Better Alignments = Better Translations?

Kuzman Ganchev
Computer & Information Science
University of Pennsylvania

João V. Graça
L^{2}F INESC-ID
Lisboa, Portugal

Ben Taskar
Computer & Information Science
University of Pennsylvania

Abstract

Automatic word alignment is a key step in training statistical machine translation systems. Despite much recent work on word alignment methods, alignment accuracy increases often produce little or no improvements in machine translation quality. In this work we analyze a recently proposed agreement-constrained EM algorithm for unsupervised alignment models. We attempt to tease apart the effects that this simple but effective modification has on alignment precision and recall trade-offs, and how rare and common words are affected across several language pairs. We propose and extensively evaluate a simple method for using alignment models to produce alignments better-suited for phrase-based MT systems, and show significant gains (as measured by BLEU score) in end-to-end translation systems for six languages pairs used in recent MT competitions.

1 Introduction

The typical pipeline for a machine translation (MT) system starts with a parallel sentence-aligned corpus and proceeds to align the words in every sentence pair. The word alignment problem has received much recent attention, but improvements in standard measures of word alignment performance often do not result in better translations. Fraser and Marcu (2007) note that *none of the tens of papers published over the last five years has shown that significant decreases in alignment error rate (AER) result in significant increases in translation performance*. In this work, we show that by changing the way the word alignment models are trained and used, we can get not only improvements in alignment performance, but also in the performance of the MT system that uses those alignments.

We present extensive experimental results evaluating a new training scheme for unsupervised word alignment models: an extension of the Expectation Maximization algorithm that allows effective injection of additional information about the desired alignments into the unsupervised training process. Examples of such information include "one word should not translate to many words" or that directional translation models should agree. The general framework for the extended EM algorithm with posterior constraints of this type was proposed by (Graça et al., 2008). Our contribution is a large scale evaluation of this methodology for word alignments, an investigation of how the produced alignments differ and how they can be used to consistently improve machine translation performance (as measured by BLEU score) across many languages on training corpora with up to hundred thousand sentences. In 10 out of 12 cases we improve BLEU score by at least $\frac{1}{4}$ point and by more than 1 point in 4 out of 12 cases.

After presenting the models and the algorithm in Sections 2 and 3, in Section 4 we examine how the new alignments differ from standard models, and find that the new method consistently improves word alignment performance, measured either as alignment error rate or weighted F-score. Section 5 explores how the new alignments lead to consistent and significant improvement in a state of the art phrase base machine translation by using posterior decoding rather than Viterbi decoding. We propose a heuristic for tuning posterior decoding in the absence of annotated alignment data and show improvements over baseline systems for six different

language pairs used in recent MT competitions.

2 Statistical word alignment

Statistical word alignment (Brown et al., 1994) is the task identifying which words are translations of each other in a bilingual sentence corpus. Figure 2 shows two examples of word alignment of a sentence pair. Due to the ambiguity of the word alignment task, it is common to distinguish two kinds of alignments (Och and Ney, 2003). Sure alignments (S), represented in the figure as squares with borders, for single-word translations and possible alignments (P), represented in the figure as alignments without boxes, for translations that are either not exact or where several words in one language are translated to several words in the other language. Possible alignments can can be used either to indicated optional alignments, such as the translation of an idiom, or disagreement between annotators. In the figure red/black dots indicates correct/incorrect predicted alignment points.

2.1 Baseline word alignment models

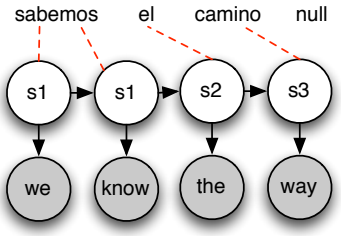
We focus on the hidden Markov model (HMM) for alignment proposed by (Vogel et al., 1996). This is a generalization of IBM models 1 and 2 (Brown et al., 1994), where the transition probabilities have a first-order Markov dependence rather than a zeroth-order dependence. The model is an HMM, where the hidden states take values from the source language words and generate target language words according to a translation table. The state transitions depend on the distance between the source language words. For source sentence $\mathbf{s}$ the probability of an alignment $\mathbf{a}$ and target sentence $\mathbf{t}$ can be expressed as:

$$p(\mathbf{t}, \mathbf{a} \mid \mathbf{s}) = \prod_j p_d(a_j | a_j - a_{j-1}) p_t(\mathbf{t}_j | \mathbf{s}_{a_j}), \quad (1)$$

where a_j is the index of the hidden state (source language index) generating the target language word at index j. As usual, a "null" word is added to the source sentence. Figure 1 illustrates the mapping between the usual HMM notation and the HMM alignment model.

2.2 Baseline training

All word alignment models we consider are normally trained using the Expectation Maximization (EM) algorithm (Dempster et al., 1977). The EM algorithm attempts to maximize the marginal likelihood of the observed data ($\mathbf{s}, \mathbf{t}$ pairs) by repeatedly finding a maximal lower bound on the likelihood and finding the maximal point of the lower bound. The lower bound is constructed by using posterior probabilities of the hidden alignments ($\mathbf{a}$) and can be optimized in closed form from expected sufficient statistics computed from the posteriors. For the HMM alignment model, these posteriors can be efficiently calculated by the Forward-Backward algorithm.



usual HMM		word alignment meaning
S_i	(hidden)	source language word i
O_j	(observed)	target language word j
a_{ij}	(transition)	distortion model
b_{ij}	(emission)	translation model

Figure 1: Illustration of an HMM for word alignment.

3 Adding agreement constraints

Graça et al. (2008) introduce an augmentation of the EM algorithm that uses constraints on posteriors to guide learning. Such constraints are useful for several reasons. As with any unsupervised induction method, there is no guarantee that the maximum likelihood parameters correspond to the intended meaning for the hidden variables, that is, more accurate alignments using the resulting model. Introducing additional constraints into the model often results in intractable decoding and search errors (e.g., IBM models 4+). The advantage of only constraining the posteriors during training is that the model remains simple while respecting more complex requirements. For example, constraints might include "one word should not translate to many words" or that translation is approximately symmetric.

The modification is to add a KL-projection step after the E-step of the EM algorithm. For each sentence pair instance $\mathbf{x} = (\mathbf{s}, \mathbf{t})$, we find the posterior

distribution $p_\theta(\mathbf{z}|\mathbf{x})$ (where $\mathbf{z}$ are the alignments). In regular EM, $p_\theta(\mathbf{z}|\mathbf{x})$ is used to complete the data and compute expected counts. Instead, we find the distribution q that is as close as possible to $p_\theta(\mathbf{z}|\mathbf{x})$ in KL subject to constraints specified in terms of expected values of features $\mathbf{f}(\mathbf{x}, \mathbf{z})$

$$\arg\min_q \mathrm{KL}(q(\mathbf{z}) \,||\, p_\theta(\mathbf{z}|\mathbf{x})) \;\text{ s.t. }\; \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] \leq \mathbf{b}. \tag{2}$$

The resulting distribution q is then used in place of $p_\theta(\mathbf{z}|\mathbf{x})$ to compute sufficient statistics for the M-step. The algorithm converges to a local maximum of the log of the marginal likelihood, $p_\theta(\mathbf{x}) = \sum_{\mathbf{z}} p_\theta(\mathbf{z}, \mathbf{x})$, penalized by the KL distance of the posteriors $p_\theta(\mathbf{z}|\mathbf{x})$ from the feasible set defined by the constraints (Graça et al., 2008):

$$\mathbf{E}_{\mathbf{x}}[\log p_\theta(\mathbf{x}) - \min_{q:\mathbf{E}_q[\mathbf{f}(\mathbf{x},\mathbf{z})]\leq\mathbf{b}} \mathrm{KL}(q(\mathbf{z}) \,||\, p_\theta(\mathbf{z}|\mathbf{x}))],$$

where $\mathbf{E}_{\mathbf{x}}$ is expectation over the training data. They suggest how this framework can be used to encourage two word alignment models to agree during training. We elaborate on their description and provide details of implementation of the projection in Equation 2.

3.1 Agreement

Most MT systems train an alignment model in each direction and then heuristically combine their predictions. In contrast, Graça et al. encourage the models to agree by training them concurrently. The intuition is that the errors that the two models make are different and forcing them to agree rules out errors only made by one model. This is best exhibited in the rare word alignments, where one-sided "garbage-collection" phenomenon often occurs (Moore, 2004). This idea was previously proposed by (Matusov et al., 2004; Liang et al., 2006) although the the objectives differ.

In particular, consider a feature that takes on value 1 whenever source word i aligns to target word j in the forward model and -1 in the backward model. If this feature has expected value 0 under the mixture of the two models, then the forward model and backward model agree on how likely source word i is to align to target word j. More formally denote the forward model $\overrightarrow{p}(\mathbf{z})$ and backward model $\overleftarrow{p}(\mathbf{z})$ where $\overrightarrow{p}(\mathbf{z}) = 0$ for $\mathbf{z} \notin \overrightarrow{\mathbf{Z}}$ and $\overleftarrow{p}(\mathbf{z}) = 0$ for $\mathbf{z} \notin \overleftarrow{\mathbf{Z}}$ ($\overrightarrow{\mathbf{Z}}$ and $\overleftarrow{\mathbf{Z}}$ are possible forward and backward alignments). Define a mixture $p(\mathbf{z}) = \frac{1}{2}\overrightarrow{p}(\mathbf{z}) + \frac{1}{2}\overleftarrow{p}(\mathbf{z})$ for $\mathbf{z} \in \overleftarrow{\mathbf{Z}} \cup \overrightarrow{\mathbf{Z}}$. Restating the constraints that enforce agreement in this setup: $\mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] = \mathbf{0}$ with

$$f_{ij}(\mathbf{x}, \mathbf{z}) = \begin{cases} 1 & \mathbf{z} \in \overrightarrow{\mathbf{Z}} \text{ and } z_{ij} = 1 \\ -1 & \mathbf{z} \in \overleftarrow{\mathbf{Z}} \text{ and } z_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

3.2 Implementation

EM training of hidden Markov models for word alignment is described elsewhere (Vogel et al., 1996), so we focus on the projection step:

$$\arg\min_q \mathrm{KL}(q(\mathbf{z}) \,||\, p_\theta(\mathbf{z}|\mathbf{x})) \;\text{ s.t. }\; \mathbf{E}_q[\mathbf{f}(\mathbf{x}, \mathbf{z})] = 0. \tag{3}$$

The optimization problem in Equation 3 can be efficiently solved in its dual formulation:

$$\arg\min_\lambda \log \sum_{\mathbf{z}} p_\theta(\mathbf{z} \mid \mathbf{x}) \exp\{\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\} \tag{4}$$

where we have solved for the primal variables q as:

$$q_\lambda(\mathbf{z}) = p_\theta(\mathbf{z} \mid \mathbf{x}) \exp\{\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\}/Z, \tag{5}$$

with Z a normalization constant that ensures q sums to one. We have only one dual variable per constraint, and we optimize them by taking a few gradient steps. The partial derivative of the objective in Equation 4 with respect to feature i is simply $\mathbf{E}_{q_\lambda}[\mathbf{f}_i(\mathbf{x}, \mathbf{z})]$. So we have reduced the problem to computing expectations of our features under the model q. It turns out that for the agreement features, this reduces to computing expectations under the normal HMM model. To see this, we have by the definition of q_λ and p_θ,

$$\begin{aligned} q_\lambda(\mathbf{z}) &= \frac{\overrightarrow{p}(\mathbf{z} \mid \mathbf{x}) + \overleftarrow{p}(\mathbf{z} \mid \mathbf{x})}{2} \exp\{\lambda^\top \mathbf{f}(\mathbf{x}, \mathbf{z})\}/Z \\ &= \frac{\overrightarrow{q}(\mathbf{z}) + \overleftarrow{q}(\mathbf{z})}{2}. \end{aligned}$$

(To make the algorithm simpler, we have assumed that the expectation of the feature $f_0(\mathbf{x}, \mathbf{z}) = \{1 \text{ if } \mathbf{z} \in \overrightarrow{\mathbf{Z}};\ -1 \text{ if } \mathbf{z} \in \overleftarrow{\mathbf{Z}}\}$ is set to zero to ensure that the two models $\overrightarrow{q}$, $\overleftarrow{q}$ are each properly normalized.) For $\overrightarrow{q}$, we have: ($\overleftarrow{q}$ is analogous)

$$\begin{aligned} & \overrightarrow{p}(\mathbf{z} \mid \mathbf{x}) e^{\lambda^\top \mathbf{f}(\mathbf{x},\mathbf{z})} \\ = & \prod_j \overrightarrow{p}_d(a_j | a_j - a_{j-1}) \overrightarrow{p}_t(\mathbf{t}_j | \mathbf{s}_{a_j}) \prod_{ij} e^{\lambda_{ij} f_{ij}(\mathbf{x}, z_{ij})} \\ = & \prod_{j, i=a_j} \overrightarrow{p}_d(i | i - a_{j-1}) \overrightarrow{p}_t(\mathbf{t}_j | \mathbf{s}_i) e^{\lambda_{ij} f_{ij}(\mathbf{x}, z_{ij})} \\ = & \prod_{j, i=a_j} \overrightarrow{p}_d(i | i - a_{j-1}) \overrightarrow{p}'_t(\mathbf{t}_j | \mathbf{s}_i). \end{aligned}$$

Where we have let $\overrightarrow{p}'_t(\mathbf{t}_j|\mathbf{s}_i) = \overrightarrow{p}_t(\mathbf{t}_j|\mathbf{s}_i)e^{\lambda_{ij}}$, and retained the same form for the model. The final projection step is detailed in Algorithm1.

Algorithm 1 AgreementProjection($\overrightarrow{p}$, $\overleftarrow{p}$)

$\lambda_{ij} \leftarrow 0 \ \forall i, j$
for T iterations **do**
  $\overrightarrow{p}'_t(j|i) \leftarrow \overrightarrow{p}_t(\mathbf{t}_j|\mathbf{s}_i)e^{\lambda_{ij}} \quad \forall i, j$
  $\overleftarrow{p}'_t(i|j) \leftarrow \overleftarrow{p}_t(\mathbf{s}_i|\mathbf{t}_j)e^{-\lambda_{ij}} \quad \forall i, j$
  $\overrightarrow{q} \leftarrow \text{forwardBackward}(\overrightarrow{p}'_t, \overrightarrow{p}_d)$
  $\overleftarrow{q} \leftarrow \text{forwardBackward}(\overleftarrow{p}'_t, \overleftarrow{p}_d)$
  $\lambda_{ij} \leftarrow \lambda_{ij} - \mathbf{E}_{\overrightarrow{q}}[a_i = j] + \mathbf{E}_{\overleftarrow{q}}[a_j = i] \ \forall i, j$
end for
return $(\overrightarrow{q}, \overleftarrow{q})$

3.3 Decoding

After training, we want to extract a single alignment from the distribution over alignments allowable for the model. The standard way to do this is to find the most probable alignment, using the Viterbi algorithm. Another alternative is to use posterior decoding. In posterior decoding, we compute for each source word i and target word j the posterior probability under our model that i aligns to j. If that probability is greater than some threshold, then we include the point $i - j$ in our final alignment. There are two main differences between posterior decoding and Viterbi decoding. First, posterior decoding can take better advantage of model uncertainty: when several likely alignment have high probability, posteriors accumulate confidence for the edges common to many good alignments. Viterbi, by contrast, must commit to one high-scoring alignment. Second, in posterior decoding, the probability that a target word aligns to none or more than one word is much more flexible: it depends on the tuned threshold.

4 Word alignment results

We evaluated the agreement HMM model on two corpora for which hand-aligned data are widely available: the Hansards corpus (Och and Ney, 2000) of English/French parliamentary proceedings and the Europarl corpus (Koehn, 2002) with EPPS annotation (Lambert et al., 2005) of English/Spanish. Figure 2 shows two machine-generated alignments

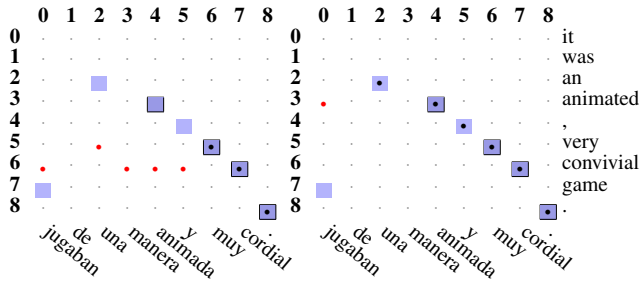


Figure 2: An example of the output of HMM trained on 100k the EPPS data. Left: Baseline training. Right: Using agreement constraints.

of a sentence pair. The black dots represent the machine alignments and the shading represents the human annotation (as described in the previous section), on the left using the regular HMM model and on the right using our agreement constraints. The figure illustrates a problem known as garbage collection (Brown et al., 1993), where rare source words tend to align to many target words, since the probability mass of the rare word translations can be hijacked to fit the sentence pair. Agreement constraints solve this problem, because forward and backward models cannot agree on the garbage collection solution.

Graça et al. (2008) show that alignment error rate (Och and Ney, 2003) can be improved with agreement constraints. Since AER is the standard metric for alignment quality, we reproduce their results using all the sentences of length at most 40. For the Hansards corpus we improve from 15.35 to 7.01 for the English → French direction and from 14.45 to 6.80 for the reverse. For English → Spanish we improve from 28.20 to 19.86 and from 27.54 to 19.18 for the reverse. These values are competitive with other state of the art systems (Liang et al., 2006).

Unfortunately, as was shown by Fraser and Marcu (2007) AER can have weak correlation with translation performance as measured by BLEU score (Papineni et al., 2002), when the alignments are used to train a phrase-based translation system. Consequently, in addition to AER, we focus on precision and recall.

Figure 3 shows the change in precision and recall with the amount of provided training data for the Hansards corpus. We see that agreement constraints improve both precision and recall when we

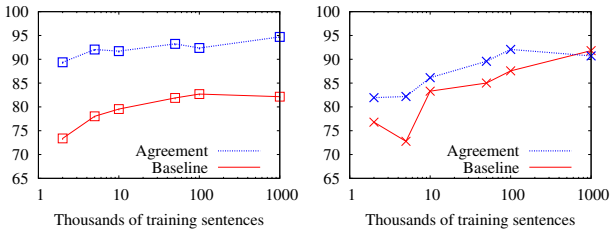


Figure 3: Effect of posterior constraints on precision (left) and recall (right) learning curves for Hansards En→Fr.

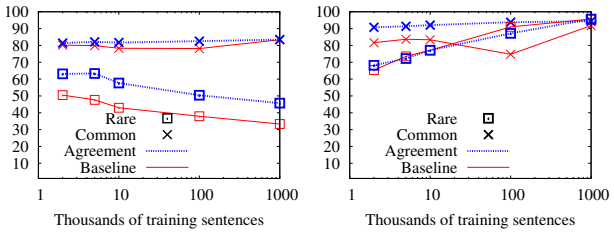


Figure 4: Left: Precision. Right: Recall. Learning curves for Hansards En→Fr split by rare (at most 5 occurances) and common words.

use Viterbi decoding, with larger improvements for small amounts of training data. We see a similar improvement on the EPPS corpus.

Motivated by the garbage collection problem, we also analyze common and rare words separately. Figure 4 shows precision and recall learning curves for rare and common words. We see that agreement constraints improve precision but not recall of rare words and improve recall but not precision of common words.

As described above an alternative to Viterbi decoding is to accept all alignments that have probability above some threshold. By changing the threshold, we can trade off precision and recall. Figure 5 compares this tradeoff for the baseline and agreement model. We see that the precision/recall curve for agreement is entirely above the baseline curve, so for any recall value we can achieve higher precision than the baseline for either corpus. In Figure 6 we break down the same analysis into rare and non rare words.

Figure 7 shows an example of the same sentence, using the same model where in one case Viterbi decoding was used and in the other case Posterior decoding tuned to minimize AER on a development set was used. An interesting difference is that by using

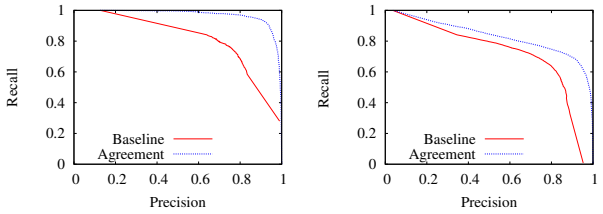


Figure 5: Precision and recall trade-off for posterior decoding with varying threshold. Left: Hansards En→Fr. Right: EPPS En→Es.

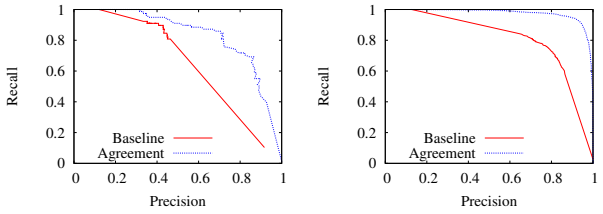


Figure 6: Precision and recall trade-off for posterior on Hansards En→Fr. Left: rare words only. Right: common words only.

posterior decoding one can have n-n alignments as shown in the picture.

A natural question is how to tune the threshold in order to improve machine translation quality. In the next section we evaluate and compare the effects of the different alignments in a phrase based machine translation system.

5 Phrase-based machine translation

In this section we attempt to investigate whether our improved alignments produce improved machine translation. In particular we fix a state of the art

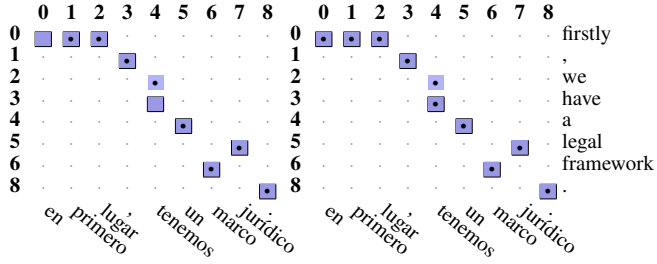


Figure 7: An example of the output of HMM trained on 100k the EPPS data using agreement HMM. Left: Viterbi decoding. Right: Posterior decoding tuned to minimize AER. The addition is *en-firstly* and *tenemos-have*.

Corpus	Train	Len	Test	Rare (%)	Unk (%)
En, Fr	1018	17.4	1000	0.3, 0.4	0.1, 0.2
En, Es	126	21.0	2000	0.3, 0.5	0.2, 0.3
En, Fi	717	21.7	2000	0.4, 2.5	0.2, 1.8
En, De	883	21.5	2000	0.3, 0.5	0.2, 0.3
En, Cz	57	23.0	2007	2.3, 6.6	1.3, 3.9
En, It	20	9.4	500	3.1, 6.2	1.4, 2.9

Table 1: Statistics of the corpora used in MT evaluation. The training size is measured in thousands of sentences and Len refers to average (English) sentence length. Test is the number of sentences in the test set. Rare and Unk are the percentage of tokens in the test set that are rare and unknown in the training data, for each language.

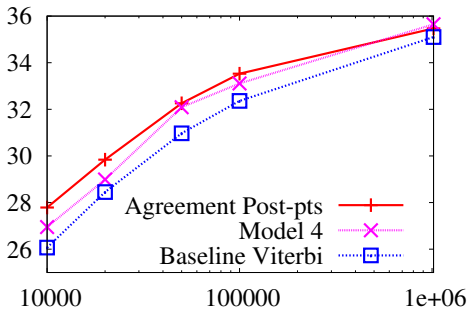


Figure 8: BLEU score as the amount of training data is increased on the Hansards corpus for the best decoding method for each alignment model.

machine translation system[1] and measure its performance when we vary the supplied word alignments. The baseline system uses GIZA model 4 alignments and the open source Moses phrase-based machine translation toolkit[2], and performed close to the best at the competition last year.

For all experiments the experimental setup is as follows: we lowercase the corpora, and train language models from all available data. The reasoning behind this is that even if bilingual texts might be scarce in some domain, monolingual text should be relatively abundant. We then train the competing alignment models and compute competing alignments using different decoding schemes. For each alignment model and decoding type we train Moses and use MERT optimization to tune its parameters on a development set. Moses is trained using the grow-diag-final-and alignment symmetrization heuristic and using the default distance base distortion model. We report BLEU scores using a script available with the baseline system. The competing alignment models are GIZA Model 4, our implementation of the baseline HMM alignment and our agreement HMM. We would like to stress that the fair comparison is between the performance of the baseline HMM and the agreement HMM, since Model 4 is more complicated and can capture more structure. However, we will see that for moderate sized data the agreement HMM performs better than both its baseline and GIZA Model 4.

5.1 Corpora

In addition to the Hansards corpus and the Europarl English-Spanish corpus, we used four other corpora for the machine translation experiments. Table 1 summarizes some statistics of all corpora. The German and Finnish corpora are also from Europarl, while the Czech corpus contains news commentary. All three were used in recent ACL workshop shared tasks and are available online[3]. The Italian corpus consists of transcribed speech in the travel domain and was used in the 2007 workshop on spoken language translation[4]. We used the development and tests sets from the workshops when available. For Italian corpus we used dev-set 1 as development and dev-set 2 as test. For Hansards we randomly chose 1000 and 500 sentences from test 1 and test 2 to be testing and development sets respectively.

Table 1 summarizes the size of the training corpus in thousands of sentences, the average length of the English sentences as well as the size of the testing corpus. We also report the percentage of tokens in the test corpus that are rare or not encountered in the training corpus.

5.2 Decoding

Our initial experiments with Viterbi decoding and posterior decoding showed that for our agreement model posterior decoding could provide better align-

[1] www.statmt.org/wmt07/baseline.html
[2] www.statmt.org/moses/
[3] http://www.statmt.org
[4] http://iwslt07.itc.it/

ment quality. When labeled data is available, we can tune the threshold to minimize AER. When labeled data is not available we use a different heuristic to tune the threshold: we choose a threshold that gives the same number of aligned points as Viterbi decoding produces. In principle, we would like to tune the threshold by optimizing BLEU score on a development set, but that is impractical for experiments with many pairs of languages. We call this heuristic posterior-points decoding. As we shall see, it performs well in practice.

5.3 Training data size

The HMM alignment models have a smaller parameter space than GIZA Model 4, and consequently we would expect that they would perform better when the amount of training data is limited. We found that this is generally the case, with the margin by which we beat model 4 slowly decreasing until a crossing point somewhere in the range of 10^5 - 10^6 sentences. We will see in section 5.3.1 that the Viterbi decoding performs best for the baseline HMM model, while posterior decoding performs best for our agreement HMM model. Figure 8 shows the BLEU score for the baseline HMM, our agreement model and GIZA Model 4 as we vary the amount of training data from 10^4 - 10^6 sentences. For all but the largest data sizes we outperform Model 4, with a greater margin at lower training data sizes. This trend continues as we lower the amount of training data further. We see a similar trend with other corpora.

5.3.1 Small to Medium Training Sets

Our next set of experiments look at our performance in both directions across our 6 corpora, when we have small to moderate amounts of training data: for the language pairs with more than 100,000 sentences, we use only the first 100,000 sentences. Table 2 shows the performance of all systems on these datasets. In the table, post-pts and post-aer stand for posterior-points decoding and posterior decoding tuned for AER. With the notable exception of Czech and Italian, our system performs better than or comparable to both baselines, even though it uses a much more limited model than GIZA's Model 4. The small corpora for which our models do not perform as well as GIZA are the ones with a lot of rare words. We suspect that the reason for this is that we do not implement smoothing, which has been shown to be important, especially in situations with a lot of rare words.

		X → En		En → X	
		Base	Agree	Base	Agree
De	GIZA M4	23.92		17.89	
	Viterbi	24.08	23.59	18.15	18.13
	post-pts	24.24	**24.65**$^{(+)}$	18.18	**18.45**$^{(+)}$
Fi	GIZA M4	18.29		11.05	
	Viterbi	18.79	18.38	11.17	11.54
	post-pts	18.88	**19.45**$^{(++)}$	11.47	**12.48**$^{(++)}$
Fr	GIZA M4	33.12		**26.90**	
	Viterbi	32.42	32.15	25.85	25.48
	post-pts	33.06	33.09$^{(\approx)}$	25.94	26.54$^{(+)}$
	post-aer	31.81	**33.53**$^{(+)}$	26.14	26.68$^{(+)}$
Es	GIZA M4	30.24		30.09	
	Viterbi	29.65	30.03	29.76	29.85
	post-pts	29.91	30.22$^{(++)}$	29.71	30.16$^{(+)}$
	post-aer	29.65	**30.34**$^{(++)}$	29.78	**30.20** $^{(+)}$
It	GIZA M4	51.66		**41.99**	
	Viterbi	**52.20**	52.09	41.40	41.28
	post-pts	51.06	51.14 $^{(--)}$	41.63	41.79 $^{(\approx)}$
Cz	GIZA M4	**22.78**		**12.75**	
	Viterbi	21.25	21.89	12.23	12.33
	post-pts	21.37	22.51 $^{(++)}$	12.16	12.47 $^{(+)}$

Table 2: BLEU scores for all language pairs using up to 100k sentences. Results are after MERT optimization. The marks $^{(++)}$and $^{(+)}$denote that agreement with posterior decoding is better by 1 BLEU point and 0.25 BLEU points respectively than the best baseline HMM model; analogously for $^{(--)}$, $^{(-)}$; while $^{(\approx)}$denotes smaller differences.

5.3.2 Larger Training Sets

For four of the corpora we have more than 100 thousand sentences. The performance of the systems on all the data is shown in Table 3. German is not included because MERT optimization did not complete in time. We see that even on over a million instances, our model sometimes performs better than GIZA model 4, and always performs better than the baseline HMM.

6 Conclusions

In this work we have evaluated agreement-constrained EM training for statistical word alignment models. We carefully studied its effects on word alignment recall and precision. Agreement training has a different effect on rare and common words, probably because it fixes different types

		X → En		En → X	
		Base	Agree	Base	Agree
Fi	GIZA M4	22.78		14.72	
	Viterbi	22.92	22.89	14.21	14.09
	post-pts	23.15	**23.43** $^{(+)}$	14.57	**14.74** $^{(\approx)}$
Fr	GIZA M4	**35.65**		**31.15**	
	Viterbi	35.19	35.17	30.57	29.97
	post-pts	35.49	35.95 $^{(+)}$	29.78	30.02 $^{(\approx)}$
	post-aer	34.85	35.48 $^{(+)}$	30.15	30.07 $^{(\approx)}$
Es	GIZA M4	**31.62**		**32.40**	
	Viterbi	31.75	31.84	31.17	31.09
	post-pts	31.88	32.19 $^{(+)}$	31.16	31.56 $^{(+)}$
	post-aer	31.93	32.29 $^{(+)}$	31.23	31.36 $^{(\approx)}$

Table 3: BLEU scores for all language pairs using all available data. The marks $^{(++)}$and $^{(+)}$denote that agreement with posterior decoding is better by 1 BLEU point and 0.25 BLEU points respectively than the best baseline HMM model; analogously for $^{(--)}$, $^{(-)}$; while $^{(\approx)}$denotes smaller differences.

of errors. It corrects the garbage collection problem for rare words, resulting in a higher precision. The recall improvement in common words can be explained by the idea that ambiguous common words are different in the two languages, so the un-ambiguous choices in one direction can force the choice for the ambiguous ones in the other through agreement constraints.

To our knowledge this is the first extensive evaluation where improvements in alignment accuracy lead to improvements in machine translation performance. We tested this hypothesis on six different language pairs from three different domains, and found that the new alignment scheme not only performs better than the baseline, but also improves over a more complicated, intractable model. In order to get the best results, it appears that posterior decoding is required for the simplistic HMM alignment model. The success of posterior decoding using our simple threshold tuning heuristic is fortunate since no labeled alignment data are needed: Viterbi alignments provide a reasonable estimate of aligned words needed for phrase extraction. The nature of the complicated relationship between word alignments, the corresponding extracted phrases and the effects on the final MT system still begs for better explanations and metrics. We have investigated the distribution of phrase-sizes used in translation across systems and languages, following recent investigations (Ayan and Dorr, 2006), but unfortunately found no consistent correlation with BLEU improvement. Since the alignments we extracted were better according to all metrics we used, it should not be too surprising that they yield better translation performance, but perhaps a better tradeoff can be achieved with a deeper understanding of the link between alignments and translations.

References

N. F. Ayan and B. J. Dorr. 2006. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proc. ACL*.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, M. J. Goldsmith, J. Hajic, R. L. Mercer, and S. Mohanty. 1993. But dictionaries are data too. In *Proc. HLT*.

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

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society, Ser. B*, 39(1):1–38.

A. Fraser and D. Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303.

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

P. Koehn. 2002. Europarl: A multilingual corpus for evaluation of machine translation.

P. Lambert, A.De Gispert, R. Banchs, and J. B. Mariño. 2005. Guidelines for word alignment evaluation and manual alignment. In *Language Resources and Evaluation, Volume 39, Number 4*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Proc. HLT-NAACL*.

E. Matusov, Zens. R., and H. Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proc. COLING*.

R. C. Moore. 2004. Improving IBM word-alignment model 1. In *Proc. ACL*.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *ACL*.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proc. ACL*.

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