Improving Word Alignment of Rare Words with Word Embeddings

Masoud Jalili Sabet[1] **Heshaam Faili**[1] **Gholamreza Haffari**[2]
[1] School of Electrical and Computer Engineering, University of Tehran, Iran
[2] Faculty of Information Technology, Monash University, Australia
{jalili.masoud,hfaili}@ut.ac.ir
gholamreza.haffari@monash.edu

Abstract

We address the problem of inducing word alignment for language pairs by developing an unsupervised model with the capability of getting applied to other generative alignment models. We approach the task by: *i)* proposing a new alignment model based on the IBM alignment model 1 that uses vector representation of words, and *ii)* examining the use of similar source words to overcome the problem of rare source words and improving the alignments. We apply our method to English-French corpora and run the experiments with different sizes of sentence pairs. Our results show competitive performance against the baseline and in some cases improve the results up to 6.9% in terms of precision.

1 Introduction

Statistical machine translation systems usually break the translation task into two or more subtasks and an important one is finding word alignments over a sentence-aligned bilingual corpus (Brown et al., 1990). One of the common approaches to find word alignment, uses a generative translation model that produces a sentence in one language given the sentence in another language. Most SMT systems use an implementation of the IBM alignment models (Brown et al., 1993). These models use expectation maximization (EM) algorithm in training and only require sentence-aligned bilingual texts. Inducing word alignment from bilingual texts requires large amount of sentence-aligned parallel data which is not available for most of the language pairs. It also causes more problems for rare words which are the key parts of the sentences, but at the same time, are less frequent and therefore have a more biased probability distribution (Moore, 2004).

This paper deals with the alignment task for the rare words by proposing a new alignment model based on the low-dimensional representations of the words. We explore the effect of replacing words with their vector space embeddings in IBM Alignment Model 1. In this model, instead of using a conditional multinomial distribution (to generate a target word $t_i \in T$ given a source word $s_i \in S$), we use a conditional Gaussian distribution and generate a d-dimensional word embedding $V_{t_i} \in \mathbb{R}^d$ given s_i. We then propose a method to **improve the alignments for rare words** by using their similar words and updating their distributions.

The advantages of this model are: *i)* It uses monolingual word embedding which has more available training data; *ii)* By using the extracted knowledge from monolingual data, it has better results in low-resource word alignment tasks; *iii)* The Gaussian model can be applied to all generative alignment models by replacing the conditional distribution, and thus getting even better results.

2 Related Works

2.1 Word Embeddings

Recent works on word embedding show improvements in capturing semantic features of the words (Mikolov et al., 2013; Pennington et al., 2014). Since the word alignment task requires a form of statistics or comparison between words from the source and the target languages, a good translation model

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3209–3215, Osaka, Japan, December 11-17 2016.

can result in better alignments. Bilingual word embedding models like (Zou et al., 2013; Søgaard et al., 2015) train vectors for words in both languages in the same vector space, Hence a translation model can be made by these embeddings and it can be really useful for the task of word alignment. Despite the advantages of bilingual embedding models, to achieve a good performance by these models and building more informative vector representations for words, a large amount of data is required, which is not available in low-resource language pairs. On the other hand, providing a good monolingual corpus for most of the languages needs less effort and building a model that mostly uses monolingual data is more reasonable.

2.2 Alignment Model

IBM alignment models such as model 1, usually have problems with aligning rare words (Moore, 2004). In a sentence pair (S, T), for a target word $t \in T$ and two source words $s, s_{rare} \in S$ when s_{rare} is a rare word and s is a normal word, the translation model will more likely generate $p(t|s_{rare}) > p(t|s)$, and therefore most of the target words will align to s_{rare}. The reason for this problem is the rare word s_{rare} has co-occurrence with only a few target words and it increases the conditional probabilities for those target words. If a source word s which is similar to s_{rare} exists and has co-occurrence with more target words, those target words could be used to improve the distribution of s_{rare}.

There were proposed methods to overcome the problem of low-resource languages based on using different word alignment methods and combining the results like in (Xiang et al., 2010). We try to provide a combination of alignments to get better performance for the rare words.

3 Vector Model

In this section, we develop a conditional model $p(t|s)$ that, given a source word s, assigns probabilities to a target word t using a conditional Gaussian distribution. We explain the required modifications needed in IBM alignment model 1 for using word embedding and briefly review the EM algorithm in this model. We then present the method to use the similar words for updating the distributions of each word and improving the alignment results. The embedding of word $w \in W$ will be written as $\mathbf{v}_w \in \mathbb{R}^d$. The proposed model will be called the *Vector Model*.

3.1 Alignment Model

The IBM alignment models use a translation model in the form of conditional probability $p(t|s)$. In a similar way to (Lin et al., 2015), given a source word $s \in S$, instead of the probability of the target word $t \in T$, the probability of a vector representation $\mathbf{v}_t \in \mathbb{R}^d$ of the word can be calculated. Correspondingly, each source word $s \in S$ is represented by mean μ_s and covariance matrix Σ_s:

$$\begin{aligned} p(t|s) &= p(\mathbf{v}_t|\mu_s, \Sigma_s) \\ &= \frac{\exp\left(\frac{-1}{2}(\mathbf{v}_t - \mu_s)^T \Sigma_s^{-1}(\mathbf{v}_t - \mu_s)\right)}{\sqrt{(2\pi)^d |\Sigma_s|}} \end{aligned} \tag{1}$$

In this model, vector representations are only used for target words and source words are replaced with distributions in the target language vector space.

3.2 EM Algorithm

The EM algorithm should have some modifications to update the Gaussian distributions. The expectation step is similar to the original model. For all sentence pairs (S, T):

$$\forall s \in S, t \in T :$$

$$count(t|s) += \frac{p(t|s)}{\sum_{s' \in S} p(t|s')} \tag{2}$$

$$total(s) += \frac{p(t|s)}{\sum_{s' \in S} p(t|s')} \tag{3}$$

The new conditional probabilities are calculated in the maximization step and for making means and covariances for the source words, these probabilities are used as weights for weighted averaging of the vectors:

$$p(t|s) = \frac{count(t|s)}{total(s)} \tag{4}$$

$$\forall s : \mu_s = \frac{1}{\sum_t p(t|s)} \sum_t p(t|s)\mathbf{v}_t \tag{5}$$

$$\forall s : \Sigma_s = \frac{1}{\sum_t p(t|s)} \sum_t p(t|s)(\mathbf{v}_t - \mu_s)(\mathbf{v}_t - \mu_s)^T \tag{6}$$

In the initialization of the EM algorithm, for all pairs of words, the probability $p(t|s)$ is equal to 1. Therefore, the distributions of source words are made based on their co-occurrences with the target words, which is a similar situation to initialization of the IBM model 1.

3.3 Using Similar Words for Improvement

The proposed model in §3.1 only uses word embedding in the target vector space and each source word corresponds to a distribution. As discussed in §2.2, the distributions of the rare source words will be biased to a small set of target words.

For each source word s, similar source words to s can be found using the cosine similarity between their corresponding vectors. We call the top *k* similar words to each word s, the neighbors of s and represent it as $(N_k(s))$. A weight will be given to each neighbor based on the cosine similarity with the source word:

$$\forall x \in N_k(s) : w_s(x) = \frac{sim(s,x)}{\sum_{x' \in N_k(s)} sim(s,x')} \tag{7}$$

where $w_s(x)$ is the similarity weight of the neighbor word x to the word s and $sim(x,y)$ is the cosine similarity of $\mathbf{v}_x$ and $\mathbf{v}_y$.

In order to use the neighbors for updating the distributions of source words, in the maximization step of the EM algorithm, calculation of the means and covariances will have additional steps:

$$\forall s : \mu_s = \lambda\mu_s + (1-\lambda) \sum_{s' \in N_k(s)} w_s(s')\mu_{s'} \tag{8}$$

$$\forall s : \Sigma_s = \lambda\Sigma_s + (1-\lambda) \sum_{s' \in N_k(s)} w_s(s')\Sigma_{s'} \tag{9}$$

where λ is the linear interpolation parameter and it should be estimated.

4 Experiments

4.1 Data

The language pair used for all the experiments is English-French. The test set is a word-aligned bilingual corpus that contains 447 sentence pairs (Och and Ney, 2000b). Using larger data sets can result in better learning for the model, and in order to create corpora with different sizes, we appended more parallel sentences from the training set to the test data (Note that the proposed model is unsupervised and it does not use the gold alignment in the training). We created corpora with different sizes of sentence pairs and the experiments use these corpora.

For creating the word embeddings, we used the tool word2vec[1] (Mikolov et al., 2013). For the input, we used the English sentences and the French sentences separately and created two sets of vectors. The number of dimensions for vectors was set to 200.

[1] https://code.google.com/p/word2vec/

The models are evaluated using the Alignment Error Rate (AER) and using both possible and sure alignment links. The baseline model is IBM model 1 which has the same learning method and is the most similar model to the proposed one. For making the IBM alignments, the tool Giza++ was used (Och and Ney, 2000a). We believe that the same kind of improvement to this baseline can be achieved by other generative models like IBM Models, by applying the proposed model in this paper.

4.2 Number of Iterations

To see the improvement of the model in each iteration, the 1K corpus is used. Figure 1 illustrates the performance of our vector model compared to IBM model for different number of iterations.

As can be seen from the Figure 1, the proposed vector model has better performance during all iterations of the training and it seems that 5 iterations for training is suitable for both vector model and the IBM model. Therefore, for the rest of the experiments, the models will train for 5 iterations.

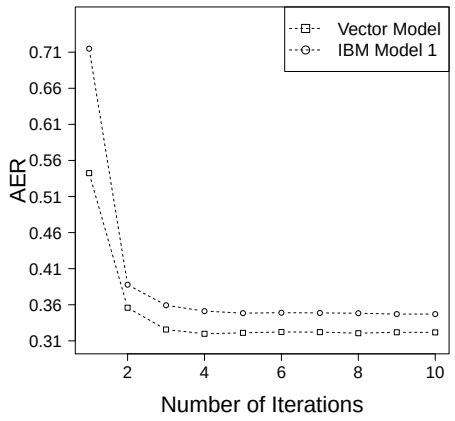

Figure 1: AER as a function of the number of iterations.

4.3 Estimation of the Parameter Lambda

To find the best value for the parameter λ, we used the 1k corpus. The AER is evaluated for different values of λ. Figure 2 shows that the best performance can be achieved by the value 0.4.

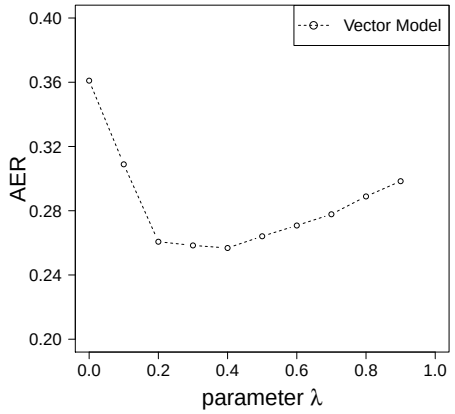

Figure 2: AER of Vector model based on the value of λ.

4.4 Number of Neighbors

To see the improvement with using different number of neighbors for updating the source words, we used the 1k corpus and changed the number of neighbors used for updating the distributions. In this experiment, for the parameter λ we used the value 0.4.

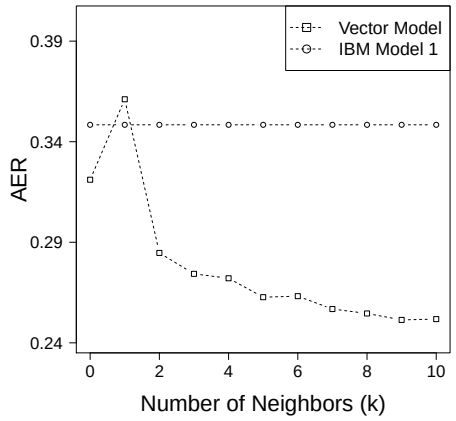

Figure 3: AER based on the number of neighbors.

Figure 3 illustrates the improvement of AER by using more neighbors for updating the distributions. The improvement is up to 9.7% in AER. The results are not improving significantly with more than 7 neighbors, and for the next experiments, the parameter k will be set to 7.

4.5 Size of the Corpus

This experiment is for evaluating the performance of the model based on the size of the corpus. In this experiment, the three models: *a)* IBM model 1, *b)* Vector model using 7 neighbors and *c)* Vector model with no neighbors are trained on the corpora with five different sizes 1k, 5k, 10k, 20k and 100K.

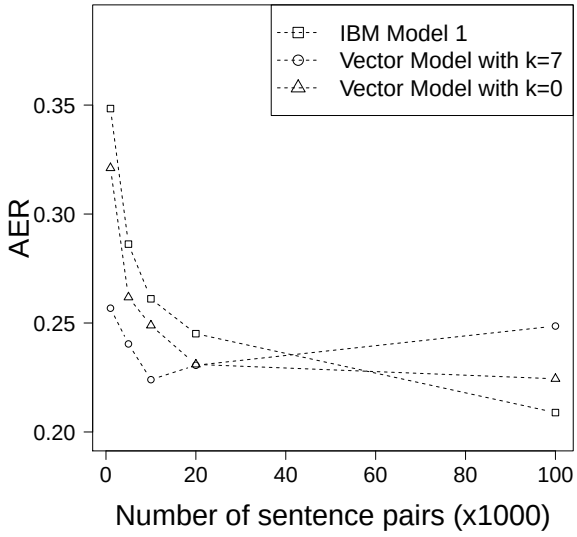

Figure 4: AER of Vector model and baseline based on the size of corpus.

Figure 4 illustrates that the two proposed models are outperforming the baseline for small data sizes. It also shows that the performance of the vector model with 7 neighbors is promising in small corpora

which is useful for low-resource languages. When the size of the corpus grows, the vector model gets to a certain threshold and will stop improving. Using the vector representations for making the translation model can be a good method for small corpora, but in a large corpus, making the probability model is far better than the approximation of the same model.

4.6 Precision on Rare Words

The last experiment is for evaluating the performance of the model based on the precision of the alignments of the rare words. As a definition for rare words, in this experiment, we call a word to be rare if it appears less that 20 times in the corpus. The precision of the alignment for the rare words is calculated by this formula:

$$precision = \frac{\#\text{of correct alignments for rare words}}{\#\text{of alignments for rare words produced by the model}} \tag{10}$$

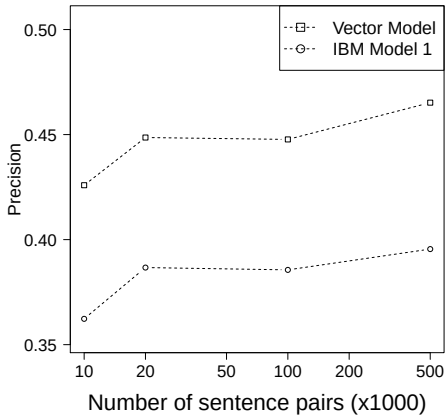

Figure 5: Precision of the Vector model and baseline on rare words based on the size of the corpus.

As shown in Figure 5, the proposed method produces better alignments for the rare words on all different sizes of the corpora.

5 Conclusion

We presented an unsupervised method to find word alignments for language pairs that uses monolingual word embeddings. We then, studied the usage of neighbors for improving the word alignment which made the model more useful for small corpora. We showed that the proposed method finds better alignments for the rare words and outperforms the baseline on different sizes of the corpora. Using the neighbors improved the performance of the model for rare words up to 6.9% compared to the baseline. The proposed Vector model has the capability of being applied to other generative alignment models which is not studied yet. Our work could be extended to other IBM models in the future works.

References

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

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.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. *arXiv preprint arXiv:1503.06760*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Robert C Moore. 2004. Improving ibm word-alignment model 1. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 518. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2000a. Giza++: Training of statistical translation models.

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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*.

Bing Xiang, Yonggang Deng, and Bowen Zhou. 2010. Diversify and combine: Improving word alignment for machine translation on low-resource languages. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 22–26. Association for Computational Linguistics.

Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.