Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Contrastive Unsupervised Word Alignment with Non-Local Features

Yang Liu and **Maosong Sun**
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu 221009, China
{liuyang2011,sms}@tsinghua.edu.cn

Abstract

Word alignment is an important natural language processing task that indicates the correspondence between natural languages. Recently, unsupervised learning of log-linear models for word alignment has received considerable attention as it combines the merits of generative and discriminative approaches. However, a major challenge still remains: it is intractable to calculate the expectations of non-local features that are critical for capturing the divergence between natural languages. We propose a contrastive approach that aims to differentiate observed training examples from noises. It not only introduces prior knowledge to guide unsupervised learning but also cancels out partition functions. Based on the observation that the probability mass of log-linear models for word alignment is usually highly concentrated, we propose to use top-n alignments to approximate the expectations with respect to posterior distributions. This allows for efficient and accurate calculation of expectations of non-local features. Experiments show that our approach achieves significant improvements over state-of-the-art unsupervised word alignment methods.

Introduction

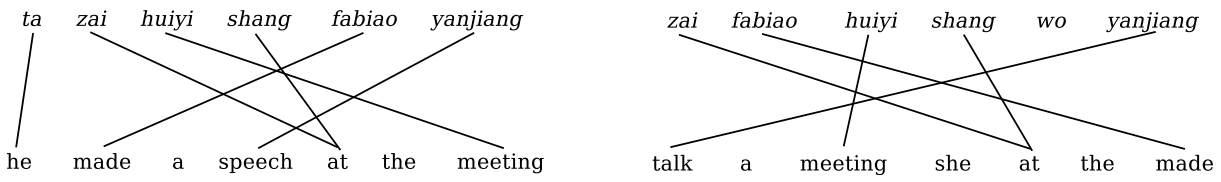
Word alignment is a natural language processing (NLP) task that aims to identify the correspondence between words in natural languages (Brown et al. 1993). Word-aligned parallel corpora are an indispensable resource for many NLP tasks such as machine translation and cross-lingual IR.

Current word alignment approaches can be roughly divided into two categories: *generative* and *discriminative*. Generative approaches are often based on generative models (Brown et al. 1993; Vogel, Ney, and Tillmann 1996; Liang, Taskar, and Klein 2006), the parameters of which are learned by maximizing the likelihood of unlabeled data. One major drawback of these approaches is that they are hard to extend due to the strong dependencies between submodels. On the other hand, discriminative approaches overcome this problem by leveraging log-linear models (Liu, Liu, and Lin 2005; Blunsom and Cohn 2006) and linear models (Taskar, Lacoste-Julien, and Klein 2005; Moore, Yih, and Bode 2006; Liu, Liu, and Lin 2010) to include arbitrary features. However, labeled data is expensive to build and hence is unavailable for most language pairs and domains.

As generative and discriminative approaches seem to be complementary, a number of authors have tried to combine the advantages of both in recent years (Berg-Kirkpatrick et al. 2010; Dyer et al. 2011; Dyer, Chahuneau, and Smith 2013). They propose to train log-linear models for word alignment on unlabeled data, which involves calculating two expectations of features: one ranging over all possible alignments given observed sentence pairs and another over all possible sentence pairs and alignments. Due to the complexity and diversity of natural languages, it is intractable to calculate the two expectations. As a result, existing approaches have to either restrict log-linear models to be locally normalized (Berg-Kirkpatrick et al. 2010) or only use local features to admit efficient dynamic programming algorithms on compact representations (Dyer et al. 2011). Although it is possible to use MCMC methods to draw samples from alignment distributions (DeNero, Bouchard-Côté, and Klein 2008) to calculate expectations of non-local features, it is computationally expensive to reach the equilibrium distribution. Therefore, including non-local features, which are critical for capturing the divergence between natural languages, still remains a major challenge in unsupervised learning of log-linear models for word alignment.

In the paper, we present a contrastive learning approach to training log-linear models for word alignment on unlabeled data. Instead of maximizing the likelihood of log-linear models on the observed data, our approach follows contrastive estimation methods (Smith and Eisner 2005; Gutmann and Hyvärinen 2012) to guide the model to assign higher probabilities to observed data than to *noisy* data. To calculate the expectations of non-local features, we propose an approximation method called top-n sampling based on the observation that the probability mass of log-linear models for word alignment is highly concentrated. Hence, our approach has the following advantages over previous work:

1. *Partition functions canceled out*. As learning only involves observed and noisy training examples, our training objective cancels out partition functions that comprise exponentially many sentence pairs and alignments.
2. *Efficient sampling*. We use a dynamic programming algorithm to extract top-n alignments, which serve as samples



(a) (b)

Figure 1: (a) An observed (romanized) Chinese sentence, an English sentence, and the word alignment between them; (b) a noisy training example derived from (a) by randomly permutating and substituting words. As the training data only consists of sentence pairs, word alignment serves as a latent variable in the log-linear model. In our approach, the latent-variable log-linear model is expected to assign higher probabilities to observed training examples than to noisy examples.

to compute the approximate expectations.

3. *Arbitrary features*. The expectations of both local and non-local features can be calculated using top-n approximation accurately and efficiently.

Experiments on multilingual datasets show that our approach achieves significant improvements over state-of-the-art unsupervised alignment systems.

Latent-Variable Log-Linear Models for Unsupervised Word Alignment

Figure 1(a) shows a (romanized) Chinese sentence, an English sentence, and the word alignment between them. The links indicate the correspondence between Chinese and English words. Word alignment is a challenging task because both the lexical choices and word orders in two languages are significantly different. For example, while the English word "at" corresponds to a discontinuous Chinese phrase "*zai ... shang*", the English function word "the" has no counterparts in Chinese. In addition, a verb phrase (e.g., "made a speech") is usually followed by a prepositional phrase (e.g., "at the meeting") in English but the order is reversed in Chinese. Therefore, it is important to design features to capture various characteristics of word alignment.

To allow for unsupervised word alignment with arbitrary features, latent-variable log-linear models have been studied in recent years (Berg-Kirkpatrick et al. 2010; Dyer et al. 2011; Dyer, Chahuneau, and Smith 2013). Let $\mathbf{x}$ be a pair of source and target sentences and $\mathbf{y}$ be the word alignment. A latent-variable log-linear model parametrized by a real-valued vector $\boldsymbol{\theta} \in \mathbb{R}^{K\times 1}$ is given by

$$P(\mathbf{x};\boldsymbol{\theta}) = \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} P(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) \quad (1)$$

$$= \frac{\sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}))}{Z(\boldsymbol{\theta})} \quad (2)$$

where $\boldsymbol{\phi}(\cdot) \in \mathbb{R}^{K\times 1}$ is a feature vector and $Z(\boldsymbol{\theta})$ is a partition function for normalization:

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y})) \quad (3)$$

We use $\mathcal{X}$ to denote all possible pairs of source and target strings and $\mathcal{Y}(\mathbf{x})$ to denote the set of all possible alignments for a sentence pair $\mathbf{x}$. Let l and m be the lengths of the source and target sentences in $\mathbf{x}$, respectively. Then, the number of possible alignments for $\mathbf{x}$ is $|\mathcal{Y}(\mathbf{x})| = 2^{l\times m}$. In this work, we use 5 *local* features (translation probability product, relative position absolute difference, link count, monotone and swapping neighbor counts) and 11 *non-local* features (cross count, source and target linked word counts, source and target sibling distances, source and target maximal fertilities, multiple link types) that prove to be effective in modeling regularities in word alignment (Taskar, Lacoste-Julien, and Klein 2005; Moore, Yih, and Bode 2006; Liu, Liu, and Lin 2010).

Given a set of training examples $\{\mathbf{x}^{(i)}\}_{i=1}^{I}$, the standard training objective is to find the parameter that maximizes the log-likelihood of the training set:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left\{ L(\boldsymbol{\theta}) \right\} \quad (4)$$

$$= \arg\max_{\boldsymbol{\theta}} \left\{ \log \prod_{i=1}^{I} P(\mathbf{x}^{(i)};\boldsymbol{\theta}) \right\} \quad (5)$$

$$= \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{I} \log \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}^{(i)})} \exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x}^{(i)},\mathbf{y})) - \log Z(\boldsymbol{\theta}) \right\} \quad (6)$$

Standard numerical optimization methods such as L-BFGS and Stochastic Gradient Descent (SGD) require to calculate the partial derivative of the log-likelihood $L(\boldsymbol{\theta})$ with respect to the k-th feature weight $\boldsymbol{\theta}_k$

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \sum_{i=1}^{I} \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x}^{(i)})} P(\mathbf{y}|\mathbf{x}^{(i)};\boldsymbol{\theta})\boldsymbol{\phi}_k(\mathbf{x}^{(i)},\mathbf{y}) - \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} P(\mathbf{x},\mathbf{y};\boldsymbol{\theta})\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y}) \quad (7)$$

$$= \sum_{i=1}^{I} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(i)};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x}^{(i)},\mathbf{y})] - \mathbb{E}_{\mathbf{x},\mathbf{y};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y})] \quad (8)$$

As there are exponentially many sentences and alignments, the two expectations in Eq. (8) are intractable to calculate for non-local features that are critical for measuring the fertility and non-monotonicity of alignment (Liu, Liu, and Lin 2010). Consequently, existing approaches have to use only local features to allow dynamic programming algorithms to calculate expectations efficiently on lattices (Dyer et al. 2011). Therefore, how to calculate the expectations of non-local features accurately and efficiently remains a major challenge for unsupervised word alignment.

Contrastive Learning with Top-n Sampling

Instead of maximizing the log-likelihood of the observed training data, we propose a contrastive approach to unsupervised learning of log-linear models. For example, given an observed training example as shown in Figure 1(a), it is possible to generate a *noisy* example as shown in Figure 1(b) by randomly shuffling and substituting words on both sides. Intuitively, we expect that the probability of the observed example is higher than that of the noisy example. This is called *contrastive learning*, which has been advocated by a number of authors (see Related Work).

More formally, let $\tilde{\mathbf{x}}$ be a noisy training example derived from an observed example $\mathbf{x}$. Our training data is composed of pairs of observed and noisy examples: $D = \{\langle \mathbf{x}^{(i)}, \tilde{\mathbf{x}}^{(i)} \rangle\}_{i=1}^{I}$. The training objective is to maximize the difference of probabilities between observed and noisy training examples:

$$\begin{aligned}
&\boldsymbol{\theta}^* \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ J(\boldsymbol{\theta}) \right\} && (9) \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ \log \prod_{i=1}^{I} \frac{P(\mathbf{x}^{(i)})}{P(\tilde{\mathbf{x}}^{(i)})} \right\} && (10) \\
&= \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{I} \log \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(i)})} \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\mathbf{x}^{(i)}, \mathbf{y})) \right. \\
&\qquad \left. - \log \sum_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}}^{(i)})} \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\tilde{\mathbf{x}}^{(i)}, \mathbf{y})) \right\} && (11)
\end{aligned}$$

Accordingly, the partial derivative of $J(\boldsymbol{\theta})$ with respect to the k-th feature weight $\boldsymbol{\theta}_k$ is given by

$$\begin{aligned}
&\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} \\
&= \sum_{i=1}^{I} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(i)})} P(\mathbf{y}|\mathbf{x}^{(i)}; \boldsymbol{\theta}) \boldsymbol{\phi}_k(\mathbf{x}^{(i)}, \mathbf{y}) \\
&\qquad - \sum_{\mathbf{y} \in \mathcal{Y}(\tilde{\mathbf{x}}^{(i)})} P(\mathbf{y}|\tilde{\mathbf{x}}^{(i)}; \boldsymbol{\theta}) \boldsymbol{\phi}_k(\tilde{\mathbf{x}}^{(i)}, \mathbf{y}) && (12) \\
&= \sum_{i=1}^{I} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(i)};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x}^{(i)}, \mathbf{y})] - \mathbb{E}_{\mathbf{y}|\tilde{\mathbf{x}}^{(i)};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\tilde{\mathbf{x}}^{(i)}, \mathbf{y})] && (13)
\end{aligned}$$

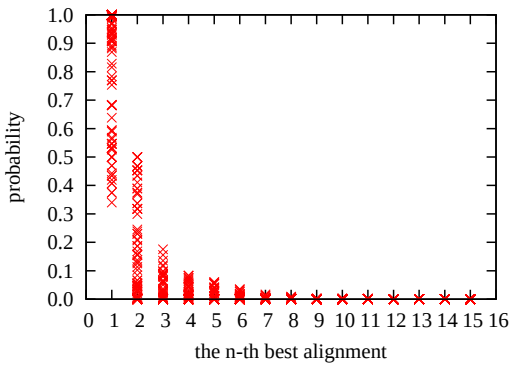


Figure 2: Distributions of log-linear models for alignment on short sentences (≤ 4 words).

The key difference is that our approach cancels out the partition function $Z(\boldsymbol{\theta})$, which poses the major computational challenge in unsupervised learning of log-linear models. However, it is still intractable to calculate the expectation with respect to the posterior distribution $\mathbb{E}_{\mathbf{y}|\mathbf{x};\boldsymbol{\theta}}[\boldsymbol{\phi}(\mathbf{x}, \mathbf{y})]$ for non-local features due to the exponential search space (i.e., $|\mathcal{Y}(\mathbf{x})| = 2^{l \times m}$). One possible solution is to use Gibbs sampling to draw samples from the posterior distribution $P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$ (DeNero, Bouchard-Côté, and Klein 2008). But the Gibbs sampler usually runs for a long time to converge to the equilibrium distribution.

Fortunately, by definition, only alignments with highest probabilities play a central role in calculating expectations. If the probability mass of the log-linear model for word alignment is concentrated on a small number of alignments, it will be efficient and accurate to only use most likely alignments to approximate the expectation.

Figure 2 plots the distributions of log-linear models parametrized by 1,000 random feature weight vectors. We used all the 16 features. The true distributions were calculated by enumerating all possible alignments for short Chinese and English sentences (≤ 4 words). We find that top-5 alignments usually account for over 99% of the probability mass.

More importantly, we also tried various sentence lengths, language pairs, and feature groups and found this concentration property to hold consistently. One possible reason is that the exponential function enlarges the differences between variables dramatically (i.e., $a > b \Rightarrow \exp(a) \gg \exp(b)$).

Therefore, we propose to approximate the expectation using most likely alignments:

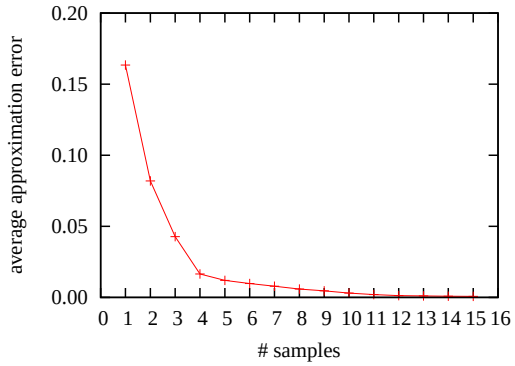


Figure 3: Average approximation errors of top-n sampling on short sentences ($\leq$ 4 words). The true expectations of local and non-local features are exactly calculated by full enumeration.

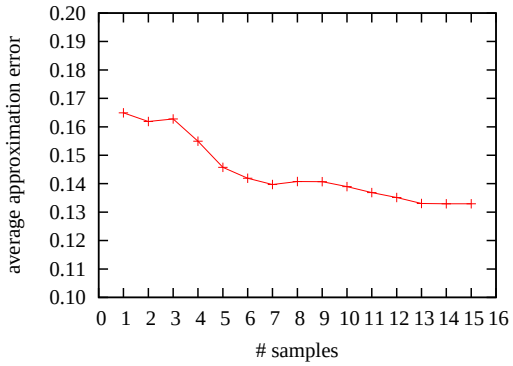


Figure 4: Average approximation errors of top-n sampling on long sentences ($\leq$ 100 words). The true expectations of local features are exactly calculated by dynamic programming on lattices.

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{y}|\mathbf{x};\boldsymbol{\theta}}[\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y})] \\
= & \sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y}) && (14) \\
= & \frac{\sum_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}))\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}'\in\mathcal{Y}(\mathbf{x})}\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}'))} && (15) \\
\approx & \frac{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x};\boldsymbol{\theta})}\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}))\boldsymbol{\phi}_k(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}'\in\mathcal{N}(\mathbf{x};\boldsymbol{\theta})}\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}'))} && (16)
\end{aligned}
$$

where $\mathcal{N}(\mathbf{x};\boldsymbol{\theta}) \subseteq \mathcal{Y}(\mathbf{x})$ contains the most likely alignments depending on $\boldsymbol{\theta}$:

$$
\begin{aligned}
& \forall \mathbf{y}_1 \in \mathcal{N}(\mathbf{x};\boldsymbol{\theta}), \forall \mathbf{y}_2 \in \mathcal{Y}(\mathbf{x}) \backslash \mathcal{N}(\mathbf{x};\boldsymbol{\theta}) : \\
& \boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}_1) > \boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}_2) && (17)
\end{aligned}
$$

Let the cardinality of $\mathcal{N}(\mathbf{x};\boldsymbol{\theta})$ be n. We refer to Eq. (16) as top-n sampling because the approximate posterior distribution is normalized over top-n alignments:

$$
P_{\mathcal{N}}(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}))}{\sum_{\mathbf{y}'\in\mathcal{N}(\mathbf{x})}\exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\mathbf{x},\mathbf{y}'))} \quad (18)
$$

In this paper, we use the beam search algorithm proposed by Liu, Liu, and Lin (2010) to retrieve top-n alignments from the full search space. Starting with an empty alignment, the algorithm keeps adding links until the alignment score will not increase. During the process, local and non-local feature values can be calculated in an incremental way efficiently. The algorithm generally runs in $O(bl^2m^2)$ time, where b is the beam size. As it is intractable to calculate the objective function in Eq. (11), we use the stochastic gradient descent algorithm (SGD) for parameter optimization, which requires to calculate partial derivatives with respect to feature weights on single training examples.

Experiments

Approximation Evaluation

To measure how well top-n sampling approximates the true expectations, we define the *approximation error* $E(D,\boldsymbol{\theta})$ as

$$
\frac{1}{I\times K}\sum_{i=1}^{I}||\boldsymbol{\delta}_{\mathcal{Y}}(\mathbf{x}^{(i)},\tilde{\mathbf{x}}^{(i)},\boldsymbol{\theta}) - \boldsymbol{\delta}_{\mathcal{N}}(\mathbf{x}^{(i)},\tilde{\mathbf{x}}^{(i)},\boldsymbol{\theta})||_1 \quad (19)
$$

where $\boldsymbol{\delta}_{\mathcal{Y}}(\cdot)$ returns the true difference between the expectations of observed and noisy examples:

$$
\boldsymbol{\delta}_{\mathcal{Y}}(\mathbf{x},\tilde{\mathbf{x}},\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{y}|\mathbf{x};\boldsymbol{\theta}}[\boldsymbol{\phi}(\mathbf{x},\mathbf{y})] - \mathbb{E}_{\mathbf{y}|\tilde{\mathbf{x}};\boldsymbol{\theta}}[\boldsymbol{\phi}(\tilde{\mathbf{x}},\mathbf{y})] \quad (20)
$$

Similarly, $\boldsymbol{\delta}_{\mathcal{N}}(\cdot)$ returns the approximate difference. $||\cdot||_1$ is the L_1 norm.

In addition, we define *average approximation error* on a set of random feature weight vectors $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{T}$:

$$
\frac{1}{T}\sum_{t=1}^{T}E(D,\boldsymbol{\theta}^{(t)}) \quad (21)
$$

Figure 3 shows the average approximation errors of our top-n sampling method on short sentences (up to 4 words) with 1,000 random feature weight vectors. To calculate the true expectations of both local and non-local features, we need to enumerate all alignments in an exponential space. We randomly selected $1,000$ short Chinese-English sentence pairs. One *noisy* example was generated for each observed example by randomly shuffling, replacing, inserting, and deleting words. We used the beam search algorithm (Liu, Liu, and Lin 2010) to retrieve n-best lists. We plotted the approximation errors for n up to 15. We find that the average approximation errors drop dramatically when n ranges from 1 to 5 and approach zero for large values of n, suggesting that a small value of n might suffice to approximate the expectations.

Figure 4 shows the average approximation errors of top-n sampling on long sentences (up to 100 words) with 1,000

# samples	Gibbs	Top-n
1	1.5411	0.1653
5	0.7410	0.1477
10	0.6550	0.1396
50	0.5498	0.1108
100	0.5396	0.1086
500	0.5180	0.0932

Table 1: Comparison with Gibbs sampling in terms of average approximation error.

noise generation	French-English	Chinese-English
SHUFFLE	8.93	21.05
DELETE	9.03	21.49
INSERT	12.87	24.87
REPLACE	13.13	25.59

Table 2: Effect of noise generation in terms of alignment error rate (AER) on the validation sets.

random feature weight vectors. To calculate the true expectations, we follow Dyer et al. (Dyer et al. 2011) to use a dynamic programming algorithm on lattices that compactly represent exponentially many asymmetric alignments. The average errors decrease much less dramatically than in Figure 3 but still maintain at a very low level (below 0.17). This finding implies that the probability mass of log-linear models is still highly concentrated for long sentences.

Table 1 compares our approach with Gibbs sampling. We treat each link l as a binary variable and the alignment probability is a joint distribution over $m \times n$ variables, which can be sampled successively from the conditional distribution $P(l|\mathbf{y}\backslash\{l\})$. Starting with random alignments, the Gibbs sampler achieves an average approximation error of 0.5180 with 500 samples and takes a very long time to converge. In contrast, our approach achieves much lower errors than Gibbs even only using one sample. Therefore, using more likely alignments in sampling improves not only the accuracy but also efficiency.

Alignment Evaluation

We evaluated our approach on French-English and Chinese-English alignment tasks. For French-English, we used the dataset from the HLT/NAACL 2003 alignment shared task (Mihalcea and Pedersen 2003). The training set consists of 1.1M sentence pairs with 23.61M French words and 20.01M English words, the validation set consists of 37 sentence pairs, and the test set consists of 447 sentence pairs. Both the validation and test sets are annotated with gold-standard alignments. For Chinese-English, we used the dataset from Liu et al. (2005). The training set consists of 1.5M sentence pairs with 42.1M Chinese words and 48.3M English words, the validation set consists of 435 sentence pairs, and the test set consists of 500 sentence pairs. The evaluation metric is alignment error rate (AER) (Och and Ney 2003).

The baseline systems we compared in our experiments include

1. GIZA++ (Och and Ney 2003): unsupervised training of IBM models 1-5 (Brown et al. 1993) and HMM (Vogel, Ney, and Tillmann 1996) using EM,
2. Berkeley (Liang, Taskar, and Klein 2006): unsupervised training of joint HMMs using EM,
3. fast_align (Dyer, Chahuneau, and Smith 2013): unsupervised training of log-linear models based on IBM model 2 using EM,
4. Vigne (Liu, Liu, and Lin 2010): supervised training of log-linear models using minimum error rate training (Och 2003).

n	French-English	Chinese-English
1	8.93	21.05
5	8.83	21.06
10	8.97	21.05
50	8.88	21.07
100	8.92	21.05

Table 3: Effect of n in terms of AER on the validation sets.

features	French-English	Chinese-English
local	15.56	25.35
local + non-local	8.93	21.05

Table 4: Effect of non-local features in terms of AER on the validation sets.

As both GIZA++ and fast_align produce asymmetric alignments, we use the *grow-diag-final-and* heuristic (Koehn et al. 2007) to generate symmetric alignments for evaluation. While the baseline systems used all the training sets, we randomly selected 500 sentences and generated noises by randomly shuffling, replacing, deleting, and inserting words. [1]

We first used the validation sets to find the optimal setting of our approach: noisy generation, the value of n, feature group, and training corpus size.

Table 2 shows the results of different noise generation strategies: randomly shuffling, inserting, replacing, and deleting words. We find shuffling source and target words randomly consistently yields the best results. One possible reason is that the translation probability product feature (Liu, Liu, and Lin 2010) derived from GIZA++ suffices to evaluate lexical choices accurately. It is more important to guide the aligner to model the structural divergence by changing word orders randomly.

Table 3 gives the results of different values of sample size n on the validation sets. We find that increasing n does not lead to significant improvements. This might result from the high concentration property of log-linear models. Therefore, we simply set $n = 1$ in the following experiments.

[1] As the translation probability product feature derived from GIZA++ is a very strong dense feature, using small training corpora (e.g., 50 sentence pairs) proves to yield very good results consistently (Liu, Liu, and Lin 2010). However, if we model translation equivalence using millions of sparse features (Dyer et al. 2011), the unsupervised learning algorithm must make full use of all parallel corpora available like GIZA++. We leave this for future work.

system	model	supervision	algorithm	French-English	Chinese-English
GIZA++	IBM model 4	unsupervised	EM	6.36	21.92
Berkeley	joint HMM	unsupervised	EM	5.34	21.67
fast_align	log-linear model	unsupervised	EM	15.20	28.44
Vigne	linear model	supervised	MERT	4.28	19.37
this work	log-linear model	unsupervised	SGD	5.01	20.24

Table 5: Comparison with state-of-the-art aligners in terms of AER on the test sets.

Table 4 shows the effect of adding non-local features. As most structural divergence between natural languages are non-local, including non-local features leads to significant improvements for both French-English and Chinese-English. As a result, we used all 16 features in the following experiments.

Table 5 gives our final result on the test sets. Our approach outperforms all unsupervised aligners significantly statistically ($p < 0.01$) except for the Berkeley aligner on the French-English data. The margins on Chinese-English are generally much larger than French-English because Chinese and English are distantly related and exhibit more non-local structural divergence. Vigne used the same features as our system but was trained in a supervised way. Its results can be treated as the upper bounds that our method can potentially approach.

We also compared our approach with baseline systems on French-English and Chinese-English translation tasks but only obtained modest improvements. As alignment and translation are only loosely related (i.e., lower AERs do not necessarily lead to higher BLEU scores), imposing appropriate structural constraints (e.g., the *grow*, *diag*, *final* operators in symmetrizing alignments) seems to be more important for improving translation translation quality than developing unsupervised training algorithms (Koehn et al. 2007).

Related Work

Our work is inspired by three lines of research: unsupervised learning of log-linear models, contrastive learning, and sampling for structured prediction.

Unsupervised Learning of Log-Linear Models

Unsupervised learning of log-linear models has been widely used in natural language processing, including word segmentation (Berg-Kirkpatrick et al. 2010), morphological segmentation (Poon, Cherry, and Toutanova 2009), POS tagging (Smith and Eisner 2005), grammar induction (Smith and Eisner 2005), and word alignment (Dyer et al. 2011; Dyer, Chahuneau, and Smith 2013). The contrastive estimation (CE) approach proposed by Smith and Eisner (2005) is in spirit most close to our work. CE redefines the partition function as the set of each observed example and its noisy "neighbors". However, it is still intractable to compute the expectations of non-local features. In contrast, our approach cancels out the partition function and introduces top-n sampling to approximate the expectations of non-local features.

Contrastive Learning

Contrastive learning has received increasing attention in a variety of fields. Hinton (2002) proposes contrastive divergence (CD) that compares the data distribution with reconstructions of the data vector generated by a limited number of full Gibbs sampling steps. It is possible to apply CD to unsupervised learning of latent-variable log-linear models and use top-n sampling to approximate the expectation on posterior distributions within each full Gibbs sampling step. The noise-contrastive estimation (NCE) method (Gutmann and Hyvärinen 2012) casts density estimation, which is a typical unsupervised learning problem, as supervised classification by introducing noisy data. However, a key limitation of NCE is that it cannot be used for models with latent variables that cannot be integrated out analytically. There are also many other efforts in developing contrastive objectives to avoid computing partition functions (LeCun and Huang 2005; Liang and Jordan 2008; Vickrey, Lin, and Koller 2010). Their focus is on choosing assignments to be compared with the observed data and developing sub-objectives that allow for dynamic programming for tractable sub-structures. In this work, we simply remove the partition functions by comparing pairs of observed and noisy examples. Using noisy examples to guide unsupervised learning has also been pursued in deep learning (Collobert and Weston 2008; Tamura, Watanabe, and Sumita 2014).

Sampling for Structured Prediction

Widely used in NLP for inference (Teh 2006; Johnson, Griffiths, and Goldwater 2007) and calculating expectations (DeNero, Bouchard-Côté, and Klein 2008), Gibbs sampling has not been used for unsupervised training of log-linear models for word alignment. Tamura, Watanabe, and Sumita (2014) propose a similar idea to use beam search to calculate expectations. However, they do not offer in-depth analyses and the accuracy of their unsupervised approach is far worse than the supervised counterpart in terms of F1 score (0.55 vs. 0.89).

Conclusion

We have presented a contrastive approach to unsupervised learning of log-linear models for word alignment. By introducing noisy examples, our approach cancels out partition functions that makes training computationally expensive. Our major contribution is to introduce top-n sampling to calculate expectations of non-local features since the probability mass of log-linear models for word alignment is usually concentrated on top-n alignments. Our unsuper-

vised aligner outperforms state-of-the-art unsupervised systems on both closely-related (French-English) and distantly-related (Chinese-English) language pairs.

As log-linear models have been widely used in NLP, we plan to validate the effectiveness of our approach on more structured prediction tasks with exponential search spaces such as word segmentation, part-of-speech tagging, dependency parsing, and machine translation. It is important to verify whether the concentration property of log-linear models still holds. Since our contrastive approach compares between observed and noisy training examples, another promising direction is to develop large margin learning algorithms to improve generalization ability of our approach. Finally, it is interesting to include millions of sparse features (Dyer et al. 2011) to directly model the translation equivalence between words rather than relying on GIZA++.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 61331013 and No. 61432013), The National Key Technology R & D Program (No. 2014BAK10B03), Google Focused Research Award, the Singapore National Research Foundation under its International Research Center @ Singapore Funding Initiative and administered by the IDM Programme.

References

Berg-Kirkpatrick, T.; Bouchard-Côté, A.; DeNero, J.; and Klein, D. 2010. Painless unsupervised learning with features. In *Proceedings of NAACL 2010*.

Blunsom, P., and Cohn, T. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of COLING-ACL 2006*.

Brown, P. F.; Pietra, V. J. D.; Pietra, S. A. D.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML 2008*.

DeNero, J.; Bouchard-Côté, A.; and Klein, D. 2008. Sampling alignment structure under a bayesian translation model. In *Proceedings of EMNLP 2008*.

Dyer, C.; Clark, J. H.; Lavie, A.; and Smith, N. A. 2011. Unsupervised word alignment with arbitrary features. In *Proceedings of ACL 2011*.

Dyer, C.; Chahuneau, V.; and Smith, N. A. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of NAACL 2013*.

Gutmann, M. U., and Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*.

Hinton, G. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*.

Johnson, M.; Griffiths, T.; and Goldwater, S. 2007. Bayesian inference for pcfgs via markov chain monte carlo. In *Proceedings of ACL 2007*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL 2007 (Demo and Poster)*.

LeCun, Y., and Huang, F. J. 2005. Loss functions for discriminative training of energy-based models. In *Proceedings of AISTATS 2005*.

Liang, P., and Jordan, M. I. 2008. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of ICML 2008*.

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

Liu, Y.; Liu, Q.; and Lin, S. 2005. Log-linear models for word alignment. In *Proceedings of ACL 2005*.

Liu, Y.; Liu, Q.; and Lin, S. 2010. Discriminative word alignment by linear modeling. *Computational Linguistics*.

Mihalcea, R., and Pedersen, T. 2003. An evaluation exercise for word alignment. In *Proceedings of HLT-NAACL 2003 Workshop on Building and Using Parallel Texts*.

Moore, R. C.; Yih, W.-t.; and Bode, A. 2006. Improved discriminative bilingual word alignment. In *Proceedings of COLING-ACL 2006*.

Och, F., and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*.

Och, F. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*.

Poon, H.; Cherry, C.; and Toutanova, K. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of NAACL 2009*.

Smith, N., and Eisner, J. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL 2005*.

Tamura, A.; Watanabe, T.; and Sumita, E. 2014. Recurrent neural networks for word alignment model. In *Proceedings of EMNLP 2014*.

Taskar, B.; Lacoste-Julien, S.; and Klein, D. 2005. A discriminative matching approach to word alignment. In *Proceedings of EMNLP 2005*.

Teh, Y. W. 2006. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of COLING/ACL 2006*.

Vickrey, D.; Lin, C. C.-Y.; and Koller, D. 2010. Non-local contrastive objectives. In *Proceedings of ICML 2010*.

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