Simultaneous Word-Morpheme Alignment for Statistical Machine Translation

Elif Eyigöz
Computer Science
University of Rochester
Rochester, NY 14627

Daniel Gildea
Computer Science
University of Rochester
Rochester, NY 14627

Kemal Oflazer
Computer Science
Carnegie Mellon University
PO Box 24866, Doha, Qatar

Abstract

Current word alignment models for statistical machine translation do not address morphology beyond merely splitting words. We present a two-level alignment model that distinguishes between words and morphemes, in which we embed an IBM Model 1 inside an HMM based word alignment model. The model jointly induces word and morpheme alignments using an EM algorithm. We evaluated our model on Turkish-English parallel data. We obtained significant improvement of BLEU scores over IBM Model 4. Our results indicate that utilizing information from morphology improves the quality of word alignments.

1 Introduction

All current state-of-the-art approaches to SMT rely on an automatically word-aligned corpus. However, current alignment models do not take into account the morpheme, the smallest unit of syntax, beyond merely splitting words. Since morphology has not been addressed explicitly in word alignment models, researchers have resorted to tweaking SMT systems by manipulating the content and the form of what should be the so-called "word".

Since the word is the smallest unit of translation from the standpoint of word alignment models, the central focus of research on translating morphologically rich languages has been decomposition of morphologically complex words into tokens of the right granularity and representation for machine translation. Chung and Gildea (2009) and Naradowsky and Toutanova (2011) use unsupervised methods to find word segmentations that create a one-to-one mapping of words in both languages. Al-Onaizan et al. (1999), Čmejrek et al. (2003), and Goldwater and McClosky (2005) manipulate morphologically rich languages by selective lemmatization. Lee (2004) attempts to learn the probability of deleting or merging Arabic morphemes for Arabic to English translation. Niessen and Ney (2000) split German compound nouns, and merge German phrases that correspond to a single English word. Alternatively, Yeniterzi and Oflazer (2010) manipulate words of the morphologically poor side of a language pair to mimic having a morphological structure similar to the richer side via exploiting syntactic structure, in order to improve the similarity of words on both sides of the translation.

We present an alignment model that assumes internal structure for words, and we can legitimately talk about words and their morphemes in line with the linguistic conception of these terms. Our model avoids the problem of collapsing words and morphemes into one single category. We adopt a two-level representation of alignment: the first level involves word alignment, the second level involves morpheme alignment in the scope of a given word alignment. The model jointly induces word and morpheme alignments using an EM algorithm.

We develop our model in two stages. Our initial model is analogous to IBM Model 1: the first level is a bag of words in a pair of sentences, and the second level is a bag of morphemes. In this manner, we embed one IBM Model 1 in the scope of another IBM Model 1. At the second stage, by introducing distortion probabilities at the word level, we develop an HMM extension of the initial model.

We evaluated the performance of our model on the

Proceedings of NAACL-HLT 2013, pages 32–40,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

Turkish-English pair both on hand-aligned data and by running end-to-end machine translation experiments. To evaluate our results, we created gold word alignments for 75 Turkish-English sentences. We obtain significant improvement of AER and BLEU scores over IBM Model 4. Section 2.1 introduces the concept of morpheme alignment in terms of its relation to word alignment. Section 2.2 presents the derivation of the EM algorithm and Section 3 presents the results of our experiments.

2 Two-level Alignment Model (TAM)

2.1 Morpheme Alignment

Following the standard alignment models of Brown et al. (1993), we assume one-to-many alignment for both words and morphemes. A word alignment a_w (or only a) is a function mapping a set of word positions in a source language sentence to a set of word positions in a target language sentence. A morpheme alignment a_m is a function mapping a set of morpheme positions in a source language sentence to a set of morpheme positions in a target language sentence. A morpheme position is a pair of integers (j,k), which defines a word position j and a relative morpheme position k in the word at position j. The alignments below are depicted in Figures 1 and 2.

$$a_w(1) = 1 \quad a_m(2,1) = (1,1) \quad a_w(2) = 1$$

Figure 1 shows a word alignment between two sentences. Figure 2 shows the morpheme alignment between same sentences. We assume that all unaligned morphemes in a sentence map to a special null morpheme.

A morpheme alignment a_m and a word alignment a_w are *compatible* if and only if they satisfy the following conditions: If the morpheme alignment a_m maps a morpheme of e to a morpheme of f, then the word alignment a_w maps e to f. If the word alignment a_w maps e to f, then the morpheme alignment a_m maps at least one morpheme of e to a morpheme of f. If the word alignment a_w maps e to null, then all of its morphemes are mapped to null. In sum, a morpheme alignment a_m and a word alignment a_w are *compatible* if and only if:

$$\begin{aligned} &\forall\, j,k,m,n \in \mathbb{N}^+,\ \exists\, s,t \in \mathbb{N}^+ \\ &[a_m(j,k) = (m,n) \Rightarrow a_w(j) = m] \wedge \\ &\qquad [a_w(j) = m \Rightarrow a_m(j,s) = (m,t)] \wedge \\ &\qquad\quad [a_w(j) = \text{null} \Rightarrow a_m(j,k) = \text{null}] \quad (1) \end{aligned}$$

Please note that, according to this definition of compatibility, '$a_m(j,k) = \text{null}$' does not necessarily imply '$a_w(j) = \text{null}$'.

A word alignment induces a set of compatible morpheme alignments. However, a morpheme alignment induces a unique word alignment. Therefore, if a morpheme alignment a_m and a word alignment a_w are compatible, then the word alignment is a_w is recoverable from the morpheme alignment a_m.

The two-level alignment model (TAM), like IBM Model 1, defines an alignment between words of a sentence pair. In addition, it defines a morpheme alignment between the morphemes of a sentence pair.

The problem domain of IBM Model 1 is defined over alignments between words, which is depicted as the gray box in Figure 1. In Figure 2, the smaller boxes embedded inside the main box depict the new problem domain of TAM. Given the word alignments in Figure 1, we are presented with a new alignment problem defined over their morphemes. The new alignment problem is constrained by the given word alignment. We, like IBM Model 1, adopt a bag-of-morphemes approach to this new problem. We thus embed one IBM Model 1 into the scope of another IBM Model 1, and formulate a second-order interpretation of IBM Model 1.

TAM, like IBM Model 1, assumes that words and morphemes are translated independently of their context. The units of translation are both words and morphemes. Both the word alignment a_w and the morpheme alignment a_m are hidden variables that need to be learned from the data using the EM algorithm.

In IBM Model 1, $p(\mathbf{e}|\mathbf{f})$, the probability of translating the sentence $\mathbf{f}$ into $\mathbf{e}$ with any alignment is computed by summing over all possible word alignments:

$$p(\mathbf{e}|\mathbf{f}) = \sum_a p(a, \mathbf{e}|\mathbf{f})$$

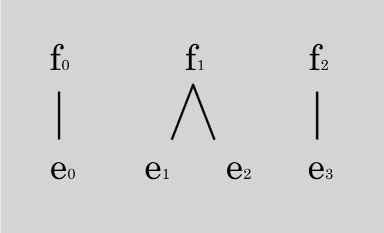

Figure 1: Word alignment

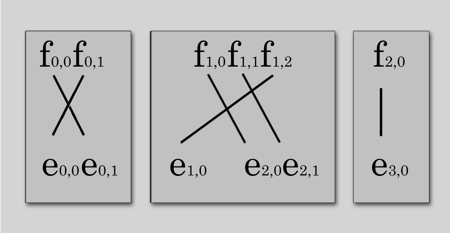

Figure 2: Morpheme alignment

In TAM, the probability of translating the sentence $\mathbf{f}$ into $\mathbf{e}$ with any alignment is computed by summing over all possible word alignments and all possible morpheme alignments that are compatible with a given word alignment a_w:

$$p(\mathbf{e}|\mathbf{f}) = \sum_{a_w} p(a_w, \mathbf{e}|\mathbf{f}) \sum_{a_m} p(a_m, \mathbf{e}|a_w, \mathbf{f}) \qquad (2)$$

where a_m stands for a morpheme alignment. Since the morpheme alignment a_m is in the scope of a given word alignment a_w, a_m is constrained by a_w.

In IBM Model 1, we compute the probability of translating the sentence $\mathbf{f}$ into $\mathbf{e}$ by summing over all possible word alignments between the words of $\mathbf{f}$ and $\mathbf{e}$:

$$p(\mathbf{e}|\mathbf{f}) = R(\mathbf{e}, \mathbf{f}) \prod_{j=1}^{|\mathbf{e}|} \sum_{i=0}^{|\mathbf{f}|} t(e_j|f_i) \qquad (3)$$

where $t(e_j \mid f_i)$ is the word translation probability of e_j given f_i. $R(\mathbf{e}, \mathbf{f})$ substitutes $\frac{P(l_\mathbf{e}|l_\mathbf{f})}{(l_\mathbf{f}+1)^{l_\mathbf{e}}}$ for easy readability.[1]

In TAM, the probability of translating the sentence $\mathbf{f}$ into $\mathbf{e}$ is computed as follows:

$$\overbrace{R(\mathbf{e}, \mathbf{f}) \prod_{j=1}^{|\mathbf{e}|} \sum_{i=0}^{|\mathbf{f}|} \Bigg(t(e_j|f_i)}^{\text{Word}} \underbrace{R(e_j, f_i) \prod_{k=1}^{|e_j|} \sum_{n=0}^{|f_i|} t(e_j^k|f_i^n) \Bigg)}_{\text{Morpheme}}$$

where f_i^n is the n^{th} morpheme of the word at position i. The right part of this equation, the contribution of morpheme translation probabilities, is in the scope of the left part. In the right part, we compute the probability of translating the word f_i into the word e_j by summing over all possible morpheme alignments between the morphemes of e_j and f_i. $R(e_j, f_i)$ is equivalent to $R(\mathbf{e}, \mathbf{f})$ except for the fact that its domain is not the set of sentences but the set of words. The length of words e_j and f_i in $R(e_j, f_i)$ are the number of morphemes of e_j and f_i.

The left part, the contribution of word translation probabilities alone, equals Eqn. 3. Therefore, canceling the contribution of morpheme translation probabilities reduces TAM to IBM Model 1. In our experiments, we call this reduced version of TAM 'word-only' (IBM). TAM with the contribution of both word and morpheme translation probabilities, as the equation above, is called 'word-and-morpheme'. Finally, we also cancel out the contribution of word translation probabilities, which is called 'morpheme-only'. In the 'morpheme-only' version of TAM, $t(e_j|f_i)$ equals 1. Bellow is the equation of $p(\mathbf{e}|\mathbf{f})$ in the morpheme-only model.

$$p(\mathbf{e}|\mathbf{f}) = \\ R(\mathbf{e}, \mathbf{f}) \prod_{j=1}^{|\mathbf{e}|} \sum_{i=0}^{|\mathbf{f}|} \prod_{k=1}^{|e_j|} \sum_{n=0}^{|f_i|} R(e_j, f_i) t(e_j^k|f_i^n) \qquad (4)$$

Please note that, although this version of the two-level alignment model does not use word translation probabilities, it is also a word-aware model, as morpheme alignments are restricted to correspond to a valid word alignment according to Eqn. 1. When presented with words that exhibit no morphology, the morpheme-only version of TAM is equivalent to IBM Model 1, as every single-morpheme word is itself a morpheme.

Deficiency and Non-Deficiency of TAM We present two versions of TAM, the word-and-

[1] $l_\mathbf{e} = |\mathbf{e}|$ is the number of words in sentence $\mathbf{e}$ and $l_\mathbf{f}$ = $|\mathbf{f}|$.

morpheme and the morpheme-only versions. The word-and-morpheme version of the model is deficient whereas the morpheme-only model is not.

The word-and-morpheme version is deficient, because some probability is allocated to cases where the morphemes generated by the morpheme model do not match the words generated by the word model. Moreover, although most languages exhibit morphology to some extent, they can be input to the algorithm without morpheme boundaries. This also causes deficiency in the word-and-morpheme version, as single morpheme words are generated twice, as a word and as a morpheme.

Nevertheless, we observed that the deficient version of TAM can perform as good as the non-deficient version of TAM, and sometimes performs better. This is not surprising, as deficient word alignment models such as IBM Model 3 or discriminative word alignment models work well in practice.

Goldwater and McClosky (2005) proposed a morpheme aware word alignment model for language pairs in which the source language words correspond to only one morpheme. Their word alignment model is:

$$P(e|f) = \prod_{k=0}^{K} P(e^k|f)$$

where e^k is the k^{th} morpheme of the word e. The morpheme-only version of our model is a generalization of this model. However, there are major differences in their and our implementation and experimentation. Their model assumes a fixed number of possible morphemes associated with any stem in the language, and if the morpheme e^k is not present, it is assigned a null value.

The null word on the source side is also a null morpheme, since every single morpheme word is itself a morpheme. In TAM, the null word is the null morpheme that all unaligned morphemes align to.

2.2 Second-Order Counts

In TAM, we collect counts for both word translations and morpheme translations. Unlike IBM Model 1, $R(e, f) = \frac{P(l_e|l_f)}{(l_f+1)^{l_e}}$ does not cancel out in the counts of TAM. To compute the conditional probability $P(l_e|l_f)$, we assume that the length of word e (the number of morphemes of word e) varies according to a Poisson distribution with a mean that is linear with length of the word f.

$$\begin{aligned} P(l_e|l_f) &= F_{\text{Poisson}}(l_e, r \cdot l_f) \\ &= \frac{\exp(-r \cdot l_f)(r \cdot l_f)^{l_e}}{l_e!} \end{aligned}$$

$F_{\text{Poisson}}(l_e, r \cdot l_f)$ expresses the probability that there are l_e morphemes in e if the expected number of morphemes in e is $r \cdot l_f$, where $r = \frac{\mathbb{E}[l_e]}{\mathbb{E}[l_f]}$ is the rate parameter. Since l_f is undefined for null words, we omit $R(e, f)$ for null words.

We introduce $T(e|f)$, the translation probability of e given f with all possible morpheme alignments, as it will occur frequently in the counts of TAM:

$$T(e|f) = t(e|f)R(e, f) \prod_{k=1}^{|e|} \sum_{n=0}^{|f|} t(e^k|f^n)$$

The role of $T(e|f)$ in TAM is very similar to the role of $t(e|f)$ in IBM Model 1. In finding the Viterbi alignments, we do not take max over the values in the summation in $T(e|f)$.

2.2.1 Word Counts

Similar to IBM Model 1, we collect counts for word translations over all possible alignments, weighted by their probability. In Eqn. 5, the count function collects evidence from a sentence pair $(\mathbf{e}, \mathbf{f})$ as follows: For all words e_j of the sentence $\mathbf{e}$ and for all word alignments $a_w(j)$, we collect counts for a particular input word f and an output word e iff $e_j = e$ and $f_{a_w(j)} = f$.

$$c_w(e|f; \mathbf{e}, \mathbf{f}, a_w) = \sum_{\substack{1 \le j \le |\mathbf{e}| \\ \text{s.t.} \\ e = e_j \\ f = f_{a_w(j)}}} \frac{T(e|f)}{\sum_{i=0}^{|f|} T(e|f_i)} \tag{5}$$

2.2.2 Morpheme Counts

As for morpheme translations, we collect counts over all possible word and morpheme alignments, weighted by their probability. The morpheme count function below collects evidence from a word pair (e, f) in a sentence pair $(\mathbf{e}, \mathbf{f})$ as follows: For all words e_j of the sentence $\mathbf{e}$ and for all word alignments $a_w(j)$, for all morphemes e_j^k of the word e_j and for all morpheme alignments $a_m(j, k)$, we collect counts for a particular input morpheme g and an output morpheme h iff $e_j = e$ and $f_{a_w(j)} = f$ and $h = e_j^k$ and $g = f_{a_m(j,k)}$.

$$c_m(h|g; \mathbf{e}, \mathbf{f}, a_w, a_m) = \sum_{\substack{1 \le j \le |\mathbf{e}| \\ \text{s.t.} \\ e = e_j \\ f = f_{a_w(j)}}} \sum_{\substack{1 \le k \le |e| \\ \text{s.t.} \\ h = e_j^k \\ g = f_{a_m(j,k)}}} \frac{T(e|f)}{\sum_{i=0}^{|\mathbf{f}|} T(e|f_i)} \frac{t(h|g)}{\sum_{i=1}^{|f|} t(h|f^i)}$$

The left part of the morpheme count function is the same as the word-counts in Eqn. 5. Since it does not contain h or g, it needs to be computed only once for each word. The right part of the equation is familiar from the IBM Model 1 counts.

2.3 HMM Extension

We implemented TAM with the HMM extension (Vogel et al., 1996) at the word level. We redefine $p(\mathbf{e}|\mathbf{f})$ as follows:

$$R(\mathbf{e}, \mathbf{f}) \sum_{a_w} \prod_{j=1}^{|e|} \Bigg(p(s(j) \mid C(f_{a_w(j-1)}))\, t(e_j|f_{a_w(j)}) \\ R(e_j, f_{a_w(j)}) \sum_{a_m} \prod_{k=1}^{|e_j|} t(e_j^k|f_{a_m(j,k)}) \Bigg)$$

where the distortion probability depends on the relative jump width $s(j) = a_w(j-1) - a_w(j)$, as opposed to absolute positions. The distortion probability is conditioned on class of the previous aligned word $C(f_{a_w(j-1)})$. We used the `mkcls` tool in GIZA (Och and Ney, 2003) to learn the word classes.

We formulated the HMM extension of TAM only at the word level. Nevertheless, the morpheme-only version of TAM also has an HMM extension, as it is also a word-aware model. To obtain the HMM extension of the morpheme-only version, substitute $t(e_j|f_{a_w(j)})$ with 1 in the equation above.

For the HMM to work correctly, we must handle jumping to and jumping from null positions. We learn the probabilities of jumping to a null position from the data. To compute the jump probability from a null position, we keep track of the nearest previous source word that does not align to null, and use the position of the previous non-null word to calculate the jump width. For this reason, we use a total of $2l_{\mathbf{f}} - 1$ words for the HMM model, the positions $> l_f$ stand for null positions between the words of f (Och and Ney, 2003). We do not allow null to null jumps. In sum, we enforce the following constraints:

$$\begin{aligned} P(i + l_f + 1|i') &= p(null|i') \\ P(i + l_f + 1|i' + l_f + 1) &= 0 \\ P(i|i' + l_f + 1) &= p(i|i') \end{aligned}$$

In the HMM extension of TAM, we perform forward-backward training using the word counts in Eqn. 5 as the emission probabilities. We calculate the posterior word translation probabilities for each e_j and f_i such that $1 \le j \le l_{\mathbf{e}}$ and $1 \le i \le 2l_{\mathbf{f}} - 1$ as follows:

$$\gamma_j(i) = \frac{\alpha_j(i)\beta_j(i)}{\sum_{m=1}^{2l_f - 1} \alpha_j(m)\beta_j(m)}$$

where α is the forward and β is the backward probabilities of the HMM. The HMM word counts, in turn, are the posterior word translation probabilities obtained from the forward-backward training:

$$c_w(e|f; \mathbf{e}, \mathbf{f}, a_w) = \sum_{\substack{1 \le j \le |\mathbf{e}| \\ \text{s.t.} \\ e = e_j \\ f = f_{a_w(j)}}} \gamma_j(a_w(j))$$

Likewise, we use the posterior probabilities in HMM morpheme counts:

$$c_m(h|g; \mathbf{e}, \mathbf{f}, a_w, a_m) = \sum_{\substack{1 \le j \le |\mathbf{e}| \\ \text{s.t.} \\ e = e_j \\ f = f_{a_w(j)}}} \sum_{\substack{1 \le k \le |e| \\ \text{s.t.} \\ h = e_j^k \\ g = f_{a_m(j,k)}}} \gamma_j(a_w(j)) \frac{t(h|g)}{\sum_{i=1}^{|f|} t(h|f^i)}$$

The complexity of the HMM extension of TAM is $O(n^3 m^2)$, where n is the number of words, and m is the number of morphemes per word.

2.4 Variational Bayes

Moore (2004) showed that the EM algorithm is particularly susceptible to overfitting in the case of rare words when training IBM Model 1. In order to prevent overfitting, we use the Variational Bayes extension of the EM algorithm (Beal, 2003). This

(a) Kasım 1996’da, Türk makamları, İçişleri Bakanlığı bünyesinde bir kayıp kişileri arama birimi oluşturdu.

(b) Kasım+Noun 1996+Num–Loc ,+Punc Türk+Noun makam+Noun–A3pl–P3sg ,+Punc İçişi+Noun–A3pl–P3sg Bakanlık+Noun–P3sg bünye+Noun–P3sg–Loc bir+Det kayıp+Adj kişi+Noun–A3pl–Acc ara+Verb–Inf2 birim+Noun–P3sg oluş+Verb–Caus–Past .+Punc

(c) In November 1996 the Turkish authorities set up a missing persons search unit within the Ministry of the Interior.

(d) in+IN November+NNP 1996+CD the+DT Turkish+JJ **author+NN–ity+N|N.–NNS** set+VB–VBD up+RP a+DT miss+VB–VBG+JJ **person+NN–NNS** search+NN unit+NN within+IN the+DT minister+NN–y+N|N. of+IN the+DT interior+NN .+.

(e) In+IN November+NNP 1996+CD the+DT Turkish+JJ authorities+NNS set+VBD up+RP a+DT missing+JJ persons+NNS search+NN unit+NN within+IN the+DT Ministry+NNP of+IN the+DT Interior+NNP .+.

Figure 3: Turkish-English data examples

amounts to a small change to the M step of the original EM algorithm. We introduce Dirichlet priors α to perform an inexact normalization by applying the function $f(v) = \exp(\psi(v))$ to the expected counts collected in the E step, where ψ is the digamma function (Johnson, 2007).

$$\theta_{x|y} = \frac{f(E[c(x|y)] + \alpha)}{f(\sum_j E[c(x_j|y)] + \alpha)}$$

We set α to 10^{-20}, a very low value, to have the effect of anti-smoothing, as low values of α cause the algorithm to favor words which co-occur frequently and to penalize words that co-occur rarely.

3 Experimental Setup

3.1 Data

We trained our model on a Turkish-English parallel corpus of approximately 50K sentences, which have a maximum of 80 morphemes. Our parallel data consists mainly of documents in international relations and legal documents from sources such as the Turkish Ministry of Foreign Affairs, EU, etc. We followed a heavily supervised approach in morphological analysis. The Turkish data was first morphologically parsed (Oflazer, 1994), then disambiguated (Sak et al., 2007) to select the contextually salient interpretation of words. In addition, we removed morphological features that are not explicitly marked by an overt morpheme — thus each feature symbol beyond the root part-of-speech corresponds to a morpheme. Line (b) of Figure 3 shows an example of a segmented Turkish sentence. The root is followed by its part-of-speech tag separated by a ‘+’. The derivational and inflectional morphemes that follow the root are separated by ‘–’s. In all experiments, we used the same segmented version of the Turkish data, because Turkish is an agglutinative language.

For English, we used the CELEX database (Baayen et al., 1995) to segment English words into morphemes. We created two versions of the data: a segmented version that involves both derivational and inflectional morphology, and an unsegmented POS tagged version. The CELEX database provides tags for English derivational morphemes, which indicate their function: the part-of-speech category the morpheme attaches to and the part-of-speech category it returns. For example, in ‘sparse+ity’ = ‘sparsity’, the morpheme *-ity* attaches to an adjective to the right and returns a noun. This behavior is represented as ‘N|A.’ in CELEX, where ‘**.**’ indicates the attachment position. We used these tags in addition to the surface forms of the English morphemes, in order to disambiguate multiple functions of a single surface morpheme.

The English sentence in line (d) of Figure 3 exhibits both derivational and inflectional morphology. For example, ‘author+ity+s’=‘authorities’ has both an inflectional suffix *-s* and a derivational suffix *-ity*, whereas ‘person+s’ has only an inflectional suffix *-s*.

For both English and Turkish data, the dashes in Figure 3 stand for morpheme boundaries, therefore the strings between the dashes are treated as indi-

	Words		Morphemes	
	Tokens	Types	Tokens	Types
English Der+Inf	1,033,726	27,758	1,368,188	19,448
English POS	1,033,726	28,647	1,033,726	28,647
Turkish Der+Inf	812,374	57,249	1,484,673	16,713

Table 1: Data statistics

visible units. Table 1 shows the number of words, the number of morphemes and the respective vocabulary sizes. The average number of morphemes in segmented Turkish words is 2.69, and the average length of segmented English words is 1.57.

3.2 Experiments

We initialized our baseline word-only model with 5 iterations of IBM Model 1, and further trained the HMM extension (Vogel et al., 1996) for 5 iterations. We call this model 'baseline HMM' in the discussions. Similarly, we initialized the two versions of TAM with 5 iterations of the model explained in Section 2.2, and then trained the HMM extension of it as explained in Section 2.3 for 5 iterations.

To obtain BLEU scores for TAM models and our implementation of the word-only model, i.e. baseline-HMM, we bypassed GIZA++ in the Moses toolkit (Och and Ney, 2003). We also ran GIZA++ (IBM Model 1–4) on the data. We translated 1000 sentence test sets.

4 Results and Discussion

We evaluated the performance of our model in two different ways. First, we evaluated against gold word alignments for 75 Turkish-English sentences. Second, we used the word Viterbi alignments of our algorithm to obtain BLEU scores.

Table 2 shows the AER (Och and Ney, 2003) of the word alignments of the Turkish-English pair and the translation performance of the word alignments learned by our models. We report the grow-diag-final (Koehn et al., 2003) of the Viterbi alignments. In Table 2, results obtained with different versions of the English data are represented as follows: 'Der' stands for derivational morphology, 'Inf' for inflectional morphology, and 'POS' for part-of-speech tags. 'Der+Inf' corresponds to the example sentence in line (d) in Figure 3, and 'POS' to line (e). 'DIR' stands for models with Dirichlet priors, and 'NO DIR' stands for models without Dirichlet priors. All reported results are of the HMM extension of respective models.

Table 2 shows that using Dirichlet priors hurts the AER performance of the word-and-morpheme model in all experiment settings, and benefits the morpheme-only model in the POS tagged experiment settings.

In order to reduce the effect of nondeterminism, we run Moses three times per experiment setting, and report the highest BLEU scores obtained. Since the BLEU scores we obtained are close, we did a significance test on the scores (Koehn, 2004). Table 2 visualizes the partition of the BLEU scores into statistical significance groups. If two scores within the same column have the same background color, or the border between their cells is removed, then the difference between their scores is not statistically significant. For example, the best BLEU scores, which are in bold, have white background. All scores in a given experiment setting without white background are significantly worse than the best score in that experiment setting, unless there is no border separating them from the best score.

In all experiment settings, the TAM Models perform better than the baseline-HMM. Our experiments showed that the baseline-HMM benefits from Dirichlet priors to a larger extent than the TAM models. Dirichlet priors help reduce the overfitting in the case of rare words. The size of the word vocabulary is larger than the size of the morpheme vocabulary. Therefore the number of rare words is larger for words than it is for morphemes. Consequently, baseline-HMM, using only the word vocab-

			BLEU EN to TR		BLEU TR to EN		AER	
			Der+Inf	POS	Der+Inf	POS	Der+Inf	POS
NO DIR	TAM	Morph only	**22.57**	**22.54**	**29.30**	**29.45**	0.293	0.276
		Word & Morph	21.95	22.37	28.81	29.01	0.286	0.282
	WORD	IBM 4	21.82	21.82	27.91	27.91	0.357	0.370
		Base-HMM	21.78	21.38	28.22	28.02	0.381	0.375
		IBM 4 Morph	17.15	17.94	25.70	26.33	N/A	N/A
DIR	TAM	Morph only	22.18	**22.52**	**29.32**	**29.98**	0.304	0.256
		Word & Morph	**22.43**	21.62	29.21	29.11	0.338	0.317
	WORD	IBM 4	21.82	21.82	27.91	27.91	0.357	0.370
		Base-HMM	21.69	21.95	28.76	29.13	0.381	0.377
		IBM 4 Morph	17.15	17.94	25.70	26.33	N/A	N/A

Table 2: AER and BLEU Scores

ulary, benefits from the use of Dirichlet priors more than the TAM models.

In four out of eight experiment settings, the morpheme-only model performs better than the word-and-morpheme version of TAM. However, please note that our extensive experimentation with TAM models revealed that the superiority of the morpheme-only model over the word-and-morpheme model is highly dependent on segmentation accuracy, degree of segmentation, and morphological richness of languages.

Finally, we treated morphemes as words and trained IBM Model 4 on the morpheme segmented versions of the data. To obtain BLEU scores, we had to unsegment the translation output: we concatenated the prefixes to the morpheme to the right, and suffixes to the morpheme to the left. Since this process creates malformed words, the BLEU scores obtained are much lower than the scores obtained by IBM Model 4, the baseline and the TAM Models.

5 Conclusion

We presented two versions of a two-level alignment model for morphologically rich languages. We observed that information provided by word translations and morpheme translations interact in a way that enables the model to be receptive to the partial information in rarely occurring words through their frequently occurring morphemes. We obtained significant improvement of BLEU scores over IBM Model 4. In conclusion, morphologically aware word alignment models prove to be superior to their word-only counterparts.

Acknowledgments Funded by NSF award IIS-0910611. Kemal Oflazer acknowledges the generous support of the Qatar Foundation through Carnegie Mellon University's Seed Research program. The statements made herein are solely the responsibility of this author(s), and not necessarily that of Qatar Foundation.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation.

Technical report, Final Report, JHU Summer Workshop.

R.H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database (Release 2) [CD-ROM]*. Linguistic Data Consortium, University of Pennsylvania [Distributor], Philadelphia, PA.

Matthew J. Beal. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, University College London.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Tagyoung Chung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In *EMNLP*, pages 718–726.

Martin Čmejrek, Jan Cuřín, and Jiří Havelka. 2003. Czech-English dependency-based machine translation. In *EACL*, pages 83–90, Morristown, NJ, USA. Association for Computational Linguistics.

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *HLT-EMNLP*.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *EMNLP-CoNLL*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395.

Young-suk Lee. 2004. Morphological analysis for statistical machine translation. In *HLT-NAACL*, pages 57–60.

Robert C. Moore. 2004. Improving IBM word alignment model 1. In *ACL*, pages 518–525, Barcelona, Spain, July.

Jason Naradowsky and Kristina Toutanova. 2011. Unsupervised bilingual morpheme segmentation and alignment with context-rich Hidden Semi-Markov Models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 895–904, Portland, Oregon, USA, June. Association for Computational Linguistics.

Sonja Niessen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Computational Linguistics*, pages 1081–1085, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2).

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological disambiguation of Turkish text with perceptron algorithm. In *CICLing*, pages 107–118, Berlin, Heidelberg. Springer-Verlag.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*, pages 836–841.

Reyyan Yeniterzi and Kemal Oflazer. 2010. Syntax-to-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In *ACL*, pages 454–464, Stroudsburg, PA, USA. Association for Computational Linguistics.