Joint Prediction of Word Alignment with Alignment Types

Anahita Mansouri Bigvand and **Te Bu** and **Anoop Sarkar**
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
{amansour,tbu,anoop}@cs.sfu.ca

Abstract

Current word alignment models do not distinguish between different types of alignment links. In this paper, we provide a new probabilistic model for word alignment where word alignments are associated with linguistically motivated alignment types. We propose a novel task of joint prediction of word alignment and alignment types and propose novel semi-supervised learning algorithms for this task. We also solve a sub-task of predicting the alignment type given an aligned word pair. In our experimental results, the generative models we introduce to model alignment types significantly outperform the models without alignment types.

1 Introduction

Word alignment is a crucial component in a statistical machine translation (SMT) system. Soft alignments, or attention, is also a crucial component in neural machine translation (NMT) systems. The classic generative model approach to word alignment is based on IBM models 1-5 (Brown et al., 1993) and the HMM model (Vogel et al., 1996; Och and Ney, 2000a). These traditional models use unsupervised algorithms to learn alignments, relying on a large amount of parallel training data without hand annotated alignments. Supervised algorithms for word alignment have become more widespread with the availability of manually annotated word-aligned data and have shown promising results (Taskar et al., 2005; Blunsom and Cohn, 2006; Moore et al., 2006; Liang et al., 2006). Manually word-aligned data are valuable resources for SMT research, but they are costly to create and are only available for a handful of language pairs. Semi-supervised methods for word alignment combine data with hand-annotated word alignments with parallel data without explicit word alignments. Even small amounts of hand-annotated word alignment data has been shown to improve the alignment and translation quality (Callison-Burch et al., 2004). In this paper we provide a novel semi-supervised word alignment model that adds alignment type information to word alignments.

Unsupervised or semi-supervised probabilistic word alignment models do not play a central role in neural machine translation (NMT) (Bahdanau et al., 2014; Sutskever et al., 2014; Luong et al., 2015; Chung et al., 2016). However, attention models, which are crucial for high-quality NMT, have been augmented with ideas from statistical word alignment (Luong et al., 2015; Cohn et al., 2016). Word alignments are also crucially important in the best performing models for NLP tasks other than machine translation. They play a central role in learning paraphrases in a source language by doing round-trips from source to target and back using word alignments (Ganitkevitch et al., 2013). Aligments also form the basis for learning multi-lingual word embeddings (Faruqui and Dyer, 2014; Lu et al., 2015) and in the projection of syntactic and semantic annotations from one language to another (Hwa et al., 2005; McDonald et al., 2011). Therefore, there is still a prominent role for word alignment in NLP, and research into improvements in word alignment is a worthy goal.

Adding additional information such as part-of-speech tags and syntactic parse information has yielded some improvements in word alignment quality. Toutanova et al. (2002) incorporated the part-of-speech (POS) tags of the words in the sentence pair as a constraint on HMM-based word alignment. Additional constraints have also been injected into generative and discriminative models

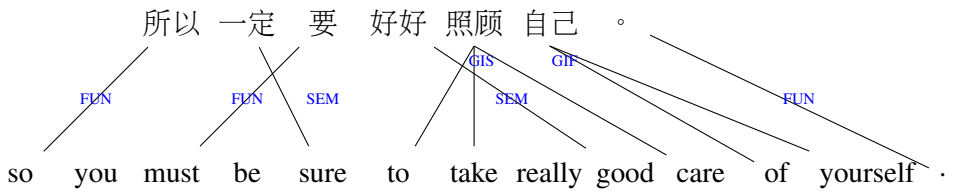


Figure 1: An alignment between a Chinese sentence and its translation in English which is enriched with alignment types . SEM (semantic), FUN (function), GIF (grammatically inferred function) and GIS (grammatically inferred semantic) are tags of the links.

by designing linguistically-motivated features (Ittycheriah and Roukos, 2005; Blunsom and Cohn, 2006; Deng and Gao, 2007; Berg-Kirkpatrick et al., 2010; Dyer et al., 2011). These models provide evidence that additional constraints can help in modelling word alignments in a log-linear model where word based features can be augmented with morphological, syntactic or semantic features. For example, such a model might learn that function words in one language tend to be aligned to function words in the other language.

In this paper, we propose a novel task which is the joint prediction of word alignment and alignment types for a given sentence pair in a parallel corpus. We present how to enhance the alignment model with alignment types. The primary contribution of this paper is to demonstrate the success of the proposed joint model (alignment-type-enhanced model) to improve word alignment and translation quality. We apply our method on Chinese-English, because the annotated alignment type training data is provided in this language pair. However, the proposed method is potentially language-independent and can be applied to any language-pair as long as alignment type annotated data is created. The alignment types themselves may be language dependent and may vary in different language pairs.

2 The Data Set

The Linguistic Data Consortium (LDC) developed a linguistically-enriched word alignment data set: the GALE Chinese-English Word Alignment and Tagging Corpus. This human annotated data set adds alignment type information to word alignments. The goal was to sub-categorize different types of alignment and draw a distinction between different types of alignment. For instance, it makes a distinction between aligned function words in both languages versus aligned content words. The goal was to improve word alignment and translation quality. Figure 1 shows an example of an enriched word alignment with alignment types extracted from the LDC data. Each link tag in the figure demonstrates the alignment type between its constituents.

The GALE Chinese-English Word Alignment and Tagging corpus contains 22313 manually word aligned sentence pairs from which we extracted 20357 sentences for training and we kept the rest as a test set. Table 1 shows the type and number of each alignment type in our training data.

Id	Alignment Type	Count
1	SEM	159,277
2	GIS	81,235
3	FUN	97,727
4	GIF	12,314
5	PDE	1,421
6	COI	3,256
7	CDE	1,608
8	TIN	1,116
9	MDE	4,615
10	NTR	34,090
11	MTA	84

Table 1: Number of each alignment type in the annotated training data

We briefly explain the existing alignment types in the GALE Chinese-English Word Alignment and Tagging Corpus. The SEM tag represents a semantic link between content words/phrases of source and translation, indicating a direct equivalence. Content words are typically nouns, verbs, adjectives and ad-

verbs. FUN refers to a Function link which indicates that a word on either side of the link is a function word. Grammatically Inferred Function (GIF) link is a type of link in which by stripping off extra words, we get a pure function link. In Grammatically Inferred Semantic (GIS) links, stripping off extra words results in pure semantic links. Alignment types PDE (DE-possesive), CDE (DE-Clause) and MDE (DE-modifier) are designed to handle the different features of the Chinese word 的(DE). In Contextually Inferred (COI) links, the extra words attached to one side of the link are required. Without these words, the grammatical structure might still be acceptable, but it is not semantically sensible. TIN (Translated Incorrectly) and NTR (Not translated) types are designed to handle the various errors that occur in the translation process, such as incorrect translation and no translation. MTA (Meta word) was designed to handle special characters that usually appear in the context of web pages.

Sub-categorizing different types of word alignments is likely to result in better word alignments. The alignment types provided by the LDC as annotations on each word alignment link, have never been used (as far as we are aware) in order to improve word alignment. A subset of this data was used in (Wang et al., 2014) to refine word segmentation for machine translation but they ignore the alignment link types in their experiments.

3 Word Alignment

Given a source sentence $\mathbf{f} = \{f_1, f_2, \ldots, f_J\}$ and a target sentence $\mathbf{e} = \{e_1, e_2, \ldots, e_I\}$, the goal in SMT is to model the translation probability $Pr(\mathbf{f}|\mathbf{e})$. In alignment models, a hidden variable $\mathbf{a} = \{a_1, a_2, \ldots, a_J\}$ is introduced which describes a mapping between source and target words. Using this terminology, $a_j = i$ denotes that f_j is aligned to e_i. The translation probability can therefore be written as a marginal probability over all alignments:

$$Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \quad (1)$$

In IBM Model 1, the alignment model is decomposed into the product of translation probabilities as follows:

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} p(f_j|e_{a_j}) \quad (2)$$

In the Hidden Markov alignment model, we assume a first order dependence for the alignments a_j. The HMM-based model has the following form:

$$Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} p(a_j|a_{j-1}, I) \cdot p(f_j|e_{a_j}) \quad (3)$$

where $p(a_j|a_{j-1}, I)$ are the alignment probabilities (transition parameters) and $p(f_j|e_{a_j})$ are the translation probabilities (emission parameters). Vogel et al. (1996) make the alignment parameters $p(i|i', I)$ independent of the absolute word positions and assume that $p(i|i', I)$ depend only on the jump width $(i - i')$. Hence, the alignment probabilities are estimated using a set of distortion parameters $c(i - i')$ as follows:

$$p(i|i', I) = \frac{c(i - i')}{\sum_{i''=1}^{I} c(i'' - i')} \quad (4)$$

where at each EM iteration $c(i - i')$ is the fractional count of transitions with jump width $i - i'$.

The HMM network is extended by I NULL words (Och and Ney, 2000a) with the following constraints on the transition probabilities ($i \leq I, i' \leq I$):

$$p(i + I|i', I) = p_0.\delta(i, i') \quad (5)$$

$$p(i + I|i' + I, I) = p_0.\delta(i, i') \quad (6)$$

$$p(i|i' + I, I) = p(i|i', I) \quad (7)$$

The parameter p_0 controls NULL insertion and is optimized on a held-out dataset.

4 Joint Model for IBM Model 1 and HMM

We consider two classic generative models, IBM Model 1 (Brown et al., 1993) and the HMM alignment model (Vogel et al., 1996) as our baselines and present how we can enhance these models with alignment types. In this section, we introduce two models (a generative and a discriminative model) for each baseline to jointly find the word alignments and the corresponding alignment types for a sentence pair.

4.1 Generative Models

4.1.1 IBM Model 1 with Alignment Types

We augment IBM Model 1 (Equation 2) with alignment type information. In addition to alignment function $a : j \rightarrow i$, our model has a tagging function: $h : j \rightarrow k$ which specifies the mapping for each alignment link (f_j, e_i) to an alignment type k. Alignment type k can be any tag in the set of all possible linguistic tags. The new generative model with alignment type has the following form:

$$Pr(\mathbf{f}, \mathbf{a}, \mathbf{h}|\mathbf{e}) = \frac{1}{(I+1)^J N^J} \prod_{j=1}^{J} p(f_j, h_j|e_{a_j}) \quad (8)$$

where N is the number of possible linguistic alignment types. Using the chain rule, we have the following enhanced IBM Model 1 which includes alignment-types:

$$Pr(\mathbf{f}, \mathbf{a}, \mathbf{h}|\mathbf{e}) = \frac{1}{(I+1)^J N^J} \times \prod_{j=1}^{J} p(f_j|e_{a_j}) \cdot p(h_j|f_j, e_{a_j}) \quad (9)$$

In order to normalize the probability, we modify the fraction in Equation 2 by adding term N^J as there are N different alignment types for each alignment link from each source word.

4.1.2 EM algorithm

Similar to IBM Model 1, we use EM algorithm to estimate the parameters of our model. In the expectation step, we need to compute the posterior probability $Pr(\mathbf{a}, \mathbf{h}|\mathbf{f}, \mathbf{e})$ which is the probability of an alignment with its types given the sentence pair. Applying the chain rule gives:

$$Pr(\mathbf{a}, \mathbf{h}|\mathbf{f}, \mathbf{e}) = Pr(\mathbf{a}|\mathbf{f}, \mathbf{e}) \times Pr(\mathbf{h}|\mathbf{a}, \mathbf{f}, \mathbf{e}) \quad (10)$$

where $Pr(\mathbf{a}|\mathbf{f}, \mathbf{e})$ is the posterior probability of IBM Model 1:

$$Pr(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \prod_{j=1}^{J} \frac{p(f_j|e_{a_j})}{\sum_{i=0}^{I} p(f_j|e_i)} \quad (11)$$

$Pr(\mathbf{h}|\mathbf{a}, \mathbf{f}, \mathbf{e})$ can be written as a product of alignment type parameters over the individual source and target words and their corresponding alignment type:

$$Pr(\mathbf{h}|\mathbf{a}, \mathbf{f}, \mathbf{e}) = \prod_{j=1}^{J} p(h_j|f_j, e_{a_j}) \quad (12)$$

Substituting Equations 11 and 12 in Equation 10 and simplifying it results in:

$$Pr(\mathbf{a}, \mathbf{h}|\mathbf{f}, \mathbf{e}) = \prod_{j=1}^{J} \frac{p(f_j|e_{a_j}) \times p(h_j|f_j, e_{a_j})}{\sum_{i=0}^{I} p(f_j|e_i)} \quad (13)$$

We collect the expected counts over all possible alignments and their alignment types, weighted by their probability. Suppose $c(f, h|e; \mathbf{f}, \mathbf{e})$ is the expected count for a word e generating a word f with an alignment type h in a sentence pair $(\mathbf{f}, \mathbf{e})$:

$$c(f, h|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a},\mathbf{h}} [Pr(\mathbf{a}, \mathbf{h}|\mathbf{f}, \mathbf{e}) \sum_{j=1}^{J} \delta(f, f_j)\delta(e, e_{a_j})\delta(h, h_j)] \quad (14)$$

where δ is the Kronecker delta function. Plugging $Pr(\mathbf{a}, \mathbf{h}|\mathbf{f}, \mathbf{e})$ (Equation 13) in Equation 14, yields

$$c(f, h|e; \mathbf{f}, \mathbf{e}) = \frac{p(f|e) \times p(h|f, e)}{\sum_{i=0}^{I} p(f|e_i)} \times \sum_{j=1}^{J} \delta(f, f_j) \sum_{i=0}^{I} \delta(e, e_i)\delta(h, h_j) \quad (15)$$

The alignment type parameters are then estimated by Equation 16.

$$p(h|f, e) = \frac{\sum_{(\mathbf{f},\mathbf{e})} c(f, h|e; \mathbf{f}, \mathbf{e})}{\sum_h \sum_{(\mathbf{f},\mathbf{e})} c(f, h|e; \mathbf{f}, \mathbf{e})} \quad (16)$$

Translation probabilities are estimated similar to IBM Model 1. This model is called IBM1+Type+Gen in the experiments section.

After training, we can jointly predict the best alignment and the best alignment types for each sentence pair:

$$\hat{\mathbf{a}}, \hat{\mathbf{h}} = \arg\max_{\mathbf{a},\mathbf{h}} \prod_{j=1}^{J} p(f_j|e_{a_j}) p(h_j|f_j, e_{a_j}) \quad (17)$$

In this decoding method, for a given sentence pair, for each source word f_j, we have to go through all the target words e_{a_j} in the target sentence and all the possible alignment types and find the pair of target position and alignment type that maximizes $p(f_j|e_{a_j})p(h_j|f_j, e_{a_j})$.

4.1.3 HMM with Alignment Types

Our HMM with alignment types model has the factor $p(f_j, h_j|e_{a_j})$ in its formulation which can be further decomposed to give:

$$Pr(\mathbf{f}, \mathbf{a}, \mathbf{h}|\mathbf{e}) = \prod_{j=1}^{J} p(a_j|a_{j-1}, I)p(f_j|e_{a_j})p(h_j|f_j, e_{a_j}) \quad (18)$$

This model is called HMM+Type+Gen in this paper. We now explain how we can estimate the parameters of this model. A compact representation of this model is $\theta = \{p(i|i', I), p(f|e), p(h_j|f, e)\}$ where $p(i|i', I)$ are the transition probabilities, $p(f_j|e_i)$ are the emission probabilities and $p(h_j|f_j, e_i)$ are the alignment type probabilities.

Let $\gamma_i(j, h) = Pr(a_j = i, h_j = h|\mathbf{f}, \theta)$ be the posterior probabilities for the HMM+Type+Gen model. Since $Pr(a_j = i, h_j = h|\mathbf{f}, \theta) = Pr(a_j = i|\mathbf{f}, \theta) \times Pr(h_j = h|a_j = i, \mathbf{f}, \theta)$, we have:

$$\gamma_i(j, h) = \gamma_i(j) \times p(h|f_j, e_i) \quad (19)$$

Equation 19 confirms that the posterior probability of the HMM+Type+Gen model is the HMM posterior multiplied by the alignment type probability factor $p(h|f_j, e_i)$ which is similar to the relationship between posterior probabilities in case of IBM Model 1 as shown in Equation 13.

Using this equation, we can compute the expected counts:

$$c(f, h|e; \mathbf{f}, \mathbf{e}) = \sum_{i,j} \gamma_i(j, h)\delta(f_j, f)\delta(e_i, e)\delta(h_j, h) \quad (20)$$

The expected counts are then normalized in the M-step to re-estimate the parameters. The transition and emission parameters are estimated as the standard HMM-based alignment model.

The EM algorithm for this model is similar to the Baum-Welch algorithm for the standard HMM-based word alignment model. The only change is that in the E-step, we need to collect the alignment type expected counts and in the M-step, alignment type parameters are re-estimated.

After training, Viterbi decoding is used to find the best word alignment and alignment types for new sentences. We define $V_i(j, h)$ to be the probability of the most probable alignment for $f_1 \dots f_j$ that f_j is aligned to e_i and the alignment type for this link is h. It can be computed recursively as follows:

$$V_i(j, h) = \max_{i',h'} \{V_{i'}(j-1, h')p(i|i', I)p(f_j|e_i)p(h|f_j, e_i)\} \quad (21)$$

4.2 Discriminative Models

Although we can use the generative models explained in Section 4.1 to estimate the alignment type probabilities $p(h|f, e)$, we can build a classifier to predict the alignment type given a pair of aligned words.

We have a set of 11 possible alignment types in the LDC data which are the possible classes in the classification problem. We use logistic regression to model the alignment type prediction problem. The rationale for using this model is that it can provide us with both the alignment type and the probability of being classified as this type.

4.2.1 Features

We used 22 different types of features in our logistic regression model as shown in Table 2. Lexical features are the heart of all lexical translation models; here, they are defined on pairs of Chinese and English words, shown by feature template (c_0, e_0) in Table 2.

Moreover, we include features taking the context into consideration. For example, (c_{-1}, c_0, e_0) uses the previous Chinese word apart from the pair of English and Chinese words. Part-of-speech (POS) tags are used to address the sparsity of the lexical features. For example, POS tags of the pair of Chinese and English words (c_{t_0}, e_{t_0}) are included. We also use the first five letters of the English word in a feature, to approximate the stem of an English word. An example is $(c_0, [e_0]_5)$ where the pair of Chinese word and the prefix of English word is used as a feature.

word-based	(c_0, e_0), (c_{-1}, c_0, e_0), $(c_{-2}, c_{-1}, c_0, e_0)$, (c_0, c_1, c_2, e_0), (c_0, e_{-1}, e_0), (c_0, e_{-1}, e_0, e_1), (c_0, e_0, e_1)
part-of-speech tag-based	(c_0, e_{t_0}, e_0), $(c_{-1}, c_0, e_{-1}, e_{t_0})$, $(c_0, e_{t_{-1}}, e_{t_0}, e_0)$ $(c_0, e_{t_{-1}}, e_0, e_{t_1})$, $(c_0, e_{t_{-1}}, e_{t_0}, e_{t_1})$, (c_{t_0}, e_{t_0}), $(c_{t_0}, c_{t_1}, c_{t_2})$, $(c_{t_0}, c_{t_{-1}}, e_{t_0})$, $(c_{t_{-1}}, c_{t_0}, c_{t_{-2}}, e_{t_0})$, $(c_{t_0}, c_{t_1}, e_{t_0})$, $(c_{t_{-1}}, c_{t_0}, c_{t_1}, e_{t_0})$
substring-based	$(c_0, [e_0]_5)$, $(c_0, [e_{-1}]_5, [e_0]_5, [e_1]_5)$, $(c_{t_0}, e_{-1}, [e_0]_5, e_{t_0})$, $(c_0, e_{t_0}, e_{t_{-1}}, [e_0]_5)$

Table 2: Feature types used in our alignment type classifier.

4.2.2 EM+Discriminative Algorithm for Alignment Types

In this section, we introduce the discriminative variants of the generative models explained in Section 4.1. These discriminative models are referred to as IBM1+Type+Disc and HMM+Type+Disc. The main difference between these models and their generative counterparts is in the way they compute alignment type probabilities $p(h|f, e)$. Whereas the generative models estimate these probabilities using the EM algorithm, the discriminative models estimate these probabilities using the logistic regression classifier. For the discriminative models, we first train a logistic regression model on the LDC data (see Section 5.1.2). The model provides us with the alignment type probabilities which are used in the decoding stage.

For IBM1+Type+Gen model, expected counts for alignment types are collected and alignment type parameters are updated in each iteration. In the EM algorithm for IBM1+Type+Disc, however, we do not need to collect the expected counts for alignment types since these parameter values are obtained from the pre-trained logistic regression classifier. However there is an important difference in the decoding step: Equation 17 is used to jointly find the best alignment and alignment types for each sentence pair. This joint decoding step makes this approach different from simply using a pipeline trained EM model followed by a discriminative classifier on the Viterbi output of the EM trained model. A comparison with the pipeline model is given in Section 5.2.

Similarly, the EM training of HMM+Type+Disc is similar to the EM training of baseline HMM. For decoding a new sentence pair, Equation 21 is used.

5 Experiments

For the experiments, we have used two datasets. The first is the GALE Chinese-English Word Alignment and Tagging corpus which is released by LDC[1]. This dataset is annotated with gold alignment and alignment types (see Section 2 for more details). The second dataset is the Hong Kong parliament proceedings (HK Hansards) for which we do not have the gold alignment and alignment types. We used 1 million sentences of the HK Hansards in the experiments to augment the training data. In the following sections, we describe three experiments. First, we examine how effective the logistic regression classifier is for alignment type prediction. Second, we present our experiments for two tasks: word alignment and the joint prediction of word alignment and alignment types. Finally, we explain the machine translation experiment.[2]

5.1 Alignment Type Prediction Given Alignments

For the alignment type prediction task given an aligned word pair, we have examined three simple maximum likelihood classifiers as well as the logistic regression classifier with the features shown in Table 2. We have trained all these classifiers on the parallel Chinese-English 20K LDC data which is annotated with gold alignment and alignment types. To obtain the word pairs, we have extracted the word pairs from the parallel sentences with the gold alignment. To get the part-of-speech tags, we annotated the 20K LDC data with the Stanford POS tagger

[1]Catalog numbers: LDC2012T16, LDC2012T20, LDC2012T24, LDC2013T05, LDC2013T23 and LDC2014T25.

[2]All our codes for the baselines and the proposed models are available at `https://github.com/sfu-natlang/align-type-tacl2017-code`.

(Toutanova et al., 2003). We ignored the gold alignment if the Chinese side of the gold alignment is not contiguous; i.e., it cannot form one Chinese word. This usually happens in the many-to-one and many-to-many alignments. There were only a small number of these discontiguous alignments as mentioned in the LDC catalog entry for this data.

5.1.1 Maximum Likelihood Classifiers

We have examined three maximum likelihood (ML) classifiers. The first model is a word-based ML classifier that uses the maximum likelihood estimate of the alignment type parameters $p(h|f,e)$, computed from the training data, to predict the alignment type for a new given pair of aligned words in a sentence pair in the test data. If the aligned words were not seen in the training data, this model backs-off to SEM as it is the most probable alignment type. The second model which is a tag-based ML classifier, uses the maximum likelihood estimate of $p(h|t_f,t_e)$ parameters of the model trained on the POS tagged data. t_f and t_e are the POS tags of the Chinese word f and the English word e, respectively. It backs-off to SEM for unseen pair of POS tags. Finally, for a pair of word (f,e), the last classifier first uses the ML estimate of $p(h|f,e)$ parameter. For unseen pair of words, it backs-off to use the ML estimate of $p(h|t_f,t_e)$ and in case the pair of POS tags was not seen, it backs-off to SEM.

5.1.2 Logistic Regression Classifier

We evaluated the logistic regression classifier which makes use of the features shown in Table 2 and the combination of different sets of these features. We assessed the performance of our features using 10-fold cross-validation. The best average cross-validation accuracy of 81.5% was achieved by a classifier that combines all the 22 features, shown in Table 2. We have used this trained classifier for the discriminative models (IBM1+Type+Disc and HMM+Type+Disc) in the experiments reported in Section 5.2.

5.1.3 Results

Table 3 shows the accuracy of the classifiers on the 2K held-out test data. The logistic regression classifier achieved the best accuracy on the test data. Since the logistic regression classifier obtains 87.5% on training, and the cross-validation accuracy variance was small we do not believe the classifier overfits on our training data.

Model	accuracy
ML word-based	73.8
ML tag-based	72.3
ML word-tag	79.1
Logistic regression	**81.4**

Table 3: Accuracy of the alignment type classifiers given the alignment.

5.2 Joint Word Alignment and Alignment Type Experiments

We measure the performance of our models using precision, recall and F1-score. Also, we evaluated the performance of our models and the baseline models on two different tasks: (1) The traditional word alignment task and (2) The joint prediction of word alignment and alignment types task. The second task is harder as the model has to predict both word alignment and alignment types correctly. Moreover, as the baseline IBM Model 1 and the baseline HMM cannot predict the alignment types, we can only make a comparison between our generative and discriminative models for the second task.

We initialized the translation probabilities of Model 1 uniformly over the word pairs that occur together in the same sentence pair. We built an HMM similar to the one proposed by Och and Ney (2003). This model is referred to as HMM in this paper. HMM was initialized with uniform transition probabilities and Model 1 translation probabilities. Model 1 was trained for 5 iterations; it is followed by 5 iterations of HMM.

To handle unseen data when the model is applied to the test data, smoothing has been used. We smooth translation probability $p(f|e)$ by backing-off to a uniform probability $1/|V|$ where $|V|$ is the source vocabulary size.

For smoothing alignment type probabilities $p(h|f,e)$, we used the following linear interpolation:

$$p^*(h|f,e) = \lambda_1 p(h|f,e) + \lambda_2 p(h|t_f,t_e) + \lambda_3 p(h) \quad (22)$$

where $\lambda_1 \geq 0, \lambda_2 \geq 0$ and $\lambda_3 \geq 0$ are the smoothing parameters and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

t_f and t_e are the POS tags of the Chinese word f and the English word e, respectively. To obtain $p(h|t_f, t_e)$, we labelled the parallel data with Stanford POS tagger and then trained a model on just these POS tags. $p(h)$ is the prior probability of alignment type h which can be estimated over the gold training data using Table 1. Both $p(h|f, e)$ and $p(h|t_f, t_e)$ are smoothed with $p(h)$ using linear interpolation.

To learn the hyper-parameters, we split the 20K LDC training data into two sets: a train set of 18K sentences and a 2K validation set. To learn p_0 and NULL emission probability, we performed a two-dimensional grid search varying p_0 in the set $\{0.05, 0.1, 0.2, 0.3, 0.4\}$ and NULL emission probability in the set {1e-7, 5e-7, ...,1e-2, 5e-2, 1e-1}. The tuned parameters that lead to the best result were achieved when $p_0 = 0.3$ and NULL emission probability was 5e-6. To tune the hyper-parameters λ_1, λ_2 and λ_3, we performed a two-dimensional grid search. The tuned parameters that lead to the best result was achieved when $\lambda_1 = 0.99$, and λ_3 =1e-15. Hence, $\lambda_2 = 1 - \lambda_1 - \lambda_3 = 9.99$e-11. We then used these learned parameters in the experiments.

Finally, for HMM-based models, we smooth transition parameters $p(i|i', I)$ by backing off to a uniform prior $1/I$.

5.2.1 Results on the LDC Alignment Type Data

Table 4 shows the models' performance for the word alignment task for all the baselines and the methods introduced in this paper. In this table, MODEL+Type+Gen denotes the proposed generative variant of MODEL while MODEL+Type+Disc denotes the proposed discriminative variant of MODEL. We trained all the models on the 20K data and tested on the 2K held-out data. We can see that our generative models consistently outperform their corresponding baselines. The best performing model, HMM+Type+Gen, achieves up to 13.9% improvement in F1-score over the baseline HMM. To compare our models against GIZA++, we add the test data to the training, and use Moses (Koehn et al., 2007) with its default parameters to obtain word alignments. We report its performance on the test data. Unlike the other models in Table 4 which are trained on 20K data, GIZA++ model is trained on 22K data.[3]

WA Results: Train(20K) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1	50.9	40.5	45.1
IBM1+Type+Disc	51.7	41.2	45.8
IBM1+Type+Gen	59.0	47.0	52.3
HMM	68.0	48.7	56.7
HMM+Type+Disc	66.2	50.5	57.3
HMM+Type+Gen	**72.9**	**58.0**	**64.6**
GIZA++	61.4	47.7	53.8

Table 4: Word alignment task results of the models trained on 20K LDC data (22K LDC data for GIZA++) and tested on 2K LDC test data.

Table 5 shows the results obtained for the joint prediction of word alignment and alignment types task. As mentioned previously, the basic IBM Model 1 and HMM are incapable of predicting the alignment types and hence are not included in this table. However, it is interesting to compute word alignments using our baselines and then apply the logistic regression classifier on the alignments to get the corresponding alignment types. In Table 5, MODEL→Disc denotes this pipelined version of MODEL. The only difference between MODEL+Type+Disc and MODEL→Disc is in the decoding step. The former jointly predicts word alignment and alignment types while the latter performs word alignment and then applies the classifier on the output of word alignment to obtain the alignment types. We also computed word alignments using GIZA++ as explained for the previous experiment and then ran our logistic regression classifier on the alignments to get the corresponding alignment types. This model is denoted as GIZA++→Disc in Table 5. The results in this table show that the generative model outperforms its discriminative counterpart. Similar to the previous experiment, HMM+Type+Gen model achieved the

[3] GIZA++ does not allow the user to run it as a classifier (a model that is trained on the training data and can be tested on new data). Initially, we performed incremental training with inc-giza-pp (Levenberg et al., 2010). Since the performance was very poor, we used GIZA++ in our experiments by appending the test data to the training data (even though our models did not see the test data) and reported the result of Viterbi output from the trained GIZA++ model on the combined data.

best result.

WA+Type Results: Train(20K) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1+Type+Disc	44.0	37.5	40.5
IBM1+Type+Gen	47.8	40.8	44.0
HMM+Type+Disc	55.3	45.2	49.8
HMM+Type+Gen	**59.2**	**50.5**	**54.5**
IBM1→Disc	42.9	36.6	39.5
HMM→Disc	57.2	43.8	49.6
GIZA++→Disc	52.2	43.5	47.5

Table 5: Results of the models trained on 20K LDC data (22K LDC data for GIZA++) and tested on 2K LDC test data for (1) joint prediction of word alignment and alignment types task, and (2) word alignment models followed by the discriminative classifier.

5.2.2 Results with Augmented Model

We conducted another experiment to see whether we can improve the current results by augmenting the training data. We trained on the 20K LDC data with gold alignment and alignment types, and 1 million Hong Kong Hansards which has no alignment or alignment types annotations and tested on the 2K held-out data. Although Hong Kong Hansards data is not annotated, it can augment our vocabulary. We built a model with the 20K LDC data, called LDC model. We then trained a model using the 20K LDC data and the 1 million HK Hansards data, called the augmented model. The alignment type parameters of the augmented model are initialized based on the maximum likelihood estimate of the 20K LDC data. Table 6 and 7 show the results of the augmented model for the word alignment task and the joint prediction of word alignment and alignment types tasks respectively.

5.2.3 Results with Augmented Model and Back-off Smoothing

Using the augmented model purely was not effective in estimating the translation probabilities $p(f|e)$, and hence did not contribute to any improvement compared to the previous experiment. This is due to the fact that HK Hansards data is from a different domain compared to our test LDC data. Since the 2K test data is from the LDC data, we applied a back-off smoothing technique: we estimated $p(f|e)$ from the LDC model if the word pair (f, e) was seen by the LDC model, and we used the augmented model to compute $p(f|e)$ otherwise.

WA Results: Train(20K+1M) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1	49.7	39.6	44.1
IBM1+Type+Disc	50.5	40.2	44.8
IBM1+Type+Gen	59.5	47.4	52.8
HMM	67.7	48.8	56.7
HMM+Type+Disc	66.1	50.7	57.4
HMM+Type+Gen	**73.1**	**58.2**	**64.8**
GIZA++	60.0	47.0	52.7

Table 6: Word alignment task results for the augmented model.

WA+Type Results: Train(20K+1M) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1+Type+Disc	42.9	36.6	39.5
IBM1+Type+Gen	48.0	40.9	44.2
HMM+Type+Disc	55.2	45.4	49.8
HMM+Type+Gen	**59.2**	**50.4**	**54.5**
IBM1→Disc	41.9	35.7	38.6
HMM→Disc	56.7	43.7	49.4
GIZA++→Disc	51.0	42.8	46.5

Table 7: Results using the augmented model for (1) joint prediction of word alignment and alignment types task, and (2) word alignment models followed by the discriminative classifier.

Table 8 shows the results of the augmented model after the smoothing step is done for the word alignment task. Compared to the results in Table 4, all the models performed better, with the HMM+Type+Gen outperforming all the other methods.

The results for the joint prediction task are shown in Table 9. This confirms our success in improving the performance of all the methods, compared to the results in Table 5.

Statistical significance tests were performed using the approximate randomization test (Yeh, 2000) with 10000 iterations. The generative models significantly outperform their baseline and discriminative

WA Results: Train(20K+1M) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1	52.7	42.0	46.7
IBM1+Type+Disc	53.5	42.6	47.4
IBM1+Type+Gen	60.3	48.0	53.5
HMM	69.4	50.0	58.1
HMM+Type+Disc	67.1	51.9	58.5
HMM+Type+Gen	**74.5**	**59.2**	**66.0**
GIZA++	60.0	47.0	52.7

Table 8: Word alignment task results, back-off using the augmented model.

WA+Type Results: Train(20K+1M) + Test(2K)			
Model	Prec.	Rec.	F1-score
IBM1+Type+Disc	45.3	38.6	41.7
IBM1+Type+Gen	48.6	41.5	44.8
HMM+Type+Disc	55.9	46.2	50.6
HMM+Type+Gen	**60.3**	**51.3**	**55.4**
IBM1→Disc	44.3	37.8	40.8
HMM→Disc	58.2	44.9	50.7
GIZA++→Disc	51.0	42.8	46.5

Table 9: Results with back-off using the augmented model for (1) joint prediction of word alignment and alignment types task, and (2) word alignment models followed by the discriminative classifier.

counterparts (p-value < 0.0001).

5.3 Machine Translation Experiment

To see whether the improvement in F1-score by our generative model also improves BLEU score, we aligned the 20K LDC data and 1 million sentences of the HK hansards data using the augmented model and tested on 919 sentences of MTC part 4 (LDC2006T04). We trained models in each translation direction and then symmetrized the produced alignments using the grow-diag-final heuristic (Och and Ney, 2003). We used Moses (Koehn et al., 2007) with standard features, and tuned the weights with MERT (Och, 2003). An English 5-gram language model is trained using KenLM (Heafield, 2011) on the Gigaword corpus (Parker et al., 2011). We give a comparison between HMM+Type+Gen model, our baseline HMM, GIZA++ HMM and standard GIZA++ (as used by Moses) in Table 10. We report the BLEU scores and TER computed using MultEval (Clark et al., 2011).

Model	BLEU	TER
GIZA++ HMM	23.4	70.4
GIZA++ (Moses)	23.2	69.1
HMM	23.5	68.3
HMM+Type+Gen	**24.4**	**67.8**

Table 10: Comparison of the BLEU and TER scores. GIZA++ (Moses) is the version used in the Moses MT system.

The generative model improves over GIZA++ HMM by 1.0 BLEU points. It also improves over the standard GIZA++ by 1.2 BLEU points. HMM+Type+Gen significantly outperforms GIZA++ HMM (p-value=0.00036) and GIZA++ IBM4 (p-value=0.0004) evaluated by MultEval.

6 Discussion

Figure 2 shows the performance of baseline HMM and HMM+Type+Gen model for two word alignment examples extracted from the test data, where squares indicate the gold standard alignments. Numbers in the circles show the ids of the predicted tags by the HMM+Type+Gen model, where id of each tag is defined in Table 1. The incorrectly predicted tags are shown with the $*$ symbol.

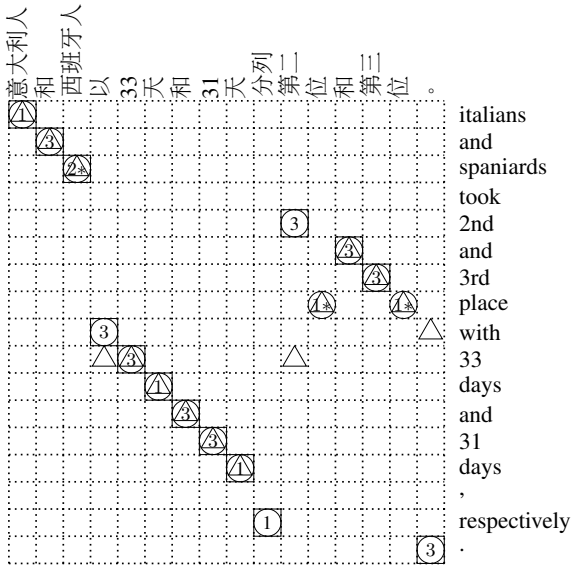
In both examples, HMM+Type+Gen model identifies difficult alignments over long distances compared to the baseline HMM. For example, Figure 2(a) illustrates how our baseline HMM makes a mistake by aligning the Chinese word "。" to "with" possibly because the transition probabilities were dominant in the baseline HMM. HMM+Type+Gen model however avoids this mistake by making use of the alignment type information. The model takes into account the fact that "。" and "." are function words and should be aligned to each other with a FUN tag. Figure 2(b) shows that HMM+Type+Gen model favors aligning 见面 (meet) to "meet", whereas baseline HMM incorrectly aligns 见面 (meet) to "jintao". We hypothesize that this occurs because p(SEM| 见面, meet) has a high value.

To give a detailed analysis of the precision of the generative model in alignment type prediction, we present a confusion matrix on the test data in Table 11 where the vertical axis represents the ac-

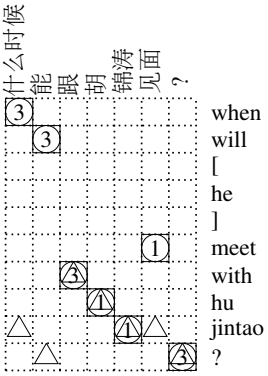
tual alignment type and the horizontal axis represents the predicted alignment type. From the confusion matrix, we found that our model works well in predicting SEM, FUN, GIS, GIF, MDE and CDE alignment types since the numbers on the diagonal are the largest in the row. PDE is hard to be distinguished from MDE. COI and TIN can be easily mis-predicted by the model. NTR and MTA are omitted from this table as all the predictions for these alignment types are zero. For MTA, it is probably because this type occurs in our training data rarely. An alignment type is NTR if either Chinese or English list of tokens for that alignment is empty. In other words, NTR alignment type is used when some words are dropped during the translation process. We could predict NTR for the Chinese words that are aligned to NULLs. However, predicting NTR for such cases worsened the F-score of the generative model (2.0 points drop for HMM+Type+Gen model). Hence, we do not predict NTR alignment type for any Chinese words. In total, just for the confusion matrix, 10216 alignments (or 25.54% of all alignments) are not included in Table 11 which shows the alignment type predictions for the word pairs that were correctly aligned by HMM+Type+Gen[4]

	SEM	FUN	GIS	GIF	MDE	PDE	CDE	COI	TIN
SEM	11374	136	2002	10	0	0	0	14	3
FUN	196	8172	21	16	2	0	0	1	1
GIS	2790	31	3312	18	2	1	2	8	0
GIF	16	118	26	772	0	0	0	0	2
MDE	0	0	1	0	293	26	2	0	0
PDE	1	0	1	2	40	55	0	0	0
CDE	0	5	0	0	0	0	79	0	0
COI	91	2	48	0	0	0	0	38	0
TIN	22	12	11	3	0	0	0	0	5

Table 11: Confusion matrix on the LDC test data. The vertical axis represents the actual alignment type and the horizontal axis represents the predicted alignment type.



(a)



(b)

Figure 2: A comparison between the performance of baseline HMM and HMM+Type+Gen model for two test sentences; ○: HMM+Type+Gen, △: HMM and □: gold alignment. Numbers in the circles show the ids of the predicted alignment types by the HMM+Type+Gen model, where ids are given in Table 1. The incorrectly predicted alignment types are shown with the $*$ symbol.

7 Related Work

There has been several studies on semi-supervised word alignment models. Callison-Burch et al. (2004) improve alignment and translation quality by interpolating hand-annotated word-aligned data and automatic sentence-aligned data. They showed that a much higher weight should be assigned to the model trained on word-aligned data. Fraser and Marcu (2006) propose a semi-supervised training approach to word alignment, based on IBM Model 4, that alternates the EM step which is applied on a large training corpus with a discriminative er-

[4]We should note that these incorrectly predicted alignments are only kept out of the confusion matrix. All alignments, correct or incorrect, are included in all the results we show in the other tables.

ror training step on a small hand-annotated sub-corpus. The alignment problem is viewed as a search problem over a log-linear space with features (sub-models) coming from the IBM Model 4. In the proposed algorithm, discriminative training controls the contribution of sub-models while an EM-like procedure is used to estimate the sub-model parameters. Unlike previous approaches (Och and Ney, 2003; Fraser and Marcu, 2006; Fraser and Marcu, 2007) that use discriminative methods to tune the weights of generative models, Gao et al. (2010b) propose a semi-supervised word alignment technique that integrate discriminative and generative methods. They propose to use a discriminative word aligner to produce high precision partial alignments that can be served as constraints for the EM algorithm. On the other hand the discriminative word aligner uses the generative aligner's output as features. This feedback loop iteratively improves the quality of both aligners. Niehues and Vogel (2008) propose a discriminative model that directly models the alignment matrix. Although the discriminative model provides the flexibility to use manually word-aligned data to tune its weights, it still relies on the model parameters of IBM models and alignment links from GIZA++ as features. Gao et al. (2010a) present a semi-supervised algorithm that extends IBM Model 4 by using partial manual alignments. Partial alignments are fixed and treated as constraints into the EM training.

DeNero and Klein (2010) present a supervised model for extracting phrase pairs under a discriminative model by using word alignments. They consider two types of alignment links, *sure* and *possible*, that are extracted from the manually word-aligned data. *Possible* alignment links dictate which phrase pairs can be extracted from a sentence pair.

Among the unsupervised methods, (Toutanova et al., 2002) utilises additional source of information apart from the parallel sentences. Part-of-speech tags of the words in the sentence pair are incorporated as a linguistic constraint on HMM-based word alignment. The part-of-speech tag translation probabilities in this model are then learned along with other probabilities using the EM algorithm. POS tags as used in (Toutanova et al., 2002) were also utilised to act similarly to word classes in (Och and Ney, 2000a; Och and Ney, 2000b); however, the improvements provided by the HMM with POS tag model over HMM alignment model of (Och and Ney, 2000b) was for small training data sizes (<50K parallel corpus).

All previous studies on word alignment have assumed that word alignments are untyped. To our knowledge, the alignment types for word alignment provided by the LDC as annotations on word alignment links, have never been used to improve word alignment. Our work differs from the previous works as it proposes a new task of jointly predicting word alignment and alignment types. A semi-supervised learning algorithm is presented to solve this task. Our method is semi-supervised as it combines LDC data, which is annotated with alignment and alignment types, with sentence aligned (but not word aligned) data from the HK Hansards corpus. Our generative algorithm makes use of the gold alignment and alignment types data to initialize the alignment type parameters and then the EM training is used to re-estimate the parameters of the model in an unsupervised manner. In the generative model, POS tags were only used to smooth the alignment type parameters, unlike the approach in (Toutanova et al., 2002).

8 Conclusion

We incorporate alignment types into standard word alignment models. We train on the GALE Chinese-English word alignment and tagging corpus which enriches word alignment with 11 linguistic alignment types. We proposed a new task of jointly predicting word alignment and alignment types. The proposed generative HMM with alignment types achieves up to 13.9% improvement in F1-score over the baseline HMM. This model improved the BLEU score by 1.0 points over the GIZA++ HMM. It also improved the BLEU score by 1.2 points over the standard GIZA++ aligner. In the future, we plan to use alignment type information as a feature function for feature rich word alignment models. We also plan to explore how alignments types can improve attention models for neural MT models. The alignment types we predict for each aligned word pair can be used for other NLP tasks such as projection of part-of-speech tags and dependency trees from a resource-rich language to a resource-poor language.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590. Association for Computational Linguistics.

Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72. Association for Computational Linguistics.

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

Chris Callison-Burch, David Talbot, and Miles Osborne. 2004. Statistical machine translation with word-and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 175. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*.

John DeNero and Dan Klein. 2010. Discriminative modeling of extraction sets for machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1453–1463. Association for Computational Linguistics.

Yonggang Deng and Yuqing Gao. 2007. Guiding statistical word alignment models with prior knowledge. In *Annual Meeting of the Association for Computational Linguistics*, volume 45, page 1.

Chris Dyer, Jonathan Clark, Alon Lavie, and Noah A Smith. 2011. Unsupervised word alignment with arbitrary features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 409–419. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471. The Association for Computer Linguistics.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 769–776. Association for Computational Linguistics.

Alexander Fraser and Daniel Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 51–60, Prague, Czech Republic, June. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Qin Gao, Nguyen Bach, and Stephan Vogel. 2010a. A semi-supervised word alignment algorithm with partial manual alignments. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 1–10. Association for Computational Linguistics.

Qin Gao, Francisco Guzman, and Stephan Vogel. 2010b. Emdc: a semi-supervised approach for word alignment. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 349–357. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 89–96. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 394–402. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Deep multilingual correlation for improved word embeddings. In *Proceedings of NAACL*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the conference on empirical methods in natural language processing*, pages 62–72. Association for Computational Linguistics.

Robert C Moore, Wen-tau Yih, and Andreas Bode. 2006. Improved discriminative bilingual word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 513–520. Association for Computational Linguistics.

Jan Niehues and Stephan Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000a. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1086–1090. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000b. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. 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.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. Technical report, Technical Report. Linguistic Data Consortium, Philadelphia.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80. Association for Computational Linguistics.

Kristina Toutanova, H Tolga Ilhan, and Christopher D Manning. 2002. Extensions to hmm-based statistical word alignment models. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 87–94. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Xiaolin Wang, Masao Utiyama, Andrew M Finch, and Eiichiro Sumita. 2014. Refining word segmentation using a manually aligned corpus for statistical machine translation. In *EMNLP*, pages 1654–1664.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 947–953. Association for Computational Linguistics.