Word Alignment Quality in the IBM 2 Mixture Model ⋆

Jorge Civera and Alfons Juan

ITI/DSIC, Universidad Politécnica de Valencia, Spain

Abstract. Finite mixture modelling is a standard pattern recognition technique. However, in statistical machine translation (SMT), the use of mixture modelling is currently being explored. Two main advantages of the mixture approach are first, its flexibility to find an appropriate tradeoff between model complexity and the amount of training data available and second, its capability to learn specific probability distributions that better fit subsets of the training dataset. This latter advantage is even more important in SMT, since it is widely accepted that most state-of-the-art translation models proposed have limited application to restricted semantic domains. In this work, we revisit the mixture extension of the well-known M2[1] translation model. The M2 mixture model is evaluated on a word alignment large-scale task obtaining encouraging results that prove the applicability of finite mixture modelling in SMT.

1 Introduction

Finite mixture modelling is a popular approach for density estimation in many scientific areas [1]. On the one hand, mixtures are flexible enough for finding an appropriate tradeoff between model complexity and the amount of training data available. Usually, model complexity is controlled by varying the number of mixture components while keeping the same parametric form for all components. On the other hand, maximum likelihood estimation of mixture parameters can be reliably accomplished by the well-known *Expectation-Maximisation (EM)* algorithm [2, 3].

One of the most interesting properties of mixture modelling is its capability to learn a specific probability distribution in a multimodal dataset that better explains the general data generation process. In translation tasks, these multimodal datasets are not an exception, but the general case. Indeed, it is easy to find corpora from which several topics could be drawn. These topics define sets of topic-specific lexicons that need to be translated taking into the Semitic context in which they are found. This semantic ambiguity problem could be overcome by learning topic-dependent translation models that capture together the semantic context and the translation process. The application of finite mixture modelling to SMT is currently being explored with successful results [4–6].

Previous work on finite mixture modelling applied to SMT has mainly focused on the mixture extension of word-based alignment models, more precisely, the well-known

⋆ Work partially supported by *Ministerio de Educación y Ciencia*, the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01, and the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

[1] Known as IBM 2 model in the literature.

Civera J. and Juan A. (2008).
Word Alignment Quality in the IBM 2 Mixture Model.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 93-102

IBM alignment models [7, 8]. In [4], a mixture extension of the M2 model is proposed, reporting appealing results on a small synthetic task [4]. However, the question that arises is whether these positive results on a small task can be extrapolated to large-scale tasks. This paper presents an alternative evaluation of the M2 mixture model on a word alignment shared task that serves as a reference task in SMT [9–13].

Indeed, word alignment is the first step towards the construction of modern phrase-based SMT systems [14–17]. It involves the induction of a word mapping from a (source) language into another (target) language over bilingual sentences. The second phase uses statistics over these learnt word alignments to translate new sentences.

In this paper, we first review the M2 in Section 2, before deriving the M2 mixture model in Section 3. In Section 4, we introduce the evaluation metrics that are used to assess word alignment quality of the proposed model on the shared task presented in Section 5. Section 6 is devoted to experimental results and Section 7 concludes and provides an outlook on future work.

2 The M2 model

2.1 The Model

Let (x, y) be a pair of source-target sentences; i.e. x is a sentence in a certain source language and y is its corresponding translation in a different target language. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the source and target vocabularies, respectively. The IBM alignment models are parametric models for the translation probability $p(x \mid y)$; i.e., the probability that x is the source sentence from which we get a given translation y.

The IBM alignment models assume that each source word is *connected to exactly one* target word. Also, it is assumed that the target sentence has an initial *NULL* or *empty* word to which source words with no direct translation are connected. Formally, a hidden variable $a = a_1 a_2 \cdots a_{|x|}$ is introduced to reveal, for each source word position j, the target word position $a_j \in \{0, 1, \ldots, |y|\}$ to which it is connected. Thus,

$$p(x \mid y) = \sum_{a \in \mathcal{A}(x,y)} p(x, a \mid y) \tag{1}$$

where $\mathcal{A}(x, y)$ denotes the set of all possible alignments between x and y. The term $p(x, a \mid y)$ can be factorised as source position-dependent probabilities

$$p(x, a \mid y) = \prod_{j=1}^{|x|} p(x_j, a_j \mid a_1^{j-1}, x_1^{j-1}, y) \tag{2}$$

In the case of the IBM model 2, it is assumed that a_j only depends on j and $|y|$, and that x_j only depends on the target word to which it is connected, y_{a_j}. Hence,

$$p(x_j, a_j | a_1^{j-1}, x_1^{j-1}, y) := p(a_j | j, |y|)\, p(x_j | y_{a_j}) \tag{3}$$

and the set of unknown parameters $\boldsymbol{\Theta}$ comprises

$$\boldsymbol{\Theta} = \begin{cases} p(i \mid j, |y|) & \forall\, i \in \{0, 1, \ldots, |y|\},\ j \in \{1, \ldots, |x|\} \text{ and } |y| \\ p(u \mid v) & u \in \mathcal{X}, v \in \mathcal{Y}. \end{cases} \tag{4}$$

Note that the alignment parameters defined here are slightly different from those defined in the original parametrisation [8], which also depend on $|x|$, $p(i \mid j, |x|, |y|)$.

Putting Eqs. (1), (2) and (3) together, we define the M2 model, after some straightforward manipulations, as follows:

$$p(x \mid y) = \prod_{j=1}^{|x|} \sum_{i=0}^{|y|} p(i \mid j, |y|)\, p(x_j | y_i). \tag{5}$$

2.2 Maximum Likelihood Estimation

It is not difficult to derive an EM algorithm to perform maximum likelihood estimation of $\boldsymbol{\Theta}$ with respect to a collection of N independent training samples $(X, Y) = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. The log-likelihood function is:

$$L(\boldsymbol{\Theta}) = \sum_{n=1}^{N} \log \sum_{a_n} p(x_n, a_n | y_n) \tag{6}$$

with

$$\begin{aligned} p(x_n, a_n | y_n) &= \prod_{j=1}^{|x_n|} p(a_{nj} | j, |y_n|)\, p(x_{nj} | y_{n a_{nj}}) \\ &= \prod_{j=1}^{|x_n|} \prod_{i=0}^{|y_n|} \left[p(i \mid j, |y_n|)\, p(x_{nj} | y_{ni}) \right]^{a_{nji}} \end{aligned} \tag{7}$$

where, for convenience, the alignment variable, $a_{nj} \in \{0, 1, \ldots, |y_n|\}$, has been rewritten as an indicator vector in Eq. (7), $a_{nj} = (a_{nj0}, \ldots, a_{nj|y_n|})$, with 1 in position a_{nji} and zeros elsewhere.

Now, we can define A as the set of alignment indicator vectors associated with the bilingual pairs (X, Y) with

$$A = (a_1, \ldots, a_n, \ldots, a_N)^t \tag{8}$$

where variable A is the missing data in the M2 model.

The EM algorithm maximises Eq. (6) iteratively, through the application of two basic steps in each iteration: the E(xpectation) step and the M(aximisation) step.

The E step computes the expected value of the logarithm of $p(X, A \mid Y)$, given the (incomplete) data samples (X, Y) and a current estimate of $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^{(k)}$. Given that the alignment variables in A are independent from each other, we can compute the E step as the Q function in the EM terminology,

$$Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) = \sum_{n=1}^{N} E(\log p(x_n, a_n \mid y_n; \boldsymbol{\Theta}) \mid x_n, y_n, \boldsymbol{\Theta}^{(k)}) \tag{9}$$

$$= \sum_{n=1}^{N} \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} a_{nji}^{(k)} \left[\log p(i \mid j, |y_n|) + \log p(x_{nj} \mid y_{ni}) \right] \tag{10}$$

with

$$a_{nji}^{(k)} = \frac{p(i \mid j, |y_n|)^{(k)}\, p(x_{nj} \mid y_{ni})^{(k)}}{\sum\limits_{i'=0}^{|y_n|} p(i' \mid j, |y_n|)^{(k)}\, p(x_{nj} \mid y_{ni'})^{(k)}}. \tag{11}$$

That is, the expectation of word x_{nj} to be connected to y_{ni} is our current estimation of the probability of x_{nj} to be translated into y_{ni}, instead of any other word in y_n (including the NULL word).

Then, the M step finds a new estimate of $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^{(k+1)}$, by maximising Eq. (9), using Eq. (11) instead of the missing a_{nji}. This results in:

$$p(i \mid j, |y|)^{(k+1)} = \frac{\sum\limits_{\substack{n=1\\ j\le|x_n|\\ |y_n|=|y|}}^{N} a_{nji}^{(k)}}{\sum\limits_{i'=0}^{|y|} \sum\limits_{\substack{n=1\\ j\le|x_n|\\ |y_n|=|y|}}^{N} a_{nji'}^{(k)}} \qquad \forall i, j \text{ and } |y|; \tag{12}$$

and

$$p(u|v)^{(k+1)} = \frac{\sum\limits_{n=1}^{N} \sum\limits_{\substack{j=1\\ x_{nj}=u}}^{|x_n|} \sum\limits_{\substack{i=0\\ y_{ni}=v}}^{|y_n|} a_{nji}^{(k)}}{\sum\limits_{u'\in\mathcal{X}} \sum\limits_{n=1}^{N} \sum\limits_{\substack{j=1\\ x_{nj}=u'}}^{|x_n|} \sum\limits_{\substack{i=0\\ y_{ni}=v}}^{|y_n|} a_{nji}^{(k)}} \qquad \forall u \in \mathcal{X} \text{ and } v \in \mathcal{Y}. \tag{13}$$

An initial estimate for $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^0$, is required for the EM algorithm to start. In the case of the M2 model, we use the initial solution given by the M1 model, which is a particular case of the M2 model in which alignment probabilities are uniformly distributed; i.e.,

$$p(i \mid j, |y|)^{(k+1)} = \frac{1}{|y|+1} \qquad \forall\, i, j \text{ and } |y|. \tag{14}$$

3 Mixture of M2 models

3.1 The Model

A finite mixture model is a probability (density) function of the form:

$$p(z) = \sum_{t=1}^{T} p(t)\, p(z \mid t) \tag{15}$$

where T is the *number of mixture components* and, for each component t, $p(t) \in [0, 1]$ is its *prior or coefficient* and $p(z \mid t)$ is its *component-conditional probability (density) function*. It can be seen as a generative model that first selects the tth component with probability $p(t)$ and then generates z in accordance with $p(z \mid t)$. It is clear that finite mixture modelling allows generalisation of any given probabilistic model by simply using more than one component.

In this work, we are interested in modelling the translation probability $p(x \mid y)$ using a T-component, y-conditional mixture of M2 models:

$$p(x \mid y) = \sum_{t=1}^{T} p(t)\, p(x \mid y, t) \tag{16}$$

where

$$p(x|y,t) = \prod_{j=1}^{|x|} \sum_{i=0}^{|y|} p(i \mid j, |y|, t)\, p(x_j|y_i, t) \tag{17}$$

Note that we could have made $p(t)$ to depend on y in Eq. 16 but, for simplicity, this is left for future work. Thus, the global vector of parameters $\boldsymbol{\Theta}$ is

$$\boldsymbol{\Theta} = (p(1), \ldots, p(t), \ldots, p(T); \boldsymbol{\Theta}_1, \ldots \boldsymbol{\Theta}_t, \ldots, \boldsymbol{\Theta}_T)^t. \tag{18}$$

where for each component t, $p(t)$ is its mixture prior or coefficient and $\boldsymbol{\Theta}_t$ comprises the component-conditional parameters

$$\boldsymbol{\Theta}_t = \begin{cases} p(i \mid j, |y|, t) & \forall\, i \in \{0, 1, \ldots, |y|\},\ j \in \{1, \ldots, |x|\} \text{ and } |y| \\ p(u \mid v, t) & u \in \mathcal{X}, v \in \mathcal{Y}. \end{cases} \tag{19}$$

It is easy to extend the EM algorithm developed in the previous section to the case of M2 mixtures. The log-likelihood function of $\boldsymbol{\Theta}$ with respect to N training samples is

$$L(\boldsymbol{\Theta}) = \sum_{n=1}^{N} \log \sum_{z_n} \sum_{a_n} p(x_n, z_n, a_n|y_n) \tag{20}$$

where $z_n = (z_{n1}, \ldots, z_{nT})$ is an indicator vector for the component generating x_n, and

$$p(x_n, z_n, a_n \mid y_n) = \prod_{t=1}^{T} [p(t)\, p(x_n, a_n \mid y_n, t)]^{z_{nt}} \tag{21}$$

with

$$p(x_n, a_n \mid y_n, t) = \prod_{j=1}^{|x_n|} \prod_{i=0}^{|y_n|} [p(i \mid j, |y_n|, t) p(x_{nj} \mid y_{ni}, t)]^{a_{nji}}$$

where, as in the previous section, $a_{nji} = 1$ means that the nth training pair has its source position j connected to target position i. Note that data completion in the mixture case includes the alignments A and the component labels

$$Z = (z_1, \ldots, z_n, \ldots, z_N)^t \tag{22}$$

as well. Thus, the Q function for the M2 mixture model becomes

$$\begin{aligned} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) = \sum_{n=1}^{N} \sum_{t=1}^{T} & z_{nt}^{(k)} \log p(t) \\ + \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} & (z_{nt}\, a_{nji})^{(k)} \left[\log p(i \mid j, |y_n|, t) + \log p(x_{nj} \mid y_{ni}, t)\right]. \end{aligned} \tag{23}$$

with

$$z_{nt}^{(k)} = \frac{p(t)^{(k)}\, p(x_n \mid y_n, t)^{(k)}}{\sum\limits_{t'=1}^{T} p(t')^{(k)}\, p(x_n \mid y_n, t')^{(k)}} \tag{24}$$

and the expected value of $z_{nt}\, a_{nji}$,

$$(z_{nt}\, a_{nji})^{(k)} = z_{nt}^{(k)}\, a_{njit}^{(k)} \tag{25}$$

with

$$a_{njit}^{(k)} = \frac{p(i \mid j, |y_n|, t)^{(k)}\, p(x_{nj} \mid y_{ni}, t)^{(k)}}{\sum\limits_{i'=0}^{|y_n|} p(i' \mid j, |y_n|, t)^{(k)}\, p(x_{nj} \mid y_{ni'}, t)^{(k)}} \tag{26}$$

Note that Eq. (26) is just a component-conditional version of Eq. (11).

The M step now includes an updating rule for the mixture coefficients,

$$p(t)^{(k+1)} = \frac{1}{N} \sum_{n=1}^{N} z_{nt}^{(k)} \qquad \forall t \tag{27}$$

and component-conditional versions of Eq. (12) and (13):

$$p(i \mid j, |y|, t)^{(k+1)} = \frac{\sum\limits_{\substack{n=1 \\ j \le |x_n| \\ |y_n| = |y|}}^{N} z_{nt}^{(k)}\, a_{njit}^{(k)}}{\sum\limits_{i'=0}^{|y|} \sum\limits_{\substack{n=1 \\ j \le |x_n| \\ |y_n| = |y|}}^{N} z_{nt}^{(k)}\, a_{nji't}^{(k)}} \qquad \forall t, i, j \text{ and } |y| \tag{28}$$

and

$$p(u \mid v, t)^{(k+1)} = \frac{\sum\limits_{n=1}^{N} \sum\limits_{\substack{j=1 \\ x_{nj}=u}}^{|x_n|} \sum\limits_{\substack{i=0 \\ y_{ni}=v}}^{|y_n|} z_{nt}^{(k)}\, a_{njit}^{(k)}}{\sum\limits_{u' \in \mathcal{X}} \sum\limits_{n=1}^{N} \sum\limits_{\substack{j=1 \\ x_{nj}=u'}}^{|x_n|} \sum\limits_{\substack{i=0 \\ y_{ni}=v}}^{|y_n|} z_{nt}^{(k)}\, a_{njit}^{(k)}} \qquad \forall t, u \text{ and } v. \tag{29}$$

The initialisation technique for the M2 model can be easily extended to the mixture case; i.e. by using a solution from a simpler mixture of IBM1 models.

3.2 Viterbi Alignment

In Eq. (1), we introduced the concept of alignment as an assignment between source and target words, more precisely between source and target positions. However, this alignment information was missing in the translation process, and we had to marginalise over all possible values of the alignment variable.

In practise, we are interested in the most probable alignment, also known as the Viterbi alignment,

$$\hat{a} = \operatorname*{argmax}_{a} p(x, a \mid y; \boldsymbol{\Theta}). \tag{30}$$

Assuming a conventional M2 model, Eq. (30) can be trivially maximised

$$\hat{a} = \operatorname*{argmax}_{a} \prod_{j=1}^{|x|} \max_{a_j} p(a_j \mid j, |y|) \, p(x_j \mid y_{a_j}). \tag{31}$$

In other words, the Viterbi alignment for the M2 model is computed as a local maximisation for each source position, being its asymptotic cost $O(|x| \cdot |y|)$.

Nevertheless, the computation of the Viterbi alignment for the M2 mixture model is approximated by maximising over the components in the mixture,

$$\hat{a} \approx \operatorname*{argmax}_{a} \max_{t=1,\ldots,T} \; p(t) \prod_{j=1}^{|x|} \max_{a_j} p(a_j \mid j, |y|, t) \, p(x_j \mid y_{a_j}, t) \tag{32}$$

being its asymptotic cost $O(T \cdot |x| \cdot |y|)$.

4 Evaluation Metrics

Word alignment is considered to be a complex and ambiguous task [18], and therefore we need an annotation scheme that allows ambiguous alignments to be defined. The experts conducting the annotation process are permitted to use two types of alignments: S (sure) and P (probable), such that $S \subseteq P$. Both of them may contain many-to-one and one-to-many relationships. P alignments are specially useful in cases like idiomatic expressions, free translations and missing function words.

Given a Viterbi alignment A defined as

$$A = \{(j, a_j) \mid 1 \leq a_j \leq |y|\} \quad \forall j \; 1 \leq j \leq |x| \tag{33}$$

where the NULL alignments has been intentionally left out of the evaluation, precision and recall measures can be computed

$$\text{recall} = \frac{|A \cap S|}{|S|}, \quad \text{precision} = \frac{|A \cap P|}{|A|} \tag{34}$$

as well as the alignment error rate (AER) [9] that is related to the well-known F-measure

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{35}$$

These definitions of precision, recall and AER are based on the assumption that a recall error can occur only if an S alignment is not found and a precision error can occur only if the found alignment is not even P.

AER has been widely used in the scientific community to evaluate word alignment quality until very recently [9–13, 19]. However in [20], Fraser and Marcu claim that

AER, though derived from the F-measure, does not penalise unbalanced precision and recall, where $S \subset P$. As a result, AER is low correlated with translation quality, as previously reported in [21]. For this reason, they suggest to use an α-optimised F-measure that controls the contribution of precision and recall,

$$\text{F-measure}(\alpha) = \frac{\text{precision} \cdot \text{recall}}{\alpha \cdot \text{recall} + (1 - \alpha) \cdot \text{precision}} \tag{36}$$

so that this metric is highly correlated with SMT performance.

5 Corpora

The corpus employed in the experiments was the French-English Hansard task consisting of the debates of the Canadian parliament. This corpus is one of the resources that were used during the word alignment shared task organised during the HLT/NAACL 2003 workshop on “Building and Using Parallel Texts” [22].

The independent test set is that defined in [23] which was manually labelled by two annotators. Each annotator comes up with a S and P alignment set. The S alignment sets from each annotator are intersected to defined the reference S alignment set, while the reference P alignment set is the result of the union of the P alignment sets from both annotators. The definition of the S and P alignment sets in this way guarantees an alignment error rate of zero percent when we compare the S alignments of each annotator with the reference alignment. The corpus statistics are shown in Table 1.

Table 1. Statistics on the French-English Hansard task (K denotes $\times 10^3$, and M denotes $\times 10^6$).

	Training set		Trial set		Test set	
	Fr	En	Fr	En	Fr	En
sentence pairs	1.1M		37		447	
average length	20	17	19	17	17	15
vocabulary size	87K	68K	0.3K	0.3K	1.9K	1.7K
running words	24M	20M	0.7K	0.7K	7.8K	7.0K
singletons	27K	20K	0.3K	0.2K	1.3K	1.1K

6 Experimental Results

The objective of these experiments is to study the evolution of AER and α-optimised F-measure on the Hansard task as a function of the number of components in the M2 mixture model. The results with the GIZA++ toolkit are for sanity check reasons. Smoothing parameters were manually tuned on the trial partition to minimise AER.

Table 2 presents AER figures on the test partition for M2 mixture model. Each number in Table 2 is an average over values obtained from 10 randomised initialisation, that are used to estimate confidence intervals computed as twice the standard deviation. These experiments were performed for both directions, English-French (En-Fr) and French-English (Fr-En) and varying the number of components in the mixture model ($T = 1, 2, 3$). Experiments beyond 3 components per mixture were not run because of

Table 2. AER figures on the test partition of the Hansard corpus for the M2 mixture model varying the number of components in the mixture ($T = 1,\ 2,\ 3$) and the conventional M2 model implemented in the GIZA++ toolkit.

AER	GIZA++	1	2	3
Fr-En	20.0	19.6	19.0 ± 0.1	18.8 ± 0.1
En-Fr	18.3	17.6	17.2 ± 0.1	16.8 ± 0.1

Table 3. F-measure ($\alpha = 0.2$) figures on the test partition of the Hansard corpus for the M2 mixture model varying the number of components in the mixture ($T = 1,\ 2,\ 3$) and the conventional M2 model implemented in the GIZA++ toolkit.

F-measure	GIZA++	1	2	3
Fr-En	85.5	86.1	86.6 ± 0.2	86.8 ± 0.1
En-Fr	85.8	86.6	87.1 ± 0.1	87.4 ± 0.1

memory requirements. The number of iterations per model was $mix\,1^5\,2^5$ for the M2 mixture model. Viterbi alignments were calculated according to Eq. (32).

In Table 2, there is a statistically significant improvement when we go from the conventional single-component M2 model to the multiple-component M2 mixture model for both language directions. Besides, the decrease in AER on the English-French direction from two to three components is also statistically significant.

To have a broader view of the benefits and properties of the models in question, we decided to carry out an evaluation in terms of α-optimised F-measure shown in Table 3. According to [20] and being aware of the differences between our work and that presented in [20], we set $\alpha = 0.2$ in order to compute the corresponding F-measure that would be fairly correlated with the performance of phrasal SMT performance.

Similarly to the AER results in Table 2, the computed F-measure shows that there is a significant improvement when we compare the conventional M2 model to the multiple-component M2 mixture model. However, the small difference between two and three components in terms of AER is diminished in the evaluation with F-measure. In any case, the interpretation of the figures in Table 3 foresees an improvement in translation quality if we train a phrase-based SMT system with the Viterbi alignments of the multiple-component M2 mixture model, instead of the conventional M2 model. This hypothesis has to be corroborated with translation experiments on the Hansard corpus.

7 Conclusions and Future Work

In this paper, we have revisited the M2 mixture model to perform an alternative evaluation based on Viterbi alignment quality. AER and F-measure results reported on a large-scale shared task, as the Hansard corpus, unveil statistically significant improvements of the multiple-component M2 mixture model over the conventional M2 model.

These encouraging results suggest the necessity of further evaluation for the M2 mixture model. This further evaluation would entail the training of a phrase-based SMT system using word alignments supplied by the M2 mixture model. To this purpose, we can employ the publicly available Moses toolkit [24], which implements a state-of-the-art phrase-based SMT system, and study the evolution of the translation quality of the resulting system as a function of the number of components in the M2 mixture model.

These results would corroborate the relation between alignment quality and translation quality, demonstrating so the appropriateness of finite mixture modeling in SMT.

Alternatively, it would be interesting to develop mixture extensions of superior IBM models, like Model 4 and 5, or the log-linear Model 6 [9] to fairly valorate the contribution of mixture modeling to state-of-the-art alignment results.

References

1. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley (2000)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm (with discussion). Journal of the Royal Stat. Society B **39** (1977) 1–38
3. Wu, C.: On the convergence properties of the EM algorithm. The Annals of Statistics **11** (1983) 95–103
4. Civera, J., Juan, A.: Mixtures of IBM Model 2. In: Proc. of EAMT'06. (2006) 159–167
5. Zhao, B., Xing, E.P.: BiTAM: Bilingual Topic AdMixture Models for Word Alignment. In: Proc. of COLING/ACL'06. (2006)
6. Civera, J., Juan, A.: Domain adaptation in statistical machine translation with mixture modelling. In: Proc. of the 2nd Workshop in Statistical Machine Translation. (2007) 177–180
7. Brown, et al.: A Statistical Approach to Machine Translation. Comp.Ling. **16** (1990) 79–85
8. Brown, et al.: The Mathematics of Statistical Machine Translation: Parameter Estimation. Comp.Ling. **19** (1993) 263–311
9. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comp.Ling. **29** (2003) 19–51
10. Mihalcea, R., Pedersen, T.: An evaluation exercise for word alignment. In: Proc. of the HLT-NAACL'03: Workshop on Building and using parallel texts. (2003) 1–10
11. Taskar, B., Lacoste-Julien, S., Klein, D.: A discriminative matching approach to word alignment. In: Proc. of HLT'05. (2005) 73–80
12. Blunsom, P., Cohn, T.: Discriminative word alignment with conditional random fields. In: Proc. of ACL '06. (2006) 65–72
13. Zhao, B., Vogel, S.: Word alignment based on bilingual bracketing. In: Proc. of the HLT-NAACL'03: Workshop on Building and using parallel texts. (2003) 15–18
14. Koehn, P., et al.: Statistical phrase-based translation. In: Proc. of NAACL'03. (2003) 48–54
15. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Comp.Ling. **30** (2004) 417–449
16. Chiang, D.: Hierarchical phrase-based translation. Comp.Ling. **33** (2007) 201–228
17. Callison-Burch, C., et al.: (meta-) evaluation of machine translation. In: Proc. of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic (2007) 136–158
18. Melamed, I.: Manual annotation of translational equivalence: The blinker project. Technical Report 98-07, Institute for Research in Cognitive Science (1998)
19. Fraser, A., Marcu, D.: Getting the structure right for word alignment: LEAF. In: Proc. of EMNLP-CoNLL'07. (2007) 51–60
20. Fraser, A., Marcu, D.: Measuring word alignment quality for statistical machine translation. Comp.Ling. **33** (2007) 293–303
21. Ayan, N.F., Dorr, B.J.: Going beyond AER: an extensive analysis of word alignments and their impact on MT. In: Proc. of CONLING/ACL'06. (2006) 9–16
22. Germann, U.: Aligned hansards of the 36th parliament of canada. http://www.isi.edu/natural-language/download/hansard/index.html (2001)
23. Och, F., Ney, H.: Improved statistical alignment models. In: Proc. of ACL. (2000) 440–447
24. Koehn, P., et al.: Moses: Open source toolkit for statistical machine translation. In: Proc. of ACL'07: Demo and Poster Sessions. (2007) 177–180