Improving Word Alignment by Adding Gromov-Wasserstein into Attention Neural Network

Yan Huang\(^1\)\(^\ast\), Tianyuan Zhang\(^1\), Huidong Zhu\(^2\)
\(^1\)Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, Henan, China
\(^2\)College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan, China
\(^\ast\)email: hy_nicer@zzuli.edu.cn

Abstract. Statistical machine translation systems usually break the translation task into two or more subtasks and an important one is finding word alignments over a parallel sentence bilingual corpus. We address the problem of introducing word alignment for language pairs by developing a novel neural network model that can applied to other generative alignment models. We use Multi-layer attention model and multi-layer model with multi-head-attention mechanism on each layer provides superior translation quality. It can be trained on bilingual data without relying on word alignment. In this paper, we cast the correspondence problem directly as an optimal distance problem. We use the Gromov-Wasserstein distance to calculated how similarities between word pairs are related across languages. The resulting alignments dramatically outperform the GIZA++ and FastAlign approach, these alignments are comparable on public data sets.

1. Introduction
Word alignment is indispensable for SMT (Statistical Machine Translation), SMT usually divides the translation task into multiple subtasks. Finding word alignments over a sentence-aligned bilingual corpus is one of the most important subtasks[1]. Word alignment from bilingual data requires large amount of parallel data for sentence alignment. For rare words that are the key parts of the sentences, doing so will cause some problems. At the same time, they have a more biased probability distribution.

One of the common ways to find word alignment is using a generative translation model, which generates a sentence in one language given the sentence in another language. Giza++ and FastAlign [2] are the most widely used tools for word alignment.

We propose an innovative word alignment approach, which is based on Gromov-Wasserstein distance for word alignment. Compared with Giza++ and FastAlign, it’s more efficient and accurate. This approach measures how distances are mapped across languages between pairs of words. We use a alignment layer in the decoder on the basis of Transformer. It’s trained to predict the next target word based on a linear combination of the encoder information. We have two separate output layers, each layer defines a probability distribution of the next target word. Then, we use Gromov-Wasserstein distance that measures whether the source outputs and target outputs are alignment or not. We validate our approach on Uyghur-Chinese data sets and compare AER (the Alignment Error Rate) to Giza++ and FastAlign. Our method improves the alignment quality by three times over the baselines On the Uyghur-Chinese data sets. As the stems and affixes of Uyghur affect the performance of Uyghur-Chinese alignments, we test the performances that whether stem segmentation or not.

2. Related works
Recent works on word embedding has shown improvements in capturing the semantic features of words[3-4]. Alkhouli et al. uses the supervised data generated by Giza++ to train a single alignment head of Transformer[5], so that its attention directly corresponds to alignments. Then this improved attention is used to improve dictionary-guided translations. Previously, Ma et al. group words used monolingual data to cluster words to obtain alignments[6]. Then they copy their parallel data and use alignments on word classes.

Arthur et al. add a lexical probability vector which is generated based on the information of a discrete lexicon, and use the attention vector to determine the lexical probabilities of which source word the model should focus on. Tamura et al. directly predict the alignments based on a recurrent neural network[7], which is conditioned on the source sequences and the target sequences.

3. Proposed method

In this section, we introduce the word-alignment based transformer architecture and Gromov-Wasserstein distance. Then we illustrate the details and training techniques of our proposed model.

![Fig. 1 General architecture of hierarchical attention neural networks for alignments](image)

3.1. Architecture

In order to train our alignment component in an unsupervised way without relying on word alignment training data, we add an alignment layer on the top of decoder. In our research, we used LSTM cells because they showed better performance in the experiment. We adopt a general hierarchical attention architecture, displayed in Figure 1, which is proposed by Yang et al.[8].

Our architecture is general a hierarchical structure that can adapted to different types of components, such as attention models, encoders. We consider a dataset \( D = \{ (s_i, t_i), i = 1, ..., N \} \) made of \( N \) parallel sentence pairs \((s_i, t_i)\). Each parallel sentence pairs is represented by the \( d \)-dimensional embedding sequence of the sentence composed of their words, \( s_i = \{ w_{i1}, w_{i2}, ..., w_{ik} \}, \) \( k \) is the maximum number of words in a sentence.

To provide inputs to neural network layers, in the first layer, we use a lookup table to convert words into word embeddings. In Equation 1, \( W \) is the one-hot vector of the words in sentences whose uses the pre-trained embedding matrix \( E \) to generates word vector \( W_i^s \) in the \( i^{th} \) sample sentence in first step.

\[
W_i^s = E^T W
\]  

3.2. Recurrent Neural Network layers

The neural network layers are made up of feed-forward module and feed-backward module. We use BiRNN (bidirectional RNN) with recurrent activation functions[9]. Since we want to share vector representations in the vector space, we need to use siamese network. Then input the source sentence and target sentence representations into a feed-forward network with a output layer containing sigmoid function, this layer is to calculate the context of words.

2
For the target sentence, simply replace S to T. The forward RNN layer receives word vectors and then generates vector sequences \( h_{it} \) for each time step. A similar RNN generates a backward RNN vector \( h_{rt} \) in reverse order.

\[
\vec{h}_{it} = \frac{\exp (u_{it}^T u_w)}{\sum_{t=1}^{T} \exp (u_{it-1}^T u_w)} \quad \text{(4)}
\]

\[
\alpha_t^\gamma = \sum_{t=1}^{T} \alpha_{t-1}^\gamma \quad \text{(5)}
\]

\[
w_t^\gamma = \sum_{t=1}^{T} \alpha_{t}^\gamma h_{rt}^{\gamma} \quad \text{(6)}
\]

We obtain a given word’s annotation by concatenating the forward hidden state \( h_{it} \) and backward hidden state \( h_{rt} \). \( h_t^\gamma = \begin{bmatrix} \vec{h}_{it} & \vec{h}_{rt} \end{bmatrix} \) summarizes the information of the whole sentence centered around \( w_t^\gamma \).

### 3.3. Attention layers

Not all words contribute equally to words in a target sentence. We use the attention mechanism and introduce vector \( u_{it} \). The purpose of this vector is measuring the importance of the sentence.

\[
\alpha_t^\gamma = \frac{\exp (u_{it}^T u_w)}{\sum_{t=1}^{T} \exp (u_{it-1}^T u_w)} \quad \text{(5)}
\]

\[
w_t^\gamma = \sum_{t=1}^{T} \alpha_{t}^\gamma h_{rt}^{\gamma} \quad \text{(6)}
\]

We get a normalized importance weight \( \alpha_t^\gamma \) through a softmax function. Then we compute the next word vector \( w_t^\gamma \) according to the weight as the weighted sum of the previous word annotations.

### 3.4. Alignment layer

The outputs of previous networks are typically used by an alignment layer to align words, the loss based on the negative log-likelihood of the correct labels[8] or on the cross-entropy between gold and predicted labels[10]. However, this may not be ideal for multi-label optimization.

In this paper, we treat alignment as a distance optimization problem. We use the Gromov-Wasserstein distance to define the optimal transmission by directly comparing samples in the metric space instead of across the space. Generally, we treat alignment as a distance optimization problem. For the source and target sentences, we calculate alignment by minimizing the distance of word vectors. As follow:

\[
GW(s,t) = \min \sum_{x \in \Pi(s,t)} \sum_{y \in \Pi(s,t)} \mathbb{I}[x \in (s,t)] \mathbb{I}[y \in (w_t^\gamma, w_j^\gamma)] ||w_t^\gamma - w_j^\gamma|| \quad \text{(7)}
\]

\[
P(s|t) = GW(s,t) \quad \text{(8)}
\]

Where \( \Pi(s,t) \) represents a set of all possible joint probability distributions of word vectors in source and target sentences. Under this joint distribution, the expectation of all s and t distances can be obtained. There is a joint distribution that minimizes this expectation. The expected lower bound (infimum) is the Wasserstein distance of \( GW(s,t) \).

### 4. Experiment

Our evaluate goal is to compare statistical alignment methods (Giza++ and FastAlign), with a new alignment approach introduced in this paper. We attempt to use the same training data for all methods to achieve the purpose of fair comparison. We adopt public training data and testing data in Uyghur-Chinese language pairs.

#### 4.1. Data

We assess our method on two standard benchmark tasks with cross-lingual embeddings. We used the
CWMT\textsuperscript{15} and CWMT\textsuperscript{17} datasets containing a total of 508,076 sentences in Chinese translated to Uyghur. We use Moses toolkit\textsuperscript{2} to preprocess the training data. For the Chinese, we use Jieba\textsuperscript{3} for word segmentation. For the attention neural network model, we apply byte pair encoding\textsuperscript{8}. For FastAlign and Giza++ we concatenate the training data and testing data. Table 1 summarizes the training data.

| Tab. 1 Training data statistics |
|---------------------------------|
| Data size | CWMT\textsuperscript{15} | CWMT\textsuperscript{17} |
| Words source | 108M | 413M |
| Words target | 136M | 461M |
| sentences | 13M | 37M |

5. Results

In our models, We use 128-dimensional LSTM for all RNN, and we use GloVe word vectors for the embedding layer\textsuperscript{4}. To prevent over-fitting, we use a constantly changing dropout rate in each module. We test different iterations to evaluate how it affects alignment.

5.1. Machine translation results

We conduct end-to-end MT experiment on Uyghur-Chinese language pairs. We use Moses to train and decode PBMT (Phrase Based Machine Translation) system. We tuned and tested each system three times to eliminate the effect of randomness, and reported the average scores. Table 2 shows the BLEU scores in different experiments. We tested different experimental results in CWMT\textsuperscript{15}, CWMT\textsuperscript{17} sets separately. To further test the model, we add the two data sets together. It should be noted that we first conduct experiment without the stem affixation in Uyghur language.

| Tab. 2 Machine translation experiments (BLEU) |
|---------------------------------------------|
| Method | CWMT\textsuperscript{15} | CWMT\textsuperscript{17} | CWMT\textsuperscript{15}+17 |
| Giza++ | 22.78 | 27.63 | 32.63 |
| FastAlign | 23.61 | 27.67 | 33.13 |
| Ours | 25.43 | 28.68 | 35.57 |

As it can be seen, the proposed method in this paper obtains very strong results, reaching 3-4 BLEU points in Uyghur-Chinese depending on the variant and direction. This is much stronger than the baseline system of Giza++ and FastAlign, with improvements of at least 10% in all cases. In order to further test our proposed method can effectively improve the quality of the machine, we need to test our method to effectively improve the accuracy of word alignment, because the word alignment accuracy can effectively improve the machine translation quality.

5.2. Alignment results

We perform word alignment experiments on Uyghur-Chinese, for which we have access to the best alignment data. We append the test sentences to the existing parallel training data to get the alignments. Baseline and proposed methods are defined in the machine translation experiments above. Table 3 presents the precision of the word alignments.

| Tab. 3 Word alignment experiments |
|----------------------------------|
| Methods | CWMT\textsuperscript{15} | CWMT\textsuperscript{17} | CWMT\textsuperscript{15}+17 |
| Giza++ | 59.7 | 61.0 | 63.8 |
From the above tables, we can find that our method get a significant improvement over the two baselines. First, we think that our system uses a more advanced match mechanism for bilingual word alignment. Our method can obtain more sufficient context information, at the same time, when calculating bilingual semantic, we can more fully consider multiple information characteristics, not only in space and time. From the above tables, we can find that it can improve word alignments accuracy by increasing training corpus. However, our method can get a higher accuracy of word alignment at the same corpus scale no matter in any case. In order to conduct word alignment, we need to iterate multiple times to get better results. But it iteratively consumes a lot of computing resources and takes a long time, to prove that our method is better in this respect, we also carry out the following experiments to detail this problem.

5.3. Number and time of iterations
In order to observe the improvement of our model in each iteration, we use CWMT15 data to perform this experiment. Figure 1 shows the model compared to Giza++ and FastAlign in different number and time of iterations.

![Fig. 2 Accuracy as different number and time of iterations.](image)

We can observe through Figure 2 that our model has achieved better results during iterations. At the same time, our method can have a faster convergence speed. In the process of initial iteration, as our methods use a neural network model, the parameters are randomly initialized in the neural network model. It makes the poor performance of our model at the beginning, and the results obtained are not ideal against the two baseline systems. But in the process of continuous iteration, our model is continuously optimized to make the performance better and better, and the convergence speed is faster than the two baselines. Watching time and the number of iterations, we find that the convergence of time is significantly faster than the number of iterations. We think that our model randomly selects a bunch of data from the entire training data in iteration. However, the two baseline systems train the entire training dataset as one iteration.

5.4. Qualitative analysis of alignment
In order to intuitively observe what factors affect the alignments of our method, we use the diagram for analysis. We will analyze alignments and the resulting attention activations by randomly selecting an example of a parallel sentence. As mentioned earlier, the optimal value problem of Gromov-Wasserstein can reasonably explain the distance between their word embedding spaces. The results are as for Figure 3.
Fig. 3 Visualization of different models when translating from Uyghur to Chinese alignments
In Figure 3, we use shades of black squares to represent different alignment weights. We define that the darker the color represents the greater the weight of the alignment. Overall, these alignments conform to our intuitions: our method can get a better alignment. But somewhat surprisingly, the alignment effect of the real word is significantly higher than the alignment of the virtual words. We speculate that this may be due to the rich morphology of Uyghur language. The words have a very rich form. An Uyghur word can be expressed by a stem and multiple affixes. Many Chinese words can be expressed by adding affixes in Uyghur language, resulting in a serious one-to-many phenomenon and reducing the accuracy of sentence-to-word alignment and seriously affects the quality Uyghur-Chinese translation. Chinese virtual words often are represented as affixes in Uyghur language.

6. Discussion
Table 1 and 2 show that the quality of machine translation can be improved by adding training corpus. At the same time, Translation quality is different under different alignment methods. As we use different sizes of training data, we can’t conclude that alignment has an important influence on machine translation. In order to further test our proposed method can effectively improve the quality of the machine, we need to test our method to effectively improve the accuracy of word alignment. From Table 3, we can find that the accuracy of alignment is different in three methods. Our method can get a higher score of alignment. Combined with Table 2 and 3, we can conclude that the alignment accuracy is directly proportional to the machine translation score. Moreover, our method can get a higher alignment accuracy, which also leads to a higher machine translation score.

Alignment is an extremely time-consuming training process, and we must guarantee training time when improving word alignment accuracy. From Figure 1, we can see that our method can converge in a shorter time. In order to analyze our method how gets a better alignment, we conduct a qualitative analysis in Figure 3. From Figure 3, our method can get more certain results in the alignment of real words. The weight of alignment is higher in ours, and our method has fewer candidates in the alignment of virtual words. So it can be said that the short calculation cost is very good.

7. Conclusion
This paper solves the problem of extracting word alignments from the Self-Attention NMT model combined a novel match calculating. We extend the neural network with an alignment layer, it contains Gromov-Wasserstein distance generalizes optimal alignment. We conduct experiments on Uyghur-Chinese machine translation. These results indicate that appropriate word alignment in improving the MT output.

Acknowledgments
This research is supported by the Henan Science and Technology Research Project (212102210075).

References
[1] Brown P F, Cocke J, Della Pietra S A and et al 1990 A statistical approach to machine translation Computational linguistics 16(2)
[2] Dyer C, Chahuneau V and Smith N A. 2013 A simple, fast, and effective reparameterization of
ibm model 2 Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies pp 644-648

[3] Mikolov T, Sutskever I, Chen K and et al 2013 Distributed representations of words and phrases and their compositionality Advances in neural information processing systems pp 3111-3119

[4] Pennington J, Socher R and Manning C 2014 Glove: Global vectors for word representation Proceedings of the 2014 conference on empirical methods in natural language processing pp 1532-1543

[5] Alkhouli T, Bretschner G and Ney H 2018 On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation arXiv preprint arXiv 1809.03985

[6] Jeff MA, Matsoukas S and Schwartz R 2011 Improving low-resource statistical machine translation with a novel semantic word clustering algorithm Proceedings of the MT Summit XIII pp 353-355

[7] Tamura A, Watanabe T and Sumita E 2014 Recurrent neural networks for word alignment model Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics vol 1 pp 1470-1480

[8] Yang Z, Yang D, Dyer C and et al 2016 Hierarchical attention networks for document classification Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies pp 1480-1489

[9] Cho K, Van Merriënoor B, Bahdanau D and et al 2014 On the properties of neural machine translation: Encoder-decoder approaches arXiv preprint arXiv 1409.1259

[10] Tang D, Qin B and Liu T 2015 Document modeling with gated recurrent neural network for sentiment classification Proceedings of the 2015 conference on empirical methods in natural language processing pp 1422-1432

[11] Shaolin Z, Yating Y, Chenggang Mi, Xiao L and Lei W 2017 Learning Bilingual Lexicon for Low-resource Language Pairs National CCF Conference on Natural Language Processing and Chinese Computing pp 760-770

[12] Alvarez-Melis D and Jaakkola T S 2018 Gromov-Wasserstein Alignment of Word Embedding Spaces arXiv preprint arXiv 1809.00013

[13] Nguyen T Q and Chiang D 2017 Improving lexical choice in neural machine translation arXiv preprint arXiv 1710.01329

[14] Tamer Alkhouli, Gabriel Bretschner and Hermann Ney 2018 On the alignment problem in multi-head attention-based neural machine translation Proceedings of the Third Conference on Machine Translation

[15] Conneau A, Lample G, Ranzato M A and et al 2018 Word translation without parallel data arXiv preprint arXiv 1710.04087

[16] Dou Z Y, Zhou Z H and Huang S 2018 Unsupervised Bilingual Lexicon Induction via Latent Variable Models Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing pp 621-626