JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 20, No. 11, pp. 2211-2216, Nov. 2019

언어 모델의 워드 임베딩을 이용한 신경망 기계 번역

정 찬 웅 · 최 희 열*
한동대학교 정보통신공학과

Neural Machine Translation with Word Embedding Transferred from Language Model

Chanung Jeong · Heeyoul Choi*
Department of Information and Communication Engineering, Handong University, Pohang, Korea

[요 약]

신경망 번역이 기계 번역의 새로운 패러다임이 된 이후, 번역은 많은 양의 병렬 말뭉치에 의존하게 되었다. 번역과 달리 언어 모델은 하나의 언어로 된 말뭉치를 사용하는데, 데이터는 풍부하다. 언어모델의 말뭉치를 신경망 기계 번역에 이용하기 위한 몇가지 방법들이 제안되었다. 이 논문에서는 학습된 언어 모델을 신경망 기계 번역에 이용하는 새로운 방법을 제시한다. 번역의 입력언어와 출력언어에 해당되는 두 개의 언어 모델을 훈련시킨 뒤, 언어 모델들의 워드 임베딩 행렬들을 신경망 기계 번역으로 이전하는 방법이다. 제안된 방법을 검증하기위해 영어-독일어 그리고 영어-핀란드어의 데이터로 실험했고, 기존모델에 비해 40% 정도의 크기만 가지는 모델로도 성능이 같거나 조금 더 좋아지는 결과(+0.57 BLEU 점수)를 확인했다.

[Abstract]

Since neural machine translation (NMT) has become a new paradigm in machine translation, it relies on large amounts of parallel corpora to train neural networks, while language models (LMs) are trained on abundant monolingual corpus. Thus, there have been a few approaches to use monolingual corpus in training NMT systems. In this paper, we propose to use pretrained LM for NMT. After training two LMs for source and target languages of NMT, we transfer the word embedding matrices from the LMs to the target NMT model. In the experiments with the task of En-De and En-Fi translation, the proposed method keeps the translation quality the same or slightly better (up to +0.57 BLEU) using only around 40% of the previous model size.

색인어 : 지식 이전, 언어 모델, 기계번역, 인공신경망, 워드 임베딩

Key word : Knowledge Transfer, Language Model, Neural Machine Translation, Neural Network, Word Embedding

http://dx.doi.org/10.9728/dcs.2019.20.11.2211

Received 20 September 2019; **Revised** 22 October 2019
Accepted 05 November 2019

***Corresponding Author; Heeyoul Choi**

Tel: +82-54-260-1303
E-mail: hchoi@handong.edu

Ⅰ. Introduction

As a deep learning application [1], neural machine translation (NMT), which is an end-to-end approach to machine translation [2, 3, 4] has widely become adopted in machine translation research, as evidenced by its success in a recent WMT'16 translation task [5, 6, 7].

To train the end-to-end neural network model, NMT relies on large amounts of parallel corpus, and some language pairs suffer a lack of such corpus. Since monolingual corpus is abundant for any language, there have been several works to use monolingual corpus to help train NMT models in addition to parallel (bilingual) corpus [8, 9]. Contrary to NMT models, language models (LMs) are trained on monolingual corpus, by maximizing the likelihood of the next word, given the previous words in the sentence. Thus, the previous approaches to use monolingual data in NMT are based on combining LM and NMT. In those works, they focused on the structure of the translated sentences from NMT expecting that such structure can be captured by LM. Note that in those previous work, only one LM on the target language is combined to the NMT decoder.

In this paper, we focus on the word embedding rather than the sentence structure. The meaning of a word in the vocabulary can be learned by training in natural language processing tasks (e.g. language modeling or neural machine translation) and represented by a word embedding vector. In other words, training brings similar words close to each other in the word embedding space and dissimilar words far away from each other.

The rationale of our work is twofold. First, we notice that around 60% of NMT model parameters are used for word embedding. If we can avoid training the word embedding in NMT, our model size can be significantly decreased, which might imply that the NMT model needs a less amount of parallel data. Also, we can detour the rare words problem [10, 11, 12, 7]. Second, we assume that word embedding in LM and NMT can be shared, because their representations show semantic meanings of words [3] and that the semantic meanings represented by word embedding vectors are not significantly different between LM and NMT. That is, instead of training the word embedding vectors in NMT, we want to just transfer the ones from LM to NMT.

After training two LMs for both source and target languages in NMT, we transfer the word embedding matrices from LMs tot he NMT model. In the experiments with the task of En-De and En-Fi translation, the proposed method keeps the translation quality the same or even slightly better (up to +0.57 BLEU for En-Fi) using only around 40% of the previous model size for NMT.

Ⅱ. Background: LM and NMT

In this section, we give a brief overview of LMs and NMT models focusing on word embedding. See [13] and [5] for details.

2-1 Language Model

LMs reflect the syntactic and semantic regularities of a given language for recognition or generation. Generally, LMs calculate a probability of a word sequence to measure how likely the sequence(or a sentence) is. Given a sentence $(w_1, w_2, \cdots, w_T)$, the probability is calculated by factorized conditional probabilities as follows:

$$p(w_1, w_2, \cdots, w_T) = \prod_{t=1}^{T} p(w_t | w_1^{t-1}), \tag{1}$$

where w_t is the t-th word, and w_1^{t-1} stands for a word sequence $(w_1, \cdots, w_{t-1})$

A neural network with one-hot vectors as its input can learn the word embedding vectors [14]. In neural network based language modeling (NNLM), the conditional probabilities are implemented by forward propagation in neural networks, and the first NNLM was based on feed forward neural network [15], which generalizes to unseen or rare n-grams. To overcome the limitation of the Markov assumption in feed forward neural network LMs, RNNLM has been proposed [13]. The ordinary RNN computes the output sequence of hidden node from an input sequence, which is obtained by

$$\boldsymbol{h}_t = \Phi(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t), \tag{2}$$

where Φ can be implemented as long short-term memory (LSTM [16]), gated recurrent units (GRU[17]) or any other RNN units [18], and x_t is the word embedding of w_t.

$$\boldsymbol{x}_t = \boldsymbol{E}1(w_t), \tag{3}$$

where $1\,(w_t)_j$ is a one-hot vector defined as

$$\mathbf{1}\,(w_t)_j = \begin{cases} 1, & \text{if } j = w_t \\ 0, & otherwise \end{cases} \tag{4}$$

$E \in \mathbb{R}^{E \times |V|}$ is a word embedding matrix, where E and $|V|$ are the word embedding dimension and the vocabulary size, respectively.

Finally, the output of the neural networks is a probability distribution for the next word

$$p(w_t = i | w_{<t}) \propto \exp(\boldsymbol{W_i h_t} + b_i), \tag{5}$$

where $\boldsymbol{W_i}$ is the i-th row vector of the matrix, $\boldsymbol{W} \in \mathbb{R}^{|V| \times E}$ and b_i is bias.

The LM model is usually trained to maximize the log-probability of the correct prediction of the next word, given the previous words in the sequence using a large training parallel corpus. This is done by stochastic gradient de- scent. It has been observed that these unsupervised word embedding vectors can be used to greatly improve supervised natural language tasks [19, 20]

2-2 Neural Machine Translation

The attention-based NMT system computes a conditional distribution over translations given a source sentence $X = (w_1^x, w_2^x, \cdots, w_T^x)$:

$$p(w_t = i | w_{<t}) \propto \exp(W_i h_t + b_i), \tag{6}$$

This is done by a neural network that consists of an encoder and a decoder. The encoder is often implemented as a bidirectional recurrent neural net- work that reads the source sentence word-by-word. Before being read by the encoder, each source word $w_t^x \in V$ is projected onto a continuous vector space:

$$\boldsymbol{x_t} = \boldsymbol{E^x} 1(w_t^x), \tag{7}$$

where $\boldsymbol{E^x} \in \mathbb{R}^{E \times |V|}$ is a source word embedding matrix. The resulting sequence of the word embedding vectors is then read by the bidirectional encoder recurrent network which consists of forward and reverse recurrent networks.

The decoder consists of two sub-components–a recurrent network and the attention mechanism [21]. The recurrent network in the decoder is unidirectional, which computes the conditional distribution over the next target word, given all the previous target words and the source sentence:

$$p(w_t^y | w_{<t}^y, X) \tag{8}$$

The decoder recurrent network maintains an internal hidden state z_t. At each time step t', it first uses the attention mechanism to make a context vector, $c_{t'}$, from the annotation vectors that are the output of the encoder. The decoder recurrent network updates its own hidden state by

$$\boldsymbol{z_{t'}} = \Phi_z(\boldsymbol{z_{t'-1}}, \boldsymbol{y_{t'-1}}, \boldsymbol{c_{t'}}). \tag{9}$$

Like the encoder, Φ_z can be implemented as either an LSTM or GRU. $y_{t'-1}$ is a target-side word embedding vector computed by

$$\boldsymbol{y_{t'-1}} = \boldsymbol{E^y} 1(w_{t'-1}^y), \tag{10}$$

similarly to Eq. (7). The probability of each word i in the target vocabulary V' is computed by

$$p(w_t^y = i | w_{<t'}^y, X) \propto \exp(\boldsymbol{W_i z_{t'}} + c_i). \tag{11}$$

As in LM, the NMT model is usually trained to maximize the log-probability of the correct translation, given a source sentence, which is done by stochastic gradient descent.

Ⅲ. Word Embedding Transfer

The main contribution of this paper is to transfer word embedding from LM to NMT. In the two LMs for source and target languages in NMT, we obtain two word embeddings.

To transfer the word embedding from LM to NMT, we have to match two factors: vocabularies and embedding dimensions. We build up a dictionary for each language based on the monolingual corpus, expecting the monolingual data to include more accurate and rich data regularities. Also, the dimension of the word embedding for LM and NMT should match each other (620 in our experiments).

In addition, we slightly change the baseline network architecture to maximize the effect of word embedding transfer. In the baseline architecture for NMT, there are actually three word embedding matrices: one for source and two for target words. Right before the softmax, the feed-forward layer is a kind of transpose of the word embedding matrix for target words. Thus, instead of the feed-forward neural network layer, we can use the word embedding matrix for target words. That is, E^y in Eq. (10) and W^y in Eq. (11) can share the same parameter in NMT, where W^y are the transpose of E^y. By this weight sharing, the model size is decreased significantly (to 71% in NMT). We call this model Baseline+ProjY in our experiments, and Baseline+PorjY+LM$_{word}$ is our final model with word embedding transferred from LM.

Likewise, for LM, we use one word embedding matrix for the input and the output layers. After obtaining the word embedding of the input words, two LSTM layers follow, then the transposed word embedding matrix is applied to get the output right before the softmax. The output dimension of the last LSTM layer should be the same as the one of the word embedding vectors. That is, as in NMT, E in Eq. (3) and W in Eq. (5) share the same parameter.

Ⅳ. Experiment

4-1 Tasks and Corpora

We evaluated the proposed word embedding transfer on two translation tasks; (1) En-De and (2) En-Fi. As a baseline task, for each language pair, we used all the parallel corpora available from WMT'16 for training, which results in 4.5M and 2M sentence pairs for En-De and En-Fi, respectively. In the case of En-De, we preprocessed the parallel corpora following [22] and ended up with 100M words on the English side. For En-Fi, we did not use any preprocessing routine other than simple tokenization. Instead of space- separated tokens, we use 30k subwords extracted through byte pair encoding (BPE), as suggested in [11]. When computing the translation quality using BLEU, we *un-BPE'd* the resulting translations, but left them tokenized.

For word embedding transfer, we trained three LMs for three languages: En, De, and Fi. Note that we have one LM for En, although we transfer the same word embedding matrix to the two NMT models for En-De and En-Fi, respectively. We used monolingual data from the WMT'16 page as in Table 1. Since the monolingual data is not abundant for En-Fi, we used both monolingual data and parallel data for the corresponding LMs. Note that the corpus of Fi for LM is twice as large as the parallel corpus. For En-De, the parallel corpus was not included for LM training, since the monolingual data of De was large enough, compared to Fi.

표 1. 언어모델을 학습하기위한 학습 데이터
Table 1. Training datasets to train three LMs

	En	De	Fi
Parallel	En-Fi data	-	En-Fi data
Mono	Europal v7/v8 News Crawl(2015)	Europal v7/v8 News Crawl(2015)	Europal v7/v8 News Crawl(2014) News Crawl(2015)

표 2. En-De 와 En-Fi 에 대한 BLEU 점수. Validation 데이터에 대한 결과는 괄호안에 있음. NMT 의 기본 모델 [4] 에 BPE 사용함.
Table 2. BLEU scores on the test sets for En-De and En-Fi with two different beam widths. The scores on the development sets are in the parentheses. The baseline in the NMT model in [4] with LSTM and BPE

	En-De		En-Fi	
Beam Width	1	12	1	12
Baseline	19.15(18.82)	21.41(20.60)	7.38(8.02)	8.91(9.20)
+ProjY	19.21(18.78)	21.35(20.49)	7.73(8.39)	8.81(9.33)
+ProjY+LM$_{word}$	18.96(18.78)	**21.60**(20.26)	8.12(8.41)	**9.48**(9.71)

4-2 Decoding and Evaluation

Once a model is trained, we use a simple forward beam search with the width set to 12 to find a translation that approximately maximizes $\log p(Y|X)$. The decoded translation is then *un-BPE'd* and evaluated against a reference sentence by BLEU (in practice, BLEU is computed over a set of sentences). We use 'newstest2013' and 'newstest2015' as the validation and test sets for En-De, and 'newsdev2015' and 'newstest2015' for En-Fi.

4-3 Results

We present the translation qualities of all the models on both En-De and En-Fi in Table 2. For En-De, the differences are marginal. However, for En-Fi, whose corpus is much smaller than En-De, the table shows significant improvement in the translation qualities.

Also, the model size to train is 38% compared to the baseline model, which means training per each iteration is faster and converges faster than the baseline model. Note that the data set for NMT of De is completely different from the LM data. Yet, the proposed method has almost the same performance with a much smaller model size.

Ⅴ. Conclusion

In this paper, we proposed to transfer word embedding from previously trained language models to neural machine translation. The experiment results show that our proposed method works efficiently and even improves the qualities of translations with a much smaller model size. This implies that NMT models can be trained more easily even when the corpus size is limited.

For future studies, we can train LMs with much larger corpus (like *Wikipedia*) and transfer the word embeddings to NMT, and even fine-tune the word embedding in NMT models. Also, it will be interesting to compare the contributions of source and target word embedding separately.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2017R1D1A1B03033341), and by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00749, Development of virtual network management technology based on artificial intelligence).

References

[1] H. Choi, Y. Min, “Intelligent Information System; Introduction to deep learning and major issues”, *Korea Information Processing Society Review*, vol. 22, no. 1, pp. 7-21, 2015.

[2] N. Kalchbrenner, P. Blunsom, “Recurrent continuous translation models”., EMNLP, Seattle, Washington. pp. 413, 2013.

[3] I. Sutskever, O. Vinyals, Q. V. Le, “Sequence to Sequence Learning with Neural Networks”, in: Advances in Neural Information Processing Systems (NIPS), Montreal, Canada. 2014.

[4] D. Bahdanau, K. Cho, Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate”, in: Proc. Int’l Conf. on Learning Representations (ICLR), Sandiego, CA. 2015.

[5] R. Sennrich, B. Haddow, A. Birch, Edinburgh “neural machine translation systems for wmt 16”, in: The First Conference on Statistical Machine Translation (WMT), Berlin, Germany. 2016.

[6] J. Chung, K. Cho, Y. Bengio, “The NYU-MILA neural machine translation systems for wmt’16”, in: The First Conference on Statistical Machine Translation (WMT), Berlin, Germany. 2016.

[7] C. Kang, Y. Ro, J. Kim, and H. Choi, “Symbolizing Numbers to Improve Neural Machine Translation,” *Journal of Digital Contents Society*, vol. 19, no. 6, pp. 1161-1167, 2018.

[8] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, Y. Bengio, “On using monolingual corpora in neural machine translation”, arXiv preprint arXiv:1503.03535.

[9] R. Sennrich, B. Haddow, A. Birch, “Improving neural machine translation models with monolingual data”, arXiv preprint arXiv:1511.06709.

[10] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, W. Zaremba, “Addressing the Rare Word Problem in Neural Machine Translation”, arXiv preprint arXiv:1410.8206

[11] R. Sennrich, B. Haddow, A. Birch, “Neural machine translation of rare words with subword units”, arXiv preprint arXiv:1508.07909

[12] H. Choi, K. Cho, Y. Bengio, “Context-dependent word representation for neural machine translation”, *Computer Speech and Language,* vol. 45, pp. 149–160, 2017.

[13] T. Mikolov, G. Corrado, K. Chen, J. Dean, “Efficient Estimation of Word Representations in Vector Space”, in: Proc. Int’l Conf. on Learning Representations (ICLR), Scottsdale, Arizona. 2013.

[14] R. Miikkulainen, M. G. Dyer, “Natural language processing with modular neural networks and distributed lexicon”, *Cognitive Science,* vol. 15, pp. 343–399, 1991.

[15] Y. Bengio, R. Ducharme, P. Vincent, “A Neural Probabilistic Language Model”, *The Journal of Machine Learning Research,* vol. 3, pp. 1137–1155, 2003

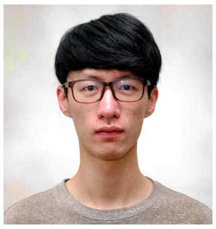
[16] S. Hochreiter, J. Schmidhuber, “Long short-term memory”, *Neural computation* vol. 9, no. 8, pp. 1735 –80. doi:10.1162/neco.1997.9.8.1735, 1997.

[17] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, “On the properties of neural machine translation: Encoder decoder approaches”, arXiv preprint arXiv:1409.1259.

[18] H. Choi, “Persistent hidden states and nonlinear transformation for long short-term memory”, *Neurocomputing,* vol. 331, pp. 458-464, 2019.

[19] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, “Natural language processing (almost) from scratch”, *The Journal of Machine*

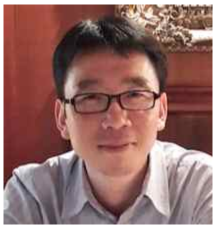
Learning Research, vol. 12, pp. 2493-2537, 2011.
[20] J. Turian, L. Ratinov, Y. Bengio, “Word representations: a simple and general method for semi-supervised learning”, in: Proceedings of the 48th annual meeting of the Association for Computational Linguistics, pp. 384-394, 2010.
[21] H. Choi, K. Cho, Y. Bengio, “Fine-grained attention mechanism for neural machine translation”, *Neurocomputing,* vol. 284, pp.171-176, 2018.
[22] S. Jean, K. Cho, R. Memisevic, Y. Bengio, “On Using Very Large Target Vocabulary for Neural Machine Translation”, in: 53rd Annual Meeting of the Association for Computational Linguistics, Beijing, China. 2015.



정찬웅(Chanung Jeong)

2016년 : 한동대학교 컴퓨터공학부

2016년~2018년: Refresh Foods Inc(San Jose, CA. USA)
2018년~현 재: 한동대학교 일반대학원 정보통신공학과 Machine Intelligence Lab 석사 과정
※관심분야 : 딥러닝(Deep Learning), NLP(Natural Language Processing), Chat-bot 등



최희열(Heeyoul Choi)

2005년: 포항공과대학교, 컴퓨터공학과 (이학석사)
2010년: Dept. of Computer Science and Engineering, Texas A&M University (Ph.D)
2010년 ~ 2011년: Indiana University (PostDoc)
2015년 ~ 2016년: University of Montreal (Visiting Researcher)

1998년 ~ 2001년: OromInfo (Programmer)
2011년~2016년: 삼성전자 종합기술원 (Research Staff Member)
2016년~현 재 : 한동대학교 전산전자공학부 조교수
※ 관심분야 : 머신러닝, 딥러닝, 인공지능