Contrastive Word Embedding Learning for Neural Machine Translation

Anonymous ACL submission

Abstract

Seq2seq models have shined in the field of Neural Machine Translation (NMT). However, word embeddings learned by NMT models tend to degenerate and be distributed into a narrow cone, named *representation degeneration problem*, which limits the representation capacity of word embeddings. In this paper, we propose a Contrastive Word Embedding Learning (CWEL) method to address this problem. CWEL combines the ideas of contrastive representation learning with embedding regularization, and adaptively minimizes the cosine similarity of word embeddings on the target side according to their semantic similarity. Experiments on multiple translation benchmark datasets show that CWEL significantly improves translation qualities. Additional analysis shows that the improvements mainly come from the well-learned word embeddings.

1 Introduction

NMT models fall into the encoder-decoder framework and have attracted widespread attention in the academic community (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017). It's shown that word embeddings learned by NMT models tend to degenerate and be distributed into a narrow cone, named *representation degeneration problem* (Gao et al., 2019) , which limits the representation power of word embeddings and doesn't have enough capacity to model the diverse semantics in natural languages (McCann et al., 2017).

To address *representation degeneration problem*, Gao et al. (2019) proposed a novel regularization method to increase the representation power of word embeddings explicitly. It's widely shown that embeddings of syntactically and semantically similar words are close to each other (Mikolov et al., 2013a,b; Pennington et al., 2014). And Wieting et al. (2019) showed that cosine similarity of sentence embeddings can represent their semantic similarity to some extend. However, Gao et al. (2019) minimized the cosine similarity of each pair of words equally regardless of their intrinsic semantic relationship, which still limits the representation power of learned embeddings.

To address this, we borrow the idea of contrastive learning (van den Oord et al., 2019), and propose to minimize the cosine similarity of the words in the batch adaptively according to their semantic similarity. Several works in NMT also utilized contrastive learning: Bhat et al. (2019) optimized a margin-based loss on LSTM-based continuous-output NMT models to maximize densities of the pretrained target embeddings; Lee et al. (2021) proposed a contrastive learning framework on sentence-level representations to address the "exposure bias" problem. Differently, our method focuses on word-level contrastive representation learning of the target words on the state-of-the-art NMT models. The contrast among anchor, positive and negative samples motivates us to take the semantic similarity between word embeddings into consideration. Our method utilizes the angle between two word embeddings as a soft signal of positive or negative samples to control the degree of minimization. It means that the larger angle between learnt word embeddings indicates more dissimilar semantics and is related to higher weights, and vice versa. So the cosine similarity minimization of semantically dis-similar words should be assigned with higher weights compared to similar ones, thus making the embedding space more distinguishable and expressive.

Specifically, we first select a fixed-size bag-of-words without repetitions for each batch through a random sampling strategy. Then, we adaptively minimize the cosine similarity between each word in the samples and other words in the bag-of-words with the computed angles as weights. In this way, we hope that the cosine similarity of words with similar semantics can be larger than dis-similar

ones. As a result, when the model generates translations, it can avoid using semantically dis-similar words to generate incorrect translations, but use semantically similar words to generate correct translations.

Briefly, we propose a framework named Contrastive Word Embedding Learning (CWEL), which aims to adaptively minimize cosine similarity of word embeddings according to their semantic similarity. Experiments are conducted on three translation benchmarks: NIST Chinese⇒English (Zh⇒En), WMT'14 English⇒German (En⇒De) and WMT'14 English⇒French (En⇒Fr). The experimental results show that the proposed model outperforms strong baseline models significantly. Extensive analyses show that our method learns more distinguishable word embeddings with more expressiveness.

2 Method

2.1 Contrastive Word Embedding Learning

Bag-of-Words Sampling Given the batch $\mathbf{B}$ which contains $|\mathbf{B}|$ source sentences and $|\mathbf{B}|$ corresponding translations, which are indicated as $\mathbf{B}_x$ and $\mathbf{B}_y$, with $\mathbf{x}_i$ and $\mathbf{y}_i$ as the i-th source and target sentences in $\mathbf{B}_x$ and $\mathbf{B}_y$ respectively. The set of all words in $\mathbf{B}_y$ is represented as S_y which does not include duplicate words. We randomly sample a bag-of-words $\text{BOW} = \{b_1, \cdots, b_m, \cdots, b_{|\text{BOW}|}\}$ without repetitions from the set S_y. Note that b_m is the m^{th} word in BOW, with $|\text{BOW}|$ as its size. Due to the limitation of memory and efficiency, we only sample a subset of S_y, with the sampled size $|\text{BOW}|$ as a hyper-parameter. And it's always satisfied that $|\text{BOW}| \leq S_y$.

Note that the sampling here is reasonable. Assume that each batch is a sampling of the whole training set, whose distribution of words is almost the same as the whole training set. By sampling BOW without repetitions in each batch, the semantic relationship among the words in the dataset can be correctly modeled, which may include not only synonyms but also antonyms, etc. All of these words are informative and necessary for contrastive learning. So it's unnecessary to sample words using semantic labels.

Weighted Contrastive Loss Given a target word as w_o in a sample $(\mathbf{x}_i, \mathbf{y}_i)$, and a word from BOW as b_j, their word embeddings are denoted as E_{w_o} and E_{b_j} respectively. It's crucial to define the semantic relationship between words b_j and w_o. If the semantic of b_j is similar to w_o, we will set b_j as a positive sample of w_o in contrastive learning, and vice versa. Here we utilize the angle between two word embedding vectors as a soft signal of semantic relationship. As a result, it acts like a kind of weight[1], which is calculated as follows:

$$W_{o,j} = \arccos(E_{w_o}, E_{b_j}) \quad (1)$$

where $\arccos$ is a function to acquire the angle between two word embeddings. The weight is computed by our NMT model on the fly. Note that the method gives the largest weights to anti-parallel word embedding vectors (negative samples), and vice versa.

Our goal is to make word embeddings more distinguishable. The cosine similarity between word embeddings is computed as follows:

$$S_{o,j} = \cos(E_{w_o}, E_{b_j}) \quad (2)$$

The reason why we resort to "arccos" and "cos" is straightforward according to Wieting et al. (2019), who shows that cosine similarity between two learnt sentence embeddings can represent their semantic similarity. Then, the weighted contrastive loss $\mathcal{L}^{\text{CL}}$ is computed as follows:

$$\mathcal{L}^{\text{CL}} = \sum_{o=1}^{N_o} \exp\left(\sum_{j=1}^{|\text{BOW}|} W_{o,j} * S_{o,j}\right) \quad (3)$$

where N_o denotes the total number of target words of all the samples in each batch.

This method has an intuitive explanation: we hope that the similarity minimization of embeddings with similar-semantics can be assigned smaller weights than dis-similar ones. As a result, the learnt embedding space can distinguish words with different semantics, but keep the high cosine similarity of semantically-similar words, which has much more expressiveness.

Note that our proposed method is significantly different from previous works of contrastive learning (He et al., 2020; Chen et al., 2020a,c,b; Khosla et al., 2020; Gunel et al., 2021). The advantages of our method are as follows: 1) soft signals of positive and negative samples; 2) without complicated data augmentation; and 3) without additional architectures. Note again that compared to Gao et al. (2019), our method considers the semantic relationship between words during the minimization.

[1] Similar ideas can also be implemented by other methods of weighting, including $(1 - \cos(E_{w_o}, E_{b_j}))/2$ and $\exp(-\cos(E_{w_o}, E_{b_j}) - 1)$. But we found that their performance gains are slightly worse.

Model	NIST Zh⇒En						WMT'14 En⇒Fr
	MT02	MT03	MT04	MT05	MT08	Avg.	
Transformer-base	46.13	44.79	45.59	44.54	34.79	42.64	41.99
+ CWEL	**47.27**†	**45.58**†	**46.87**†	**45.63**†	**35.61**†	**44.19**	**42.20**
Transformer-big	47.64	46.50	46.85	46.70	37.37	44.69	43.56
+ CWEL	**48.17**†	**47.47**†	**47.76**†	**47.87**†	**37.62**	**45.58**	**43.87**

Table 1: Case-insensitive BLEU scores (%) on NIST Zh⇒En and case-sensitive BLEU scores (%) on WMT'14 En⇒Fr translation tasks. "†" indicates statistically significant difference from Transformer ($p < 0.01$). The bold results denote the best ones among the proposed models and their corresponding inhouse baselines.

2.2 Integration into NMT

Our method can be applied to most NMT models. Without loss of generality, we take the Transformer as an example. Based on the conventional auto-regressive NMT training objective, we integrate the contrastive word embedding loss mentioned above as follows:

$$\mathcal{L} = (1-\lambda)\,\mathcal{L}^{\mathrm{CE}} + \lambda\mathcal{L}^{\mathrm{CL}} \quad (4)$$

$$\theta^{*} = \mathrm{argmin}\,(\mathcal{L};\theta) \quad (5)$$

where $\mathcal{L}^{\mathrm{CE}}$ denotes the naive Cross-Entropy loss, and λ is a hyper-parameter adopted to balance the two losses. We train our model by using the final loss from scratch and get model parameters θ^{*}. Note that we only apply $\mathcal{L}^{\mathrm{CL}}$ on the target side in this paper.

3 Experiments

We present experiments on NIST Chinese-English (Zh⇒En), WMT'14 English-German (En⇒De), and English-French (En⇒Fr) translation tasks.

3.1 Setup

Dataset For NIST Zh⇒En, the training dataset consists of 1.25M sentence pairs extracted from LDC corpora. We choose NIST 2006 (MT06) as the validation set, which has 1664 sentences, with NIST 2002 (MT02), NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05), and NIST 2008 (MT08) as test sets. For WMT'14 En⇒De and WMT'14 En⇒Fr, we perform experiments on the corpus provided by WMT'14, which contain 4.5M sentence pairs and 36M sentence pairs, respectively. newstest2013 and newstest2014 are used as validation and test sets. All statistical significance tests are conducted according to Collins et al. (2005).

Baselines We compare our proposed methods with the following baselines:

- **Transformer (Vaswani et al., 2017)** The state-of-the-art seq2seq model. We compare the results on both base and big models.
- **Yang et al. (2019)** A fine-tuning method for reducing word omission errors by contrastive learning at sentence level. We reproduce their methods on our Transformer baselines.
- **Wang et al. (2019)** An adversarial training mechanism for regularizing neural language models, which yields better generalization performance.
- **Gao et al. (2019)** A novel regularization method to address representation degeneration problem.

3.2 Results

NIST Zh⇒En and WMT'14 En⇒Fr As shown in Table 1, After the introduction of the Contrastive Word Embedding Loss (CWEL), compared with the baseline system, the performance on most test sets has been significantly improved. Particularly, the CWEL can improve the *Transformer-base* and *Transformer-big* model by about +1.5 and +0.9 BLEU points averagely on all test datasets. On the large-scale WMT'14 En⇒Fr dataset, our models surpass strong baselines by 0.21 and 0.31 BLEU scores respectively. Note that we compare other baselines on WMT'14 En⇒De dataset.

WMT'14 En⇒De As shown in Table 2, on the WMT'14 En⇒De translation task, the CWEL brings significant improvements by about +0.71 and +0.75 BLEU points compared to the *Transformer-base* and *Transformer-big* model respectively. The works of Wang et al. (2019) utilize additional architectures to do adversarial learning. However, equipped with CWEL, our *Transformer-base* model is comparable to the performance of Wang et al. (2019), slightly higher than that of Gao et al. (2019) and significantly better than Yang et al. (2019). Similarly, CWEL also

Model	Base	Big
Wang et al. (2019)	**28.43**	—
Yang et al. (2019)	27.87	28.66
Gao et al. (2019)	28.38	28.94
Transformer	27.71	28.79
+ CWEL	28.42‡	**29.54**†

Table 2: Case-sensitive BLEU (%) on WMT'14 En⇒De translation task. "‡" and "†" indicate statistically significant difference with $p < 0.05$ and $p < 0.01$ from Transformer respectively. The bold results denote the best ones among the proposed models and their corresponding inhouse baselines.

makes our *Transformer-big* model significantly better than that of Yang et al. (2019) and Gao et al. (2019). This more detailed comparison among inhouse baselines and related works indicates that the word embeddings learned by CWEL really help to improve the translation performance.

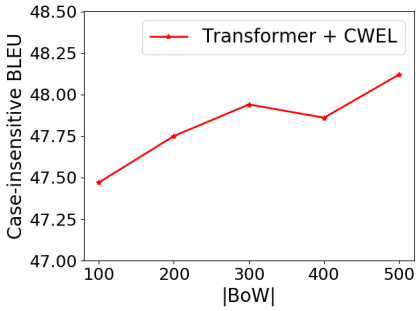


Figure 1: Performance on the validation set of NIST Zh⇒En dataset with different sizes of bag-of-words.

4 Analysis

4.1 Size of Bag-of-Words

In this section, we explore the effects of different sizes of bag-of-words on translation performance. According to Figure 1, it's obvious that a larger size of bag-of-words brings much more gains of translation performance, which is similar to larger batch size in previous works of contrastive learning (Chen et al., 2020a). Due to the limitation of our computation resource, we did not train for a larger size of bag-of-words than 500.

4.2 Expressiveness of Embeddings

In order to confirm that the improvements in translation performance are indeed due to our learning of embeddings, we access the expressiveness of embeddings by the commonly-used singular value decomposition (Gao et al., 2019; Liu et al., 2021).

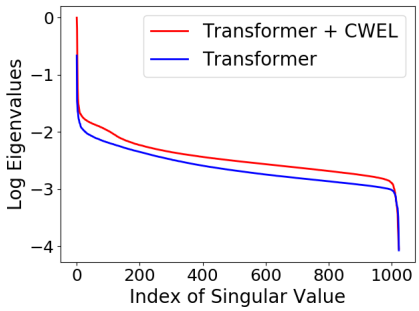
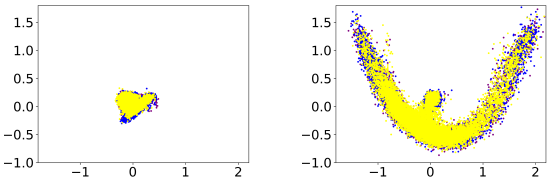


Figure 2: Singular value of embedding matrix. The models are trained on NIST Zh⇒En dataset.



(a) Embedding projection of standard Transformer. (b) Embedding projection of Transformer with CWEL.

Figure 3: Embedding visualization of standard Transformer trained with/without CWEL on NIST Zh⇒En dataset. Blue: embeddings in the source side. Yellow: embeddings in the target side. Purple: shared embeddings.

The higher singular values indicate that the embeddings are more uniformly distributed and have more expressiveness. From Figure 2, it's obvious that the model trained with CWEL gets word embeddings with much higher singular values, thus has more expressiveness.

4.3 Visualization of Embeddings

In order to further explore the representation of words learned by the CWEL-assisted NMT model, we visualize embeddings by commonly-used principal component analysis (PCA) to reduce embedding from 1024 to 2 dimensions for intuitive display in 2-dimensional space. According to Figure 3(a), the embeddings learned in standard Transformer are distributed into a narrow cone. However, with CWEL, the decoder embeddings become more distinguishable, as shown in Figure 3(b).

5 Conclusion

We combine the ideas of contrastive learning and embedding regularization, and propose Contrastive word Embedding Learning (CWEL) to alleviate *representation degeneration problem*. Experiments on several machine translation benchmarks show the superiority of our method.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Gayatri Bhat, Sachin Kumar, and Yulia Tsvetkov. 2019. A margin-based loss with synthetic negative samples for continuous-output machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 199–205, Hong Kong. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020b. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 22276–22288. Curran Associates, Inc.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020c. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. 2021. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *International Conference on Learning Representations*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *CoRR*, abs/1708.00107.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words

with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565, Long Beach, California, USA. PMLR.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

Source-1	jīnzìtǎ sì de tǒngzhì zhìdù yě zàochéng le xiàjí guānyuán zhī néng wǎngshàng kàn , chùchù tīngmìngyú shàng yī jí .
Trans.Big	a pyramid administration system has also created a system where officials at lower levels can only look forward and obey orders from higher levels.
CWEL	the pyramid - like ruling system has also caused lower - level officials to be able to look up and listen to orders from higher levels everywhere.
Source-2	zài nóngcūn xiǎng gǎo diǎn wénhuà huódòng , zhǎo diǎn " lè " zǐ tài nán le
Trans.Big	it is too difficult to develop some cultural activities in rural areas and find some "music."
CWEL	it is too difficult to find some "fun" if we want to engage in some cultural activities in the rural areas.
Source-3	ér zài zhè fāng miàn , jiāngxī de zuòfǎ duì nóngmín lái shuō wúyí shì fúyīn.
Trans.Big	in this respect, jiangxi's practice is no doubt good for farmers.
CWEL	in this regard, jiangxi's practice is no doubt a blessing to farmers.

Table 3: Translation examples on validation set of NIST Zh⇒En dataset. **Trans.Big** represents *Transformer-big* model. **CWEL** represents our proposed method.

A Appendix

We elaborate from three aspects.

A.1 Case Study

Here we give some examples translated by baseline and our model respectively on NIST Zh⇒En dataset. From Table 3, we can see that Transformer with CWEL correctly translate chunks [jīnzìtǎ sì de tǒngzhì zhìdù], [" lè " zì] as well as [fúyīn], compared to baseline. As mentioned before, we hope that models can utilize semantic-similar words to generate translations. Although chunks [pyramid - like ruling system] and [find some "fun"] are unseen in the training set, our proposed model successfully generates them, showing the superiority of learned word embeddings.

A.2 Other Methods of Weighting

As shown in Table 4, our proposed method of weighting slightly outperforms other methods with similar ideas on the validation set of WMT'14 En⇒De translation task. So we choose to use the angle between word embeddings as weight.

Methods of Weighting	**Big**
$W_{o,j} = \exp(-\cos(E_{w_o}, E_{b_j}) - 1)$	29.17
$W_{o,j} = (1 - \cos(E_{w_o}, E_{b_j}))/2$	29.24
$W_{o,j} = \arccos(E_{w_o}, E_{b_j})$	**29.34**

Table 4: Case-sensitive BLEU (%) on WMT'14 En⇒De translation task. The bold results denote the best method of weighting.

A.3 Implementation Details

For the implementation of Transformer, we use the code provided by fairseq[2] (Ott et al., 2019).The hyper-parameter λ is set as 0.8. The size of bag-of-words is set as 500. The batch size is set as 12288 per GPU on all the experiments. The learning rate is set as 7e-4 and 5e-4 for base and big models respectively, which is controlled by Adam optimizer (Kingma and Ba, 2014). To acquire strong baselines, dropout (Srivastava et al., 2014) is used and set as 0.1 for all the models. We use byte-pair encodings (BPE) (Sennrich et al., 2016), with $32k$ and $37k$ for NIST and WMT dataset respectively. Experiments on NIST dataset run by 4 P40 GPUs and 4 V100 GPUs on WMT dataset, with gradient accumulation as 2. On NIST Zh⇒En dataset, we run 24,000 steps for each model and save the model every two epochs, which takes 6.6 hours for a base model. On WMT'14 En⇒De dataset, we run 100,000 steps for each model and save the model every 5,000 steps, which takes 9.2 hours for a base model. On WMT'14 En⇒Fr dataset, we run 150,000 steps for each model and save the model every 10,000 steps, which takes nearly 27.6 hours for a base model. As a result, we get base models with about 66M parameters and big models with 220M parameters approximately. For hyper-parameters selection on validation sets, we try 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8 for λ, with 100, 200, 300, 400 and 500 for $|\text{BOW}|$. Other settings are the same as default settings in fairseq.

[2]https://github.com/pytorch/fairseq