Integrating Word Embeddings into IBM Word Alignment Models

Anh-Cuong Le*
Faculty of Information and Technology
Ton Duc Thang University
Ho Chi Minh City, Vietnam
leanhcuong@tdtu.edu.vn

Tuan-Phong Nguyen, Quoc-Long Tran
Department of Computer Science
University of Engineering and Technology, VNU
Ha Noi, Viet Nam
phongnt@vnu.edu.vn, longtq@vnu.edu.vn

Dao Bao Linh
GMO-Z.com Runsystem, Ha Noi, Vietnam
linhdb@runsystem.net

***Abstract*—Word alignment models are used to generate word-aligned parallel text which is used in statistical machine translation systems. Currently, the most popular word alignment models are IBM models which have been widely applied in a large number of translation systems. The parameters of IBM models are estimated by using Maximum Likelihood principle, i.e. by counting the co-occurrence of words in the parallel text. This way of parameter estimation leads to the "ambiguity" problem when some words stand together in many sentence pairs but each of them is not translation of any other. Additionally, this method requires large amount of training data to achieve good results. However, parallel text which is used to train the IBM models is usually limited for low-resource languages. In this work, we try to solve these two problems by adding semantic information to the models. Our semantic information is derived from word embeddings which only need monolingual data to train. We deploy evaluation on a language pair that has great differences in grammar structure, English-Vietnamese. Even with this challenged task, our proposed models gain significant improvements in word alignment result and help increasing translation quality.**

***Index Terms*—IBM models, word embeddings, word alignment , Vietnamese, bilingual mapping**

I. Introduction

Statistical Machine Translation (SMT) has been very successful in the last two decades since the introduction of IBM models [1]. IBM models are a sequence of models in which lower models' parameters are used to initialize higher ones. Those models are used to train a word alignment model to construct a translation model. Although IBM translation models are not being utilized in competitive translation systems nowadays, the IBM word alignment models are still a crucial component of many modern SMT systems, e.g. Moses [2] and Hiero SMT [3].

Many works had been done in order to strengthen the classic IBM models which is data-driven approach and can lead to the "garbage collection" problem for rare words. [4] used a smoothing technique to solve this problem. Another weakness of IBM models is that they are only able to produce one-to-many alignments but the fact is that alignments are frequently many-to-one or many-to-many. One effective technique to overcome this weakness is to combine the word alignments of both directions produced by the IBM models. [5] proposed various heuristics to solve that.

Another problem of IBM models is their weak ability to deal with ambiguity cases. When some words stand together in many sentence pairs but they are actually not translational correspondences, the models will be confused to determine the true alignment. This problem and the "garbage collection" problem can be solved by using linguistic knowledge. Recently, many researchers have involved in learning to incorporate semantic information into SMT systems to generate grammatical and meaningful translations and their works have provided promising results. That leads us to the idea of integrating lexical semantics to solve these problems and consequently enhance the IBM word alignment models.

There have been some effective ways to add lexical semantics to Natural Language Processing (NLP) tasks in general. Recent success of distributed representations of words gives us a straightforward way to compute the semantic similarity between words. Especially, the notable work of [6] provides us a useful technique to exploit the similarity of word pairs among two lan-

*Corresponding author

978-1-5386-6113-0/18/$31.00 ©2018 IEEE

guages. This technique does not require any bilingual corpus which is typically very limited for low-resource languages. More specifically, we will learn a linear projection between two vector spaces of two languages and then apply some simple operations to effectively compute similarity scores of words. These scores are clearly a potential factor that may help improving the IBM word alignment models.

Our method is to use the semantic similarity scores to adjust the parameters of IBM models. The scores will be added directly into the parameter updating operation in M-step of the EM algorithm. In spite of the simplicity, our method is proved to be effective. Even though our evaluation is deployed on English-Vietnamese language pair which has big differences in grammar structure, the enhanced models provide very positive results. We use Alignment Error Rate (AER) [7] as the metric to evaluate the word alignment quality. The AER of our models reduces significantly. Next, we analyze the effect of the improved alignment models on translation quality. We evaluate on phrase-based translation systems of Moses [2]. The translation systems also benefit from the improvements of word alignment quality. We gain increments up to 0.39 for the BLEU score metric [8] for the challenged language pair, English and Vietnamese.

We begin by representing about IBM word alignment models and its role in SMT. We then propose our method to integrate semantic similarity derived from word embeddings into IBM models to augment the word alignment models. In the next part, we demonstrate the experiments on English-Vietnamese translation to confirm the usefulness of our method and discuss the results. Finally, we make conclusion about this work.

II. IBM WORD ALIGNMENT MODELS AND SMT

IBM models were introduced by [1] and started the successful era of SMT. IBM models are a sequence of models with increasing complexity used to train a translation model and an alignment model. The original work of [1] proposed five models, starting with lexical translation probabilities in Model 1, then adding models for reordering and word duplication in the higher models. IBM word alignment models are still the seeds for almost currently competitive SMT systems. In these models, we defined an alignment of a sentence pair as a function $\mathbf{a} : j \rightarrow i$ that maps each foreign word position j to an English word position i.

In every IBM model, there is a specific set of free parameters. For example, Model 1 consists of only one parameter which is the translation probability $p(f|e)$; Model 2 adds the alignment parameter $a(i|j, l, m)$; Model 3 adds the distortion, null insertion and fertility parameters; etc. We denote the set of parameters as θ. We pursue maximum likelihood principle to find the parameter values that maximize the log-likelihood of the training data that consists of N sentence pairs $\{(\mathbf{e}_n, \mathbf{f}_n) : n = 1, ..., N\}$, where $\mathbf{e}_n$ and $\mathbf{f}_n$ are corresponding translations for every n. The estimated parameter set is:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} L(\theta) \tag{1}$$

Whereas, the log-likelihood $L(\theta)$ of the training data is defined as follow:

$$\begin{aligned} L(\theta) &= \sum_{n=1}^{N} \log Pr(\mathbf{f}_n|\mathbf{e}_n; \theta) \\ &= \sum_{n=1}^{N} \log \sum_{\mathbf{a}} Pr(\mathbf{f}_n, \mathbf{a}|\mathbf{e}_n; \theta) \end{aligned} \tag{2}$$

In IBM models, we have the alignment $\mathbf{a}$ as missing value that was not observed in the training data so that we cannot directly estimate the parameters that maximize $L(\theta)$. Instead, we apply the Expectation Maximization (EM) algorithm iteratively in order to approach a local maximum of $L(\theta)$.

A. Using word alignment models in SMT

Modern machine translation systems are no longer word-based models. However, generative modeling and the EM algorithm are still playing important roles in currently competitive approaches. For instance, the phrase-based model of [5] uses similar EM algorithm to the IBM models' to estimate its parameters.

In phrase-based translation, systems translate phrases in source language to phrases in target language. Hence, these systems have to learn a phrase translation table which contains possible phrases and probabilities of phrase-to-phrase translation. First, we use trained word alignment model to generate alignments for bilingual corpus using the Viterbi alignment:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \tag{3}$$

Next, the word-aligned bi-text is used to extract phrases for the models due to the following rule: two phrases are aligned if the words within them are aligned. We set the maximum length of phrases at M

words to avoid unnecessary complexity. After that, we have a set of English phrases ν_E and a set of foreign phrases ν_F. For each English phrase $u \in \nu_E$ and each foreign phrase $v \in \nu_F$, we use Maximum Likelihood principle to estimate the phrase-to-phrase translation probability of $(v|u)$ as follow:

$$p(v|u) = \frac{c(u,v)}{\sum_{v' \in \nu_F} c(u,v')}, \tag{4}$$

where $c(u,v)$ is the number of times that phrase u and phrase v are aligned in the training corpus.

III. Word Embeddings

Semantic information is obviously a useful factor for Natural Language Processing. Alternatively, words can be represented as vectors in a semantic space. This approach is also called as distributed representations of words, or word embeddings. The simplest way for this approach is to represent each word in the vocabulary as a one-hot vector, which contains only a one in a position and zeros in all other positions. The dimensionality of these vectors equals to the vocabulary size. Using this technique can overcome the "hand-crafted" drawback of semantic networks, but its vectors still cannot provide useful evidence to evaluate similarity between words. Moreover, the high dimensionality of one-hot vectors is a big shortcoming. Representing words in this way is also unable to deal with out of vocabulary (OOV). Using continuous vectors can help to solve most of those problems. In this work, we focus on this way of semantic representation to produce valuable information. Each dimension of word vectors represents a latent semantic and their dimensionality is usually low. This enables us to compute the similarity between words by applying typical vector similarity measures.

Word embeddings were firstly proposed by [9]. Their applications can be found in many NLP tasks such as statistical language modeling, named entity recognition, parsing and word sense disambiguation. [10] released word2vec containing two models (continuous bag-of-words - CBOW and skip-gram) which are both neural network implementations for learning distributed representations of words. This was a turning-point of this topic as word2vec is able to train on a large corpus (with billions of words) in just some hours on a single desktop computer and brings many interesting results. Because of its robustness, word2vec has been included in many researches since its introduction. In our work, we also use word2vec to produce semantic spaces.

The semantic spaces learned by word2vec on a large dataset capture a significant amount of semantic information [10]. First, related words that appear in similar contexts many times such as "school" and "university", "lake" and "river" have similar vector representations. Moreover, one another notable relationship allows us to do a linear operation like that "King" - "Man" + "Woman" is closest to "Queen". More interestingly, by using the skip-gram model, the output vectors also have an additive property, e.g. the result of "Vietnam" + "capital" is closest to "Hanoi".

word2vec can train very fast by using additional techniques such as hierarchical softmax and negative sampling. The details of these techniques are fully described in the original paper of [11].

IV. Integrating word embeddings into IBM models

A. Evaluating semantic similarity of words among languages

The objective of bilingual mapping is to learn a function that maps a word vector of a semantic space to its translation's vector representation in the other semantic space. This approach was firstly discovered by [6] and then was extended by [12]. This approach is quite simple and only requires monolingual corpora and a small bilingual dictionary to learn the mapping function.

[6] proposed a potential method to generate bilingual dictionaries using distributed representations. Their method is mainly based on the observation that the vectors in two vector spaces of two languages, e.g. the vectors for numbers and animals in English and Spanish, have similar geometric arrangements. Hence, they suppose that the relationship between two vector spaces then can be captured by linear mapping. First, a small bilingual dictionary is used to learn the transformation matrix between the two vector spaces. For more details, suppose that we have a set of word pairs and their corresponding vector representations $\{x_i, z_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{d_1}$ is the representation of word i in the source language, and $z_i \in \mathbb{R}^{d_2}$ is the representation of its translation. Our goal is to find a transformation matrix W such that Wx_i approximates z_i. [6] treated this task as an optimization problem:

$$\min_{W} \sum_{i=1}^{n} \|Wx_i - z_i\|^2 \tag{5}$$

and solve it by using Stochastic Gradient Descent. Once we had the transformation matrix, we can predict the translation of any given new word whose vector representation is x by computing $z = Wx$. The prediction is the word whose representation is closest to z in the target language's vector space. In spite of its simplicity, the experiments in [6] shows that this method provides us very promising results, e.g. they can achieve almost 90% precision@5 for translation of words between English and Spanish. Moreover, they also demonstrated that the method works well even for pairs of languages that are not closely related such as English and Czech, English and Vietnamese.

Using the model proposed by [6], we are able to compute similarity score between any word pair of two languages. Given a word pair (e, f) and their two vector representations $\vec{v}_e$, $\vec{v}_f$ and the learned transformation matrix W, we first compute the *distance* between e and f by the Euclidean distance between the estimated vector representation of the translation of e and the vector representation of f in the vector space of the foreign language, which is

$$D(e, f) = distance(W\vec{v}_e, \vec{v}_f) \tag{6}$$

Next, we refine the distances by this operation:

$$d(e, f) = \frac{D(e, f)}{\max_{(e', f')} D(e', f')} \tag{7}$$

so that d will lie in $[0, 1]$. Next, we compute the similarity between this pair of words as follows:

$$similarity(e, f) = \alpha^{d(e,f)} \tag{8}$$

where α is a number between $[0, 1]$. This formula simply means that a pair of words which are not closely related, or in other words, have larger distance will produce smaller similarity score. Two words having great similarity score can be semantically equivalent. We use the operations in Equation 7 and 8 instead of cosine similarity because we observed that the scale of the distances from English words to a given foreign word is quite narrow. In other words, the gap between the greatest and the smallest distances of English words to a given foreign word is quite small in comparison with the distances. Hence, we use the above equations to widen that scale of distances. Then, we normalize the similarity scores to produce a proper probability distribution:

$$s(e, f) = \frac{similarity(e, f)}{\sum_{e'} similarity(e', f)} \tag{9}$$

B. Adding similarity scores of words into IBM models

We now have a valuable semantic factor presenting the similarity of two words among two languages. Finally, we integrate this factor into IBM models by adjusting the count function in Equation 10 as follow:

$$c(f|e; \mathbf{e}, \mathbf{f}; p) = \frac{p(f|e)s(e, f)}{\sum_{i=0}^{l} p(f|e_i)s(e_i, f)} \sum_{j=1}^{l} \delta(f, f_j) \sum_{i=0}^{m} \delta(e, e_{a(j)}) \tag{10}$$

This is the modification for IBM model 1. Higher IBM models can be adjusted in the same way. In Equation 10, it can be explained that we add similarity scores along with the existing parameters of IBM models. It means that the lexical translation parameter will be fine-tuned by the semantic information derived from word embeddings. In other words, the word pair (e, f) that has high similarity score detected by word embeddings will have bigger count in the M-step of the EM algorithm. We believe that using this modification gives smoother lexical translation parameter and consequently helps the process of learning phrase translation table of phrase-based translation models. Despite the simplicity of our techniques, we will show that our boosted IBM models can considerably outperform the original ones.

V. Experiments

Our evaluation is conducted on translation between English (EN) and Vietnamese (VN). Translation between this language pair is a challenged task because they have many differences in grammar structure and the amount of parallel text containing Vietnamese is limited.

A. Experimental setup

First, we collected monolingual corpora for English and Vietnamese to train semantic spaces and language models. We used UETSegmenter [13] to run word segmentation for the Vietnamese corpus. After preprocessing steps including tokenization, word segmentation and case-normalization, each corpus consists of about 400m to 500m words with the vocabulary of nearly 50k words. The word embeddings were learned using word2vec implementation by Mikolov[1]. In practice, we just used CBOW model for faster training. Next, we manually built a bilingual dictionary

[1]https://github.com/tmikolov/word2vec

of 700 word pairs to learn the transformation matrices between English and Vietnamese vector spaces. The language models were trained using KenLM [14]. These models were used in all of our experiments.

We evaluated our boosted models and the classic IBM models in terms of both word alignment and translation quality. We have two bilingual corpora in our experiments. The first one was taken from [15] which contains over a million of movie subtitle sentence pairs collected from OpenSubtitles[2]. We selected a sub-corpus of 200k sentence pairs from this data. The second one came from the work of [16] which consists of 290k sentence pairs collected from Wikipedia and some English-learning textbooks. After pre-processing steps for these corpora, we divided each of them into parts of 10k, 20k, 50k and 100k sentence pairs to train word alignment models. Each entire corpus then was used to train phrase-based translation systems.

We utilized two popular tools in our experiments, GIZA++ [7] which implements IBM models and Moses [2] which implements phrase-based translation model. We first modified GIZA++ to add our boosted models. Each IBM model was trained in 5 iterations. The word alignment process was performed in both directions using GIZA++. These alignments were then combined with "grow-diag-final-and" heuristics by Moses. The phrase bi-lexicon was then automatically derived by Moses using the final word alignment. For word alignment measurement, we compute Alignment Error Rate (AER) [7] on a test set of 300 sentence pairs which is manually word-aligned. It is worth noting that we only compute AER for the word alignment models trained in EN-VN direction, the bi-direction alignment is only used by Moses when generating the phrase translation table. In evaluation scheme for translation, we train the classic IBM model 4 to process word alignment for the baseline systems. Our boosted systems use IBM model 4 with improved model 1 and 2. After the word alignment step, we use Moses to learn phrase-based translation models and then tuned the parameters of these models using Minimum Error Rate Training (MERT) algorithm [17]. Finally, we apply BLEU score metric [8] to evaluate the translation systems. The test set for OpenSubtitles corpus contains 3k sentence pairs, the one for Wikipedia corpus contains 5k sentence pairs. Note that we set α in Equation 8 to 0.01 for all experiments.

[2]http://www.opensubtitles.org

B. Results

1) Improving word alignment quality: Generally, integrating semantic information from word embeddings into the classic IBM models helps to gain significantly better results on word alignment. The report of AER for each model is presented in Table I and Table II. In these two tables, we denote IBM model x as x. Boosted IBM model x is denoted as **x***.

AER reduces considerably with the boosted models in both experimental corpora. We just improve the IBM model 1 and 2, and then seed the trained parameters to the higher models. The result for Wikipedia corpus is generally better than the result for OpenSubtitles corpus because in the subtitle domain, sentences are mostly picked up from conversations and spoken sentences are usually translated flexibly by the translators, in other words, they are rarely translated word-by-word. However, big improvements for word alignment can be obtained in both corpora. For each sub-corpus, the AER is decreased gradually. Our result for the boosted IMB model 4 is much better than the result for the classic IBM model 5.

Due to [18], the lexical translation parameter for EN-VN pair is quite less accurate because of the difference in grammars of this language pair. Hence, this result strongly proves that semantic information derived from word embeddings could be particularly valuable for word alignment models.

Table I: Word alignment evaluation (corpus: OpenSubtitles, metric: AER).

Model	Size of training corpus			
	10k	20k	50k	100k
123	0.4108	0.3925	0.3584	0.3241
1*23	0.3080	0.3063	0.2804	0.2568
1*2*3	0.3007	0.2975	0.2734	0.2467
1234	0.3936	0.3679	0.3436	0.3030
1*234	0.2957	0.2858	0.2641	0.2331
1*2*34	0.2880	0.2744	0.2548	0.2259
12345	0.3884	0.3638	0.3383	0.2982

Table II: Word alignment evaluation (corpus: Wikipedia, metric: AER).

Model	Size of training corpus			
	10k	20k	50k	100k
123	0.3147	0.2547	0.2356	0.1863
1*23	0.2518	0.2326	0.2170	0.1786
1*2*3	0.2445	0.2189	0.1960	0.1606
1234	0.3193	0.2399	0.2076	0.1606
1*234	0.2439	0.2185	0.1903	0.1478
1*2*34	0.2342	0.2061	0.1769	0.1387
12345	0.2702	0.2297	0.1958	0.1515

2) Improving translation quality: We evaluate the quality of translation on both directions, VN→EN and EN→VN. The BLEU scores of translation systems are presented in Table III. We can see that the phrase-based translation models also benefits from our boosted word alignment models, in both experimental corpora, again. This confirms our assumption that the lexical translation parameter adjusted by semantic information from word embeddings and the improved word alignments have a positive effect to translation quality. The enhanced word alignment model obviously helps the phrase-based translation model derive better phrase translation table, hence, it will choose more meaningful phrases in the decoding process.

Table III: Translation evaluation (metric: BLEU (%)).

Corpus	Direction	Baseline	Boosted	Delta
OpenSubtitles	VN→EN	21.92	22.27	0.35
	EN→VN	15.28	15.67	0.39
Wikipedia	VN→EN	22.89	23.12	0.33
	EN→VN	21.81	21.97	0.18

VI. Conclusion

In this paper, we have proposed a novel method to overcome the problems of IBM models that are lacking of parallel data and linguistic knowledge. First, we learn distributed representations of words for both languages on large monolingual corpora. Next, we obtain semantic similarity of words among two languages by using bilingual mapping among two learned semantic spaces. The similarity scores are then integrated into IBM models to fine-tune their parameters. The experiments employed in translation task between English and Vietnamese showed the effectiveness of our method as it helps to improve the word alignment and translation quality significantly.

Acknowledgements

This paper is supported by The Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.22.

References

[1] P. F. Brown, S. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, pp. 263–311, 1993.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[3] D. Chiang, "Hierarchical phrase-based translation," *computational linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[4] R. C. Moore, "Improving ibm word-alignment model 1," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 518.

[5] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 48–54.

[6] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[7] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[12] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," *EACL 2014*, p. 462, 2014.

[13] T.-P. Nguyen and A.-C. Le, "A hybrid approach to vietnamese word segmentation," in *Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE RIVF International Conference on*. IEEE, 2016, pp. 114–119.

[14] K. Heafield, "KenLM: faster and smaller language model queries," in *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, United Kingdom, July 2011, pp. 187–197. [Online]. Available: http://kheafield.com/professional/avenue/kenlm.pdf

[15] R. Skadiņš, J. Tiedemann, R. Rozis, and D. Deksne, "Billions of parallel words for free: Building and using the eu bookshop corpus," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014.

[16] C. Hoang, A. C. Le, P. T. Nguyen, and T. B. Ho, "Exploiting non-parallel corpora for statistical machine translation," in *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on*. IEEE, 2012, pp. 1–6.

[17] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 160–167.

[18] C. Hoang, C. A. Le, and S. B. Pham, "A systematic comparison between various statistical alignment models for statistical english-vietnamese phrase-based translation," in *Knowledge and Systems Engineering (KSE), 2012 Fourth International Conference on*. IEEE, 2012, pp. 143–150.