity, we only report our results in terms of ACC and F-score in this paper, but the results in terms of the other NEWS2010 metrics have the same character.

4.5 Results

4.5.1 Training

The convergence of the algorithm during the training procedure is shown in Fig. 5 which plots the log-probability of the sampled derivation at the end of each pass through the training corpus against iteration. It can be seen from the graph that the system rapidly improves from the poor initial alignment, and thereafter continues to gradually improve. The log-probability of the initial random alignment was -1.5e06 and is omitted.

4.5.2 Evaluation using automatic metrics

Our results on the English-to-Japanese transliteration task are summarized in Table 2. It is clear from the table that using sequence-pairs from only the sample at the final iteration of the training produces gave lower performance than the baseline system. The phrase-table derived in this way contained only 3372 sequence-pairs as opposed to over 140,000 in the phrase-table extracted from the GIZA++ alignments. Moreover these sequence-pairs

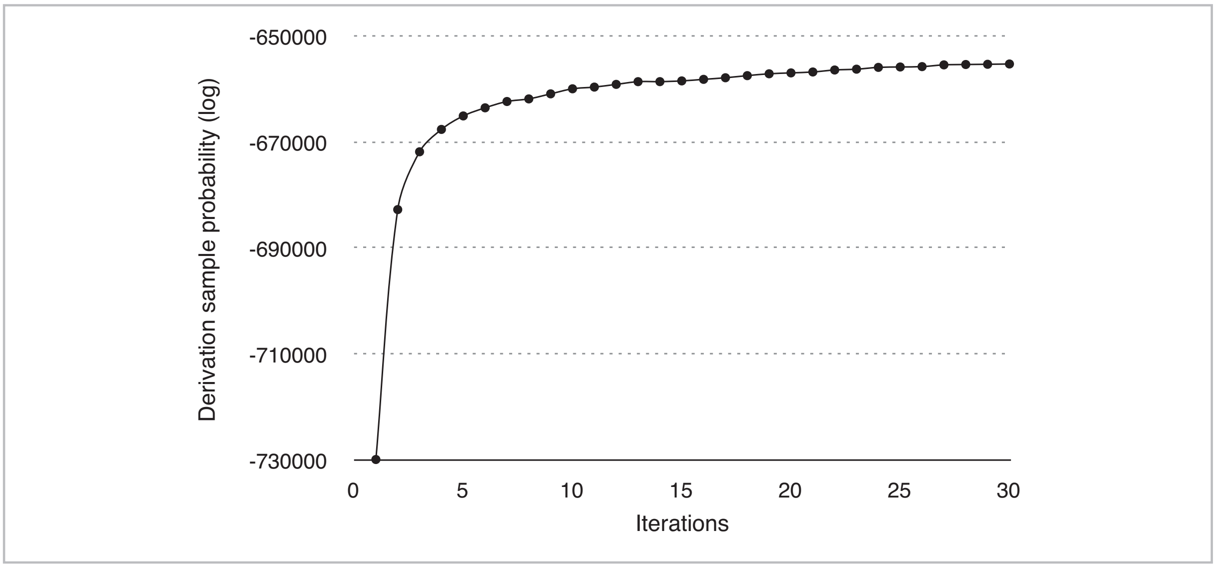

Fig.5 The evolution of the log-probability of the sampled derivation with respect to the training iteration

Table 2 The experimental results for the three systems together with some statistics of their phrasetables

Phrase Extraction Model	ACC	F-score	Phrase-table Entries	Phrase-table Overlap (%)	Avg. Phrase Length	
					En	Ja
GIZA++ and *grow-diag-final-and*	0.313	0.745	143382	100	5.41	4.80
Bayesian Aligner (tuned on baseline phrase-table)	0.278	0.726	3372	2	2.60	2.75
Bayesian Aligner (tuned on Bayesian phrase-table)	0.283	0.732	3372	2	2.60	2.75
Bayesian Aligner (+agglomerated)	0.323	0.748	102507	57	5.54	4.83
Bayesian Aligner (+integrated)	0.329	0.752	164258	100	5.46	4.81

Here +agglomerated means the sequence-pairs were extracted by agglomeration from a single sample at the end of the training. In *+integrated* the phrase-tables from the baseline system and the agglomerated system were linearly interpolated with equal weights. Differences between systems were all found to be significant by paired t-testing at a level of 0.05, except for the ACC scores for the agglomerated and integrated systems.

were short compared to those in the baseline system's phrase-table: approximately 3 characters in both source and target on average, compared to around 5 characters for the baseline system.

When a phrase-table built from agglomerations of the same set of sequence-pairs was used, a much larger phrase-table of around 100,000 phrases resulted, with sequence-pairs that are comparable in size to those of the baseline, around 5 characters. On the transliteration task, this phrase-table gave an improvement of approximately 1% in ACC over the baseline system, from a phrase-table that was about 30% smaller in size. Moreover, since the sequence-pairs are concatenations of 3,372 component sequence-pairs, this model could be stored very compactly if necessary. Further gains were obtained by interpolating the agglomerated model together with the baseline model. We believe this gain may be due to the effect of smoothing.

Our experiments were designed to favor the baseline model since the system was tuned using the MERT procedure with its own phrase-table. It is possible that our proposed system would have obtained a higher score if tuned with its own phrase-table, however we chose not to as this would have introduced additional variance from the differences in the two MERT search processes into the results. However, to verify that the agglomeration step was truly necessary, we also ran an experiment that was tuned with respect the small unagglomerated phrase-table: 'tuned on Bayesian phrase-table' in Fig. 2, rather than the baseline. As expected, performance improved possibly due to more optimal weights for the length models, since the phrases are shorter, but did not improve to the same level as the system that used the agglomerated phrase table.

It is interesting to note that the system's performance was improved dramatically simply by grouping the phrases into larger units. This highlights one of the advantages of the phrase-based translation approach. The agglomerated model, because of the way it was constructed, is not able to generate anything the simpler model cannot, but when larger sequence-pairs are used to build the target sequence the characters in the phrase carry with them the *implicit context* of the other characters in the phrase, all of which have occurred together in the same context in the training corpus. In the model with the unagglomerated sequence-pairs, this role is performed mainly by the language model. In spite of the fact that we used a 5-gram language model the system clearly benefited from a model that contained longer sequence-pairs as the basic translation unit. Looking at the phrase-table overlap figures in Fig. 2 it seems that the process of agglomeration produced a phrase table with a reasonable degree of overlap (57%) with that produced by using GIZA++ and the MOSES phrase extraction procedure.

In a final experiment we collected counts for the sequence-pairs over multiple iterations of the training process: from iteration 5 (the *burn-in*) to the final iteration. This resulted in a 37% larger phrase-table, but surprisingly did not realize any notable improvement in performance.

4.6 Decoding consistency

We ran an experiment to investigate the reasons for the improvements in system performance. Our hypothesis was that the Bayesian system had produced a phrase table that led to a more consistent decoding process. This was based on the belief that the fact that the Dirichlet process model strongly encourages reuse of the bilingual sequence-pairs it discovers. This should result in a more compact phrase-table, and should entail that similar words in the corpus are likely to be decoded in more homogenous fashion. To test the hypothesis we modified the machine translation decoder to count the number of *types* of bilingual sequence-pair used to decode the evaluation data, and re-ran the English-Japanese transliteration experiment that showed the largest gain in performance. We found that the decoding process that used the phrase-table generated from our Bayesian model (with agglomerated sequence-pairs) used a total of 3,496 unique sequence-pairs, whereas decoding using the phrase-table extracted using GIZA++ and *grow-diag-final-and* required a total of 3,970 phrase pairs during the decoding process, supporting our hypothesis. The 3,496 sequence-pairs from the Bayesian model's phrase-table, could be further analysed into 1,289 component bilingual pairs that were present in the alignment in the sample taken at the end of the training process.

4.7 Training times

Bayesian methods are often criticized for their slow performance. The current implementation will require optimization to enable it to handle long sequences, but for transliteration data, where the sequence length is short, the process is practicable. As an example, the full training process completed on the Japanese-English NEWS2010 training data set in 15 minutes, averaging around 30 seconds for each iteration over the data.

5 Comparison to the m2m aligner

In this experiment we compare our Bayesian aligner to a similar state-of-the-art alignment tool that is capable of many-to-many alignments: the publicly available m2m alignment tool [*1] [27] that is trained using the EM algorithm and is based on the principles set out in [28].

The alignments from both methodologies were used in the same way to build the joint-source channel models for a transliteration generation system built using the same techniques as the NICT entry into the NEWS2011 shared evaluation [29]. The experiments were run in the same way using the same script, the only difference being the choice of aligner used. We used data from the 2009 NEWS workshop for our experiments, and evaluated using the F-score metric used for the shared task evaluation. This system was developed from the transliteration generation system used in the experiments in Section **4**, and differs in one important repect: it incorporates a joint-source channel model directly into the decoding process. The aligners were run with their default settings, and with the same limits for source and target segment size. It may have been possible to obtain better performance from the aligners by adjusting specific parameters, but no attempt was made to do this. The results are shown in Table 3. In all experiments, the Bayesian aligner gave the best performance, and the largest improvement was on language pairs that have large grapheme set sizes on the target side. The grapheme set size is shown in Table 3 in the 'Target Types' column. The source grapheme set sizes were very similar and small (around 27) for all experiments, as the source language was either English or in the case of Jn-Jk, a romanized form of Japanese. Looking at the N-gram statistics in Table 3, for languages with large grapheme sets the number of unigrams in the Bayesian model is less than half that used by

[*1] http://code.google.com/p/m2m-aligner/

Table 3 *System performance in terms of F-score, by using alternative alignment schemes together with statistics relating to be number of parameters in the models derived from the alignments*

Language Pairs	Target Types	m2m F-score	Bayesian F-score	m2m			Bayesian		
				1-grams	2-grams	3-grams	1-grams	2-grams	3-grams
En-Ch	372	0.858	0.880	9379	44003	75513	4706	38647	72905
En-Hi	84	0.874	0.884	3114	15209	30195	1867	20218	34657
En-Ko	687	0.623	0.651	4337	11891	14112	2968	11233	14729
En-Ru	66	0.919	0.922	1638	6351	14869	1105	12607	23250
En-Ta	64	0.885	0.892	2852	14696	27869	1561	17195	30244
Jn-Jk	1514	0.669	0.767	7942	27286	38365	3532	22717	37560

the m2m model. Learning a compact model is one of the signature characteristics of the Bayesian model we use; adding a new parameter to the model is extremely costly, and the algorithm will therefore strongly prefer to learn a model in which the parameters are re-used.

Initially we considered the hypothesis that the difference in performance between these two approaches came from differences in the sparseness of the language models. Surprisingly however, the numbers of bi-grams and tri-grams in the joint language models are quite similar.

Another explanation is that the smaller number of unigrams indicates that the alignment is more self-consistent and therefore makes the generation task less ambiguous. This is supported by looking at the development set perplexity. On the Jn-Jk task where the differences between the systems are the largest, we found that a joint language model trained on the Bayesian alignment had 1-, 2-, and 3-gram perplexities of 218.3, 88.4 and 87.5 respectively, whereas the corresponding m2m model's perplexities were 321.8, 120.5 and 119.3. The number of segments used to align the corpus was the same for both systems in this experiment.

Table 4 gives an example from the data of the differences in alignment consistency. The Bayesian alignment is strongly self-consistent. The source sequence 'ara' has been aligned identically as a single unit in all cases. The m2m system also shows self-consistency, but uses a few different strategies to align the start of the sequence. Interestingly the Bayesian method in this example has aligned according to the correct linguistic readings of the kanji. This hypothesis has empirical support from the experiments reported in [29].

6 Human evaluation using speech-to-speech translation field data

In this section, we report the results from experiments in machine translation carried out to evaluate the effectiveness of our transliteration method with real-world data collected from the application of mobile translation devices in the field.

The test set for this evaluation was extract-

Table 4 *Example alignments from the m2m aligner and the Bayesian aligner, taken from a long contiguous section of the training set where both techniques disagree on the alignment*

m2m			Bayesian		
arad→荒	a→田		ara→荒	da→田	
ar→新	ae→江		ara→新	e→江	
ar→荒	ahori→堀		ara→荒	hori→堀	
ar→新	ai→井		ara→新	i→井	
ar→新	ai→居		ara→新	i→居	
ar→荒	ai→井		ara→荒	i→井	
ar→荒	ai→居		ara→荒	i→居	
araj→荒	ima→島		ara→荒	jima→島	
arak→新	i→木		ara→新	ki→木	
arak→荒	i→木		ara→荒	ki→木	
ar→荒	akid→木	a→田	ara→荒	ki→木	da→田
ar→荒	ao→尾		ara→荒	o→尾	
ar→荒	ao→生		ara→荒	o→生	
ar→荒	aoka→岡		ara→荒	oka→岡	
arasa→荒	wa→沢		ara→荒	sawa→沢	
ar→荒	aseki→関		ara→荒	seki→関	

ed from user log data of a set of speech-to-speech translation field experiments that occurred in the fiscal year 2009 [30]. As shown in Fig. 6, the field experiments were undertaken in nationwide in Japan in five broad regions of Japan: Kanto, Kansai, Kyushu, Hokkaido and Chubu. The field experiments were undertaken as part of the Ministry of Internal Affairs and Communications initiative titled "Field Testing of Automatic Speech Translation Technology and its Contribution to Local Tourism." We sampled 100 sentences the from manually transcribed user log data of the field experiments from each of the five areas. Then we selected only those sentences that contained untranslatable named entities, leaving a total of 74 sentences. All of the test sentences contained at least one proper noun written in *hiragana* or *kanji* characters.

The translation direction of the evaluation is from Japanese into English. As a baseline system we used a standard phrase-based statistical machine translation system trained on the BTEC corpus consisting of 691,829 Japanese and English sentence pairs. To see the upper bound of the machine translation performance, we manually constructed a bilingual dictionary which consists of word categories and English translations of all proper nouns in the test set. Prior to the translation, proper nouns in source sentences were replaced with high-frequency words of the same category in the training corpus [31]. The target sentences were then acquired by translating the modified source sentences. Finally, the high-frequency words in the target sentences were replaced with target words for the untrained words. The reason why we use high-frequency words is that we expect them to be already trained well, in other words, the high-frequency words may already appear frequently in phrase tables and therefore provide ample statistics. A single human judge was used to score the translation output, according to a five-level scale ranging from 'perfect' through 'acceptable' to 'nonsense'.

We evaluated the transliteration component of the system on the transliteration task used in our field experiments. Each source word was transliterated from kanji into romaji, and the output evaluated against a reference

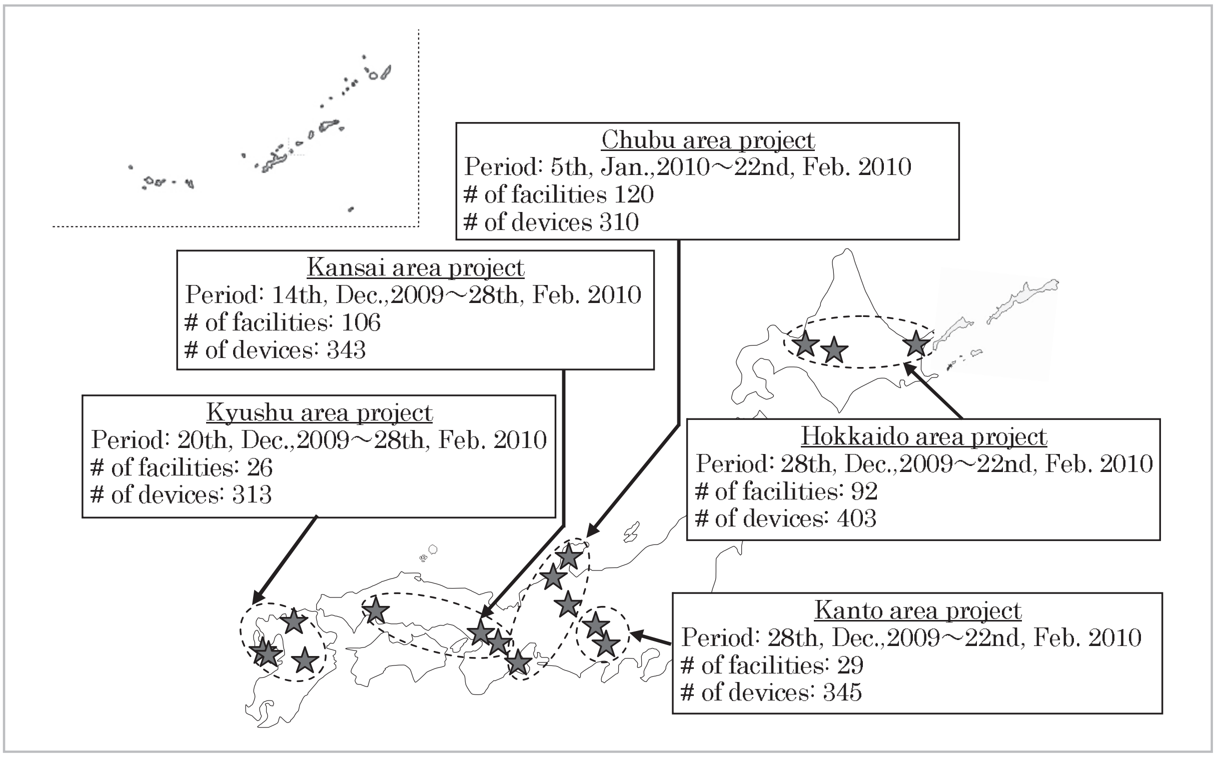

Fig.6 Overview of the five local projects

set with up to 3 references per input sequence using the same scoring metrics as the NEWS2010 workshop. The transliteration accuracy on this task was 19.14%, this is lower than the NEWS2010 task scores but this is to be expected as our task contains both multi-word sequences and also sequences that are not transliterations of each other. Nonetheless the F-score, which is a character level score based on the length of the longest common subsequence, was 71.03, indicating the performance is quite respectable at the character level.

Visual inspection of the output, showed that the biggest single cause of errors in the system is due to the assumption that all words should be transliterated. In reality, expressions may need to be translated, or partially transliterated and partially translated. One such example from the test corpus being the expression in Kanji: '伊丹空港'. This should be correctly transliterated as 'Itami Airport', however the transliteration system produced 'itamikuukou' as output - a perfect transcription into kana, but incorrect nonetheless. Future research will need to address ways to identifying when to transliterate and when to translate. The problem of re-ordering was also an issue for our approach, although these errors were found in less than 4% of the sentences in the test corpus. An example of this type of error is '富士山'. The system's output in this case was "fujisan", however the correct outputs were "Mount Fuji" or "Mt. Fuji". In this example, both transliteration and translation are required, but in addition the order of the words is swapped. We believe that modeling this re-ordering process would give rise to improvements in system performance.

Figure 7 shows the field test translation evaluation results. The vertical axis represents the acceptance ratio which is the ratio of better than acceptable translation sentences to the total number of test sentences. To see the relationship between the dictionary size and the translation performance, we controlled the dictionary size. The horizontal axis shows dictionary coverage, which is the ratio of the

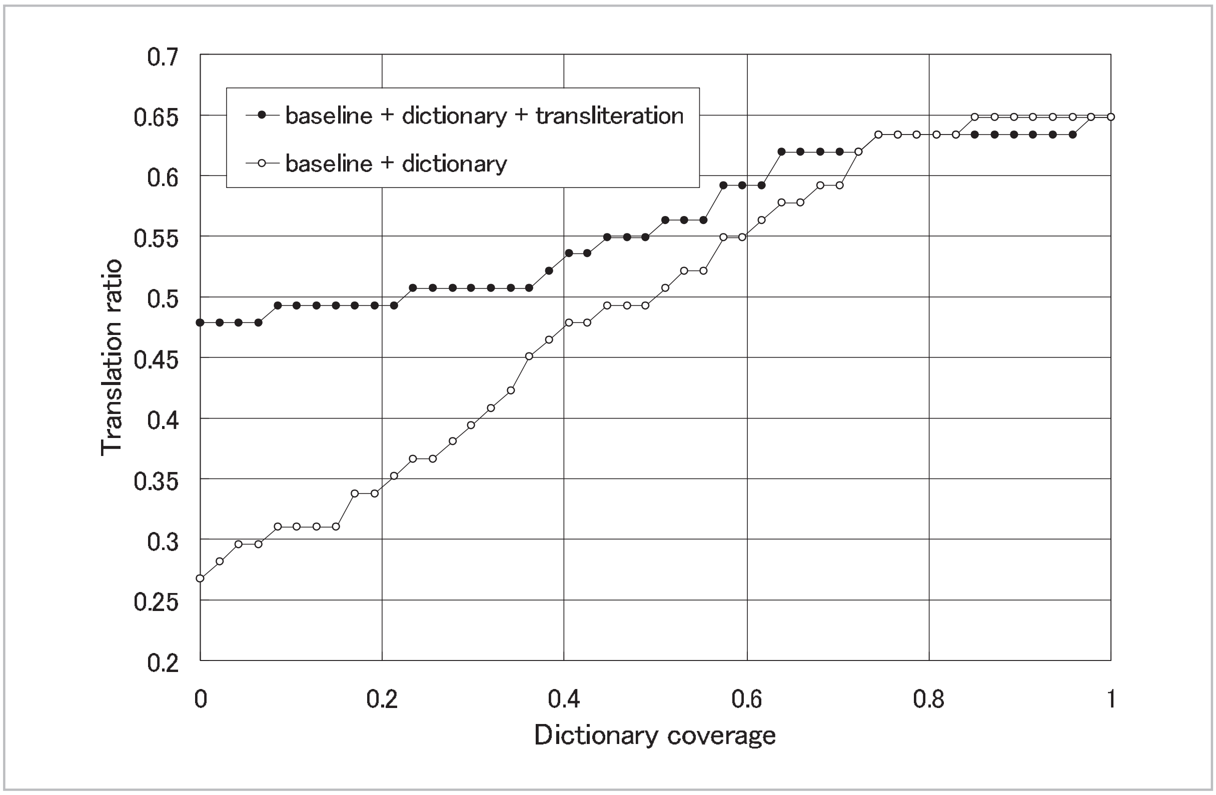

Fig.7 *Results of machine translation evaluation*

proper nouns covered by the dictionary to the total number of occurrences of proper nouns in the evaluation set. '○' plots shows machine translation with conventional dictionary usage [31]. While, '●' plots arised from translation strategy that uses the transliterated results *only if* there is no entry in the dictionary [*2].

It is clear from the figure that the use of transliteration within a machine translation system can lead to improvements in translation quality from a user's perspective. In Figure 7, the values at 0 on the x-axis represent the system performance without the ability to look up the correct translations for named entities in a dictionary. The translation performance is significantly higher at this point. The differences between the two lines on the graph gradually diminish as the dictionary is used more and more for the translation of named entities, and at a value of 1 on the x-axis of course the two lines meet, as the strategies are equivalent to each other at this point. When dictionary coverage is low, we can expect transliteration to have a substantial impact on system performance, however, when we look at the line "baseline + dictionary + transliteration" at both 0 and 1 on the x-axis we can see the effect of the errors arising from the transliteration system. Adding the dictionary to the transliteration system clearly improves the overall system performance. Therefore, further work is needed to improve the quality of machine transliteration systems.

7 Conclusion

In this paper we have presented a novel Bayesian bilingual alignment scheme and applied it to the task of building translation models for transliteration by phrase-based statistical machine translation. Furthermore, we have conducted a human evaluation to investigate the effectiveness of integrating a machine transliteration system with a machine translation system. Our results show that adding transliterations of unknown named entities into the machine translation output has a positive effect on translation quality from a user's perspective.

Our Bayesian model for bilingual alignment was motivated by the fact that traditional models of phrasal alignment rely on maximum likelihood training coupled with the EM algorithm, but have serious issues with overfitting the training data. Because of these issues, alignment is typically performed in a one-to-many manner from source-to-target and from target-to-source and the phrase extraction process proceeds heuristically from an alignment table. Our approach offers the ability to align the training data in a many-to-many fashion directly using Bayesian techniques that offer a simple yet elegant solution to the issues inherent in maximum likelihood training.

We investigated the quality of the bilingual phrasal alignment achievable with unsupervised Bayesian alignment, and designed experiments to compare directly to a standard GIZA++/*grow-diag-final-and* phrase extraction procedure and also to models built using the m2m bilingual sequence aligner. Our experiments show that the Bayesian approach is able to produce more compact models that can offer comparable or higher transliteration performance than both of the baseline systems.

One virtue of our approach stems from the fact that the Dirichlet process model is able to assign a probability to any bilingual word pair. We believe this type of model in has considerable potential utility in transliteration mining and corpus filtering, since it provides a principled way of scoring any potential transliteration candidate. In [32] this approach was used to achieve state-of-the-art levels of performance in transliteration mining tasks.

In future research we plan to improve the underlying Dirichlet process model in order to better model the data, moving to higher-order and hierarchical models.

[*2] To incorporate transliterated results into machine translation, we used the same framework as the conventional dictionary-based technique which was proposed by [31]. In this method, category of the word must be known. In our experiments, even for the transliterated words, we used a manually assigned category.

References

1 K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” ACL ’02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 311–318, Association for Computational Linguistics, 2001.

2 G. Doddington, “Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics,” Proceedings of the HLT Conference, San Diego, California, 2002.

3 X. Duan, D. Xiong, H. Zhang, M. Zhang, and H. Li, “I2r’s machine translation system for iwslt 2009,” Proceedings of the International Workshop on Spoken Language Translation, pp. 50–54, 2009.

4 A. Finch and E. Sumita, “Phrase-based machine transliteration,” Proc. 3rd International Joint Conference on NLP, Hyderabad, India, 2008.

5 T. Rama and K. Gali, “Modeling machine transliteration as a phrase based statistical machine translation problem,” NEWS ’09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Morristown, NJ, USA, pp. 124–127, Association for Computational Linguistics, 2009.

6 S. Noeman, “Language independent transliteration system using phrase based smt approach on substrings,” NEWS ’09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Morristown, NJ, USA, pp. 112–115, Association for Computational Linguistics, 2009.

7 H. Li, M. Zhang, and J. Su, “A joint source-channel model for machine transliteration,” ACL ’04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, p. 159, Association for Computational Linguistics, 2004.

8 F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” Computational Linguistics, Vol. 29, No. 1, pp. 19–51, 2003.

9 P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowa, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: open source toolkit for statistical machine translation,” ACL 2007: proceedings of demo and poster sessions, Prague, Czeck Republic, pp. 177–180, June 2007.

10 D. Yang, P. Dixon, Y. C. Pan, T. Oonishi, M. Nakamura, and S. Furui, “Combining a two-step conditional random field model and a joint source channel model for machine transliteration,” NEWS ’09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Morristown, NJ, USA, pp. 72–75, Association for Computational Linguistics, 2009.

11 D. Marcu and W. Wong, “A phrase-based, joint probability model for statistical machine translation,” In Proceedings of EMNLP, pp. 133–139, 2002.

12 M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” Speech Commun., Vol. 50, No. 5, pp. 434–451, 2008.

13 P. Blunsom, T. Cohn, C. Dyer, and M. Osborne, “A gibbs sampler for phrasal synchronous grammar induction,” Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL, Suntec, Singapore, pp. 782–790, Association for Computational Linguistics, August 2009.

14 J. DeNero, A. Bouchard-Côté, and D. Klein, “Sampling alignment structure under a bayesian translation model,” Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP ’08, Stroudsburg, PA, USA, pp. 314–323, Association for Computational Linguistics, 2008.

15 G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, “An unsupervised model for joint phrase alignment and extraction,” ACL, pp. 632–641, 2011.

16 J. Wuebker, A. Mauser, and H. Ney, “Training phrase translation models with leaving-one-out,” Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 475–484, Association for Computational Linguistics, July 2010.

17 Y. Huang, M. Zhang, and C. L. Tan, "Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars," ACL (Short Papers), pp. 534–539, 2011.

18 J. Xu, J. Gao, K. Toutanova, and H. Ney, "Bayesian semi-supervised chinese word segmentation for statistical machine translation," COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics, Morristown, NJ, USA, pp. 1017–1024, Association for Computational Linguistics, 2008.

19 D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested pitman-yor language modeling," 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: Vol. 1, Morristown, NJ, USA, pp. 100–108, Association for Computational Linguistics, 2009.

20 A. Finch and E. Sumita, "A Bayesian Model of Bilingual Segmentation for Transliteration," Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT), ed. M. Federico, I. Lane, M. Paul, and F. Yvon, pp. 259–266, 2010.

21 S. Goldwater, T. L. Griffiths, and M. Johnson, "Contextual dependencies in unsupervised word segmentation," ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Morristown, NJ, USA, pp. 673–680, Association for Computational Linguistics, 2006.

22 D. J. Aldous, "Exchangeability and related topics," in École d'été de probabilités de Saint-Flour, XIII—1983, Lecture Notes in Math., Vol. 1117, pp. 1–198, Springer, Berlin, 1985.

23 F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), pp. 295–302, 2002.

24 F. J. Och, "Minimum error rate training for statistical machine translation," Proceedings of the ACL, 2003.

25 T. Rama and K. Gali, "Modeling machine transliteration as a phrase based statistical machine translation problem," In Proc. ACL/IJCNLP Named Entities Workshop Shared Task, 2009.

26 M. Z. Haizhou Li, A Kumaran, and V. Pervouchine, "Whitepaper of news 2010 shared task on transliteration generation," In Proc. ACL Named Entities Workshop Shared Task, 2010.

27 S. Jiampojamarn, G. Kondrak, and T. Sherif, "Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion," Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, Rochester, New York, pp. 372–379, Association for Computational Linguistics, April 2007.

28 E. S. Ristad and P. N. Yianilos, "Learning string edit distance," IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 20, No. 5, pp. 522–532, May 1998.

29 A. Finch, P. Dixon, and E. Sumita, "Integrating models derived from non-parametric bayesian cosegmentation into a statistical machine transliteration system," Proceedings of the Named Entities Workshop, Chiang Mai, Thailand, pp. 23–27, Asian Federation of Natural Language Processing, Nov 2011.

30 H. KAWAI, R. ISOTANI, K. YASUDA, E. SUMITA, M. UTIYAMA, S. MATSUDA, Y. ASHIKARI, and S. NAKAMURA, "An overview of a nation-wide field experiment of speech-to-speech translation in fiscal year 2009 (Japanese only)," Proceedings of 2010 autumn meeting of Acoustical Society of Japan, pp. 99–102, 2010.

31 H. Okuma, H. Yamamoto, and E. Sumita, "Introducint a translation dictionary into phrase-based smt," The IEICE Transactions on Information and Systems, Vol. 91-D, No. 7, pp. 2051–2057, 2008.

32 T. Fukunishi, A. Finch, S. Yamamoto, and E. Sumita, "Using features from a bilingual alignment model in transliteration mining," Proceedings of the 3rd Named Entities Workshop (NEWS 2011), pp. 49–57, 2011.

(Accepted June 14, 2012)

Andrew Finch, *Ph.D.*

Senior Researcher, Multilingual Translation Laboratory, Universal Communication Research Institute

Machine Translation, Natural Language Processing

YASUDA Keiji, *Dr. Eng.*

Senior Researcher, Multilingual Translation Laboratory, Universal Communication Research Institute

Machine Translation, Natural Language Processing