SYNTACTIC RELEVANCE XLNET WORD EMBEDDING GENERATION IN LOW-RESOURCE MACHINE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic understanding is an important factor affecting the quality of machine translation of low resource agglutination language. The common methods (subword modeling, pre-training word embedding, etc.) will increase the length of the sequence, which leads to a surge in computation. At the same time, the pre-training word embedding with rich context is also the precondition to improve the semantic understanding. Although BERT uses masked language model to generate dynamic embedding in parallel, however, the fine-tuning strategy without mask will make it inconsistent with the training data, which produced human error. Therefore, we proposed a word embedding generation method based on improved XLNet, it corrects the defects of the BERT model, and improves the sampling redundancy issues in traditional XLNet. Experiments are carried out on CCMT2019 Mongolian-Chinese, Uyghur-Chinese and Tibetan-Chinese tasks, the results show that the generalization ability and BLEU scores of our method are improved compared with the baseline, which fully verifies the effectiveness of the method.

1 INTRODUCTION

The neural machine translation based on end-to-end structure Bahdanau et al. (2015) Luong et al. (2015) Gehring et al. (2017) Vaswani et al. (2017)encodes the source language X into a set of continuous dense latent vectors Z, and decodes the target language Y from Z in the same way. Compared with the traditional statistical machine translation model Koehn (2010), the neural machine translation model is a combination of language model, translation model, word alignment model and word order model. The model updates the parameters by minimizing the cross entropy between the generated token and the ground-truth. During the training phase, whether the model gradient can be stabilized or not depends on the understanding degree of the model to the language knowledge, the setting of the structure and the quality of the modeling unit. Among them, the setting of model structure parameters determines the quality of language learning, and a robust modeling unit is the basis of the above process, therefore, how to obtain the word vector with rich semantic information is the key factor for the model to produce high-quality translation. The general one-hot word vector is expressed in terms of dictionary size, with the position of the word in the dictionary set to 1 and the other positions set to 0, although this method is simple and easy to express, the dimension is too large to learn the relationship between words. The Word2Vec Church (2017) and Glove Pennington et al. (2014) models embed high-dimensional one-hot word vector into a low-dimensional space using a distributed representation algorithm, similar to the N-gram language model Chen et al. (2019), computing the cosine distance from the origin to the vector point in the low-dimensional space in the initial neighborhood to learn the relationship between words, which has been widely used as the main neural machine translation modeling unit. Although this vector representation method has many advantages, it is still difficult to distinguish polysemous words and can not be optimized dynamically for specific tasks.

Recently, the BERT model Devlin et al. (2019) has been widely used in various natural language processing tasks, including machine translation, because of its strong ability of semantic feature learning. It consists of two main models: Masked Language Model(MLM) Salazar et al. (2020) and the Next Sentence Prediction(NSP) Liu et al. (2019). The MLM model based on the self-

encoding method uses the flag bit $MASK$ to randomly replace some words, and then use the context corresponding to the $MASK$ to predict the true value of the flag bit. However, $MASK$ is not used in the fine-tuning phase, which leads to the inconsistency between pre-training data and fine-tuning data. While XLNet Yang et al. (2019), which is trained by autoregressive model, solves the inherent problem of self-encoding, and overcomes the defects of the traditional autoregressive model which only uses the previous or post information. The main objective of XLNet is to maximize the expected logarithmic likelihood of the factorization order of all possible sequences and to obtain all possible contexts of sequences in autoregressive mode. Based on the advantages of XLNet, this paper presents a neural machine translation method for agglutination language. It mainly includes the following contents:

1. In this paper, the model is divided into upstream pre-training word embedding and downstream machine translation model.

2. For the upstream task, this paper adopts the word embedding pre-training method of semantic XLNet, uses the permutation language model to sort all possible sequences of the original sequence, and then learns the prediction probability of the token through the autoregression method, which not only retains the context information of the sequence, but also avoids using the $MASK$ flag bit.

3. The two-stream self-attention model is proposed to solve the problem that the permutation language model can not clearly express the content of the text.

4. Introduce fragment loop mechanism Dai et al. (2019). In order to reduce the list size, we split the agglutinative language into stem and affix, which, however, leads to an increase in sequence length. Therefore, we divide the long sequence into several fragments and cache them, store the information of the previously generated fragments into the cache and concatenate it with the subsequent fragments, which not only preserves the original length dependence, but also speeds up the training.

5. For the downstream tasks, the new word vectors obtained by splicing are used as the input of the downstream machine translation encoder, and are trained using the maximum likelihood estimation method to obtain the final model.

2 BACKGROUND

The neural machine translation model can simulate the translation probability $P(y|x)$ of the source language $x = (x_1, ..., x_n)$ to the target language $y = (y_1, ...y_m)$ word by word, as shown in Equation 1.

$$P(y|x) = \prod_{i=1}^{I} P(y_i|y_{<i}, x, \theta) \tag{1}$$

Where $y_{<I}$ indicates the partial translation result before the i-th decoding step, and θ indicates the parameters of the NMT model. Here, the probability calculation for generating the i-th target word is similar to:

$$P(y_i|y_{<i,x,\theta}) \propto exp\left\{f(y_{i-1}, s_i, c_i; \theta)\right\} \tag{2}$$

Where s_i indicates the i-th hidden state of the decoder, c_i indicates the corresponding context information of the source language at time t, and $f(\cdot)$ indicates the nonlinear activation function in the current node of the decoder. Calculate the source language context information according to Equation 3.

$$\begin{aligned} c_i &= \sum_{j=1}^{J} \alpha_{i,j} \cdot h_j \\ \alpha_{i,j} &= \frac{exp(e_{i,j})}{\sum_{j'=1}^{J} exp(e_{i,j'})} \end{aligned} \tag{3}$$

Where h_j indicates the encoder's comment on the input x_j, $\alpha_{i,j}$ indicates its weight, and the weight is calculated according to the attention model $attention(\cdot)$, and is determined by the attention score $e_{i,j}$. The attention score is calculated as follows:

$$e_{i,j} = attention(s_{i-1}, h_j) \tag{4}$$

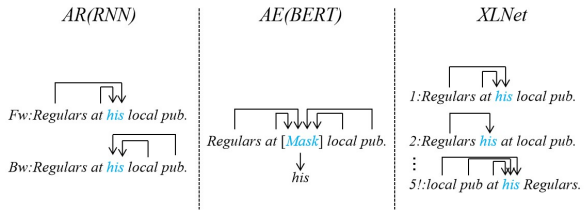

Figure 1: Permutation language model. It is the combination of Autoregressive (AR) and Autoencoding (AE) language model, which flexibly learn the context of current token.

The neural machine translation model uses the maximum likelihood estimation method to train the model parameter θ. For bilingual parallel sentence pair $\{(x^n, y^n)\}_{n=1}^{N}$ in the training set, the loss function during the training process is defined as follows:

$$Loss(\theta) = \underset{\theta}{argmax} \sum_{n=1}^{N} logP(y^n|x^n;\theta) \tag{5}$$

Although the logarithmic likelihood method is widely used, the exposure bias still exists, which directly affects the quality of the traditional neural machine translation model.

3 METHOD

The pre-training word embedding generation based on XLNet includes three aspects: Permutation language model, syntactic relevance two-stream self-attention, the fragment loop mechanism. In addition, we also introduce the downstream translation model.

3.1 PERMUTATION LANGUAGE MODEL

The permutation language model (PLM) overcomes the problems of pre-training and fine-tuning data disunity of auto-encoding language model and that autoregressive language model can only use unidirectional context, as shown in Figure 1. The specific approach is to factorize the input sequence. For a sequence of n tokens, $n!$ kinds of permutations will eventually be obtained. Note that PLM needs to retain the original position information of each word, that is, PLM is just a factorization of language modeling, not a rearrangement of position information.

Take "where to eat" as an example, when its factorization order is $2 \rightarrow 1 \rightarrow 3$, the corresponding language model is $p(where|to)p(to)p(eat|where, to)$. Essentially, the position of the original sequence is not changed. While $p(to)p(where|to)p(eat|where, to)$ is not allowed because it changes the position of each token.

If we use $n!$ kinds of decomposition methods, and the model parameters are shared, PLM can learn bidirectional context, but the calculation of traversing $n!$ kinds of paths is very large. Therefore, in reality, only partial arrangements of $n!$ kinds of paths are randomly sampled, and expectation:

$$\underset{\theta}{max} E_{z\sim Z_t}\left[\sum_{t=1}^{T} logp_\theta\left(x_{z_t}|x_{Z_{<t}}\right)\right] \tag{6}$$

In addition, one of the most important problems of permutation language model is that no matter where the predicted target position is, all cases obtained after factorization are the same, and the weights of transformer are the same for different situations, so no matter how the target position changes, the same distribution results can be obtained. The factorization results of "to" and "eat" in the above example are consistent.

However, when the standard Transformer is used to modeling PLM, there is a problem of unclear target position information perception. That is, the model does not know which position words are actually being predicted due to the sequence being scrambled, as a result, the probability of PLM

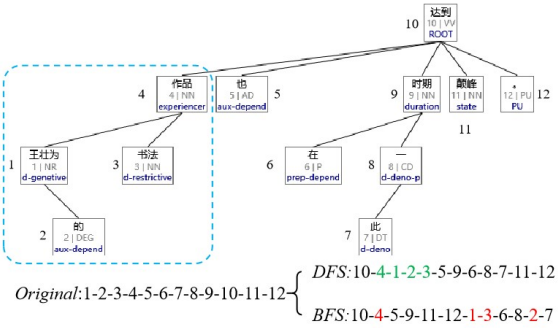

Figure 2: Syntactic relevance factorization sampling.

predicting different target words is the same under partial permutation, as shown in equation 7.

$$p_\theta\left(X_i = x|x_{z_{<t}}\right) = p_\theta\left(X_j = x|x_{z_{<t}}\right) = \frac{exp\left(e\left(x\right)^T h\left(x_{z_{<t}}\right)\right)}{\sum_{x^j} exp\left(e\left(x'\right)^T h\left(x_{z_{<t}}\right)\right)} \tag{7}$$

Therefore, aiming at the above two problems: Factorization sampling and position perception. We proposed a syntactic relevance two-stream self-attention model, which samples the factorization sequences with syntactic dependence from all alternative paths, and then uses the two-stream self-attention model to predict current target word and other words respectively. Then a new word embedding representation is generated.

3.2 SYNTACTIC RELEVANCE TWO-STREAM SELF-ATTENTION

Syntactic relevance factorization sampling Syntactic information clearly expresses the dependency of the core words (verbs, nouns) and its context. Compared with the linear modeling method, the modeling method based on syntactic distance has better extensibility and generalization.

Given a dependency tree T, we consider that there is a dependency relationship between the child node C_i and the parents node P_i, so we can obtain the factorization order by two methods: Depth first search (DFS) and Breadth first search (BFS). As shown in figure 2. The factorization order sampled by DFS can learn the subsequence information (chunk information) with the current node as the root word, while the factorization based on BFS can find the context information with syntactic relevance with the current node, that is, the attribute information of the current token. According to the blue box in figure 3, when the factorization order is $4 \rightarrow 1 \rightarrow 2 \rightarrow 3$, we get some chunk information (Wang's calligraphy works), while the factorization order is $4 \rightarrow 1 \rightarrow 3 \rightarrow 2$, we can learn the attribute information with "works" as the root node, including (whose works (Wang) and what works (calligraphy), etc.).

Two-stream self-attention We introduced a new distributed calculation method to improve the softmax output of the original Transformer model, to achieve the perception of the target position, as shown in equation 8.

$$p_\theta\left(X_{z_t} = x|x_{z_{<t}}\right) = \frac{exp\left(e\left(x\right)^T g_\theta\left(x_{z_{<t}}, z_t\right)\right)}{\sum_{x'} exp\left(e\left(x'\right)^T g_\theta\left(x_{z_{<t}}, z_t\right)\right)} \tag{8}$$

Where $g(\cdot)$ indicates a new representation, and the position information z_t is used as its input. It can be seen from equation 8 that when predict x_{z_t}, $g(\cdot)$ can only have its position information z_t and cannot contain content information x_{z_t}. When predicting other tokens $x_{z(j,j>t)}$, then the content information of x_{z_t} should be included so that there is complete contextual information. Since the traditional Transformer does not meet such requirements, the two-stream representation method is used here to replace the original representation. Two-stream refers to the query stream representation (QR) and the content stream representation (CR), respectively.

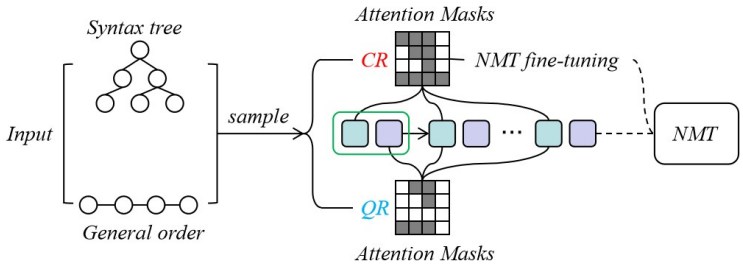

Figure 3: Two-stream self-attention mechanism. The attention mask is used to obtained the factorization, and then the context features are learned by autoregressive model, the blue box represent the content and the purple box indicates the position.

(1) QR: Mainly used to predict the current word, only contains location information, not content information of the word.

$$g_{z_t}^{(m)} \leftarrow Attention\left(Q = g_{z_t}^{(m-1)}, KV = h_{z_{<t}}^{(m-1)}; \theta\right) \tag{9}$$

The representation contains the context content information, $x_{z_{<t}}$, and the target location information z_t, but not the target content information, x_{z_t}.

(2) CR: Mainly provide QR with location information of other words, including location information and content information.

$$h_{z_t}^{(m)} \leftarrow Attention\left(Q = h_{z_t}^{(m-1)}, KV = h_{z_{<t}}^{(m-1)}; \theta\right) \tag{10}$$

This representation method is the same as the traditional transformer, and encodes both the context and x_{z_t} itself. The two-stream self-attention method can be summarized as the following process:

- The query stream (QR) in the first layer randomly initializes a vector $g_i = w$.
- The content stream (CR) uses the word vector $h_i = e(x_i)$.
- It should be noted here that the network weights of the two-streams are shared. Finally, QR is removed in the fine-tuning stage, and only the CR is used.

The model framework is shown in Figure 3.

3.3 Fragment loop mechanism

As mentioned earlier, agglutinative language needs to segment stem affixes to alleviate data sparseness, which leads to an increase in sequence length. At present, the maximum input length of BERT is 512. The traditional method for segmenting sequences has the problem of context fragmentation, that is, it can not model the semantic information of continuous documents, and the model reasoning needs repeated computation, so the reasoning is slow. Therefore, XLNet uses a fragment loop mechanism, the specific process is as follows:

- The text representation corresponding to the previous segment obtained by calculation will be repaired and cached, which will be used as an extended context summary when the model processes the next new segment. We only need to add the cache value of the previous segment in the calculation of KV, as follows:

$$h_{z_t}^{(m)} \leftarrow Attention\left(Q = h_{z_t}^{(m-1)}, KV = \left[\tilde{h}^{(m-1)}, h_{z_{<t}}^{(m-1)}\right]; \theta\right) \tag{11}$$

- The maximum dependency length has been increased by a factor of N, where n is the network depth. Since the new fragment in this paper only uses the cache value of the old fragment text to maintain the length dependence, so many repeated calculations are omitted and the training speed is accelerated.

3.4 DOWNSTREAM MODULE

For the downstream translation model, we mainly observed the translation effect of the word embedding combined with deep semantic knowledge on Transformer and RNNSearch. The model structure adopts the default configuration.

Through the above work, we obtained the hidden representation of word vector with semantic information h. In order to better fit the end-to-end translation model, we use h as the input sequence of the translation model encoder:

$$h^{mtencoder} = Encoder(h_1, ..., h_{maxlength}) \tag{12}$$

Where $Encoder(\cdot)$ indicates the translation model (Transformer or RNN) encoder. Meanwhile, we employ the decoder to predict the target sequence y_j from the final hidden state of the encoder $h^{mtencoder}$:

$$y_j = Decoder(h^{mtencoder}_{maxlength}) \tag{13}$$

Therefore, the calculation of translation probability is similar to the Eq.1, and then we still adopts the maximum likelihood estimation method to calculate the cross entropy of the model prediction output and reference:

$$P(y|x) = \prod_{j=1}^{J} P(y_j|y_{<j}, h^{mtencoder}, \theta) \tag{14}$$

$$Loss(\theta) = \underset{\theta}{argmax} \sum_{n=1}^{N} logP(y^n|h^n_{mtencoder}; \theta) \tag{15}$$

4 EXPERIMENTS

Dataset and Setting For Mo-Zh task, the training set consist of 260K bilingual sentences from CCMT2019[1] which including 2K sentence pairs for test set and validation set. For Ug-Zh task, the training set consist of 300K bilingual sentences from CCMT2019 which including 2K sentence pairs for test set and validation set. For Ti-Zh task, the training set consist of 150K bilingual sentences from CCMT2019 which including 1K sentence pairs for test set and validation set. We use directed graph discriminant method to segment the stems and affixes of source language. In addition, we limit the bilingual vocabulary to 35K words and limit the length of sentences to 50. We use BLEU scores[2] to evaluate the model. Parameters are set as follows: word embedding size = 300, hidden size = 512, number of layers=4, number of heads=6, dropout=0.25, batch size=128, and beam size=5. The parameters are updated by SGD and mini-batch with learning rate controlled by Adam. All of the source language use XLNet method[3] to obtain vector representation. We employ two TITAN X to train model and obtained by averaging the last 10 checkpoints for task. The information of dependency tree analysis is provided by Chinese tree database (CTB).

Baseline We compared our approach against various baselines:

- RNNSearch: RNNSearch[4] is a general end-to-end model built by RNN.
- Transformer: we also use Transformer[5] as the baseline because it is the mainstream machine translation framework at this stage, and it can show good performance in a variety of translation tasks.

In addition, we adopted BERT[6], GPT[7], XLNet pre-training method to compared with our pre-training method.

[1]This corpus has been uploaded to the database as supplementary material.

[2]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[3]https://github.com/zihangdai/xlnet

[4]https://github.com/lvapeab/nmt-keras

[5]https://github.com/tensorflow/tensor2tensor

[6]https://github.com/google-research/bert

[7]https://github.com/ConnorJL/GPT2

Model	WordEmb.	Mo-Zh	Ug-Zh	Ti-Zh
RNNSearch	-	25.42	28.62	24.39
RNNSearch	GPT	25.61	28.92	25.16
RNNSearch	BERT	26.03	29.65	25.84
RNNSearch	XLNet	27.01	30.09	26.2
Transformer	-	27.18	31.18	26.47
Transformer	GPT	27.39	31.26	27.25
Transformer	BERT	28.00	31.74	27.91
Transformer	XLNet	28.18	32.02	28.55
Ours(RNNSearch)	SRXLNet	27.91	32.11	28.75
Ours(Transformer)	**SRXLNet**	**28.53**	**32.43**	**29.19**

Table 1: Translation results in different languages.

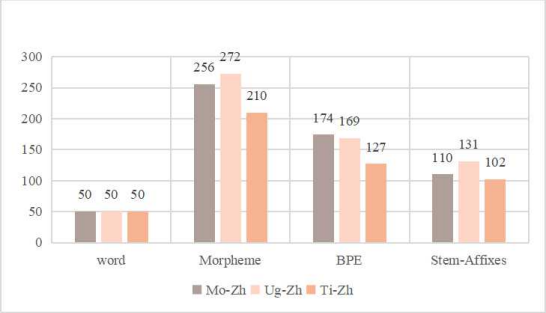

Figure 4: Sequence length corresponding to different granularity.

4.1 Results and Analysis

According to Table 1, the average BLEU scores of our method is higher than baseline system. Intuitively, Transformer has better generalization ability and translation quality than RNNSearch model, when dynamic word embedding (XLNet, GPT, BERT) is used as the basic modeling unit, BLEU scores is higher than static word embedding. In general, static word embedding is fixed in the fine-tuning stage, while dynamic word embedding will adjust dynamically according to the change of context, it show more advantages in dealing with polysemy.

On equal conditions, compared with XLNet, our method (SRXLNet) has the advantage that before learning language representation, the factorization order obtained by sampling is generated by syntactic dependency tree rather than by random extraction.

4.2 Ablation experiments

Our ablation experiments mainly divided the internal modules of the upstream pre-training model, and observed the impact of each module on the overall system.

We separated the languages into three different granularity: Morpheme, BPE, Stem-Affixes. The statistic of average length of sequences with different granularity are shown in figure 4.

We screened out all the sentences with 50 words and found that the average length of morpheme granularity was the 246 in three different tasks, the average length of BPE and stem-affixes granularity are 157 and 114, respectively. It can be seen from table 3 that the BLEU scores is the highest when stem-affixes is used, which compared with morpheme and BPE, the length of sequence can be shortened and the word formation features of language can be preserved by using stem-affixes. In addition, we found that the model improved by added fragment loop mechanism (FLM) and syntactic relevance sampling (DFS, BFS, SRXLNet) assisted word embedding representation.

Model	Mo-Zh		Ug-Zh		Ti-Zh	
	RNN	Transformer	RNN	Transformer	RNN	Transformer
+Morpheme	23.99	25.27	26.34	28.98	21.19	25.46
+BPE	25.03	25.98	27.12	30.56	22.35	26.1
+Stem-Affixes	25.24	27.18	28.62	31.18	24.39	26.47
+XLNet	27.01	28.18	30.09	32.02	26.2	28.55
+FLM	27.57	28.2	30.93	32.10	27.15	29.02
+SRXLNet-DFS	27.66	28.29	31.26	32.27	28	29.1
+SRXLNet-BFS	27.53	28.4	31.08	32.14	27.83	29.15
+SRXLNet	27.91	28.53	32.11	32.43	28.75	29.19

Table 2: The Ablation Experiment.

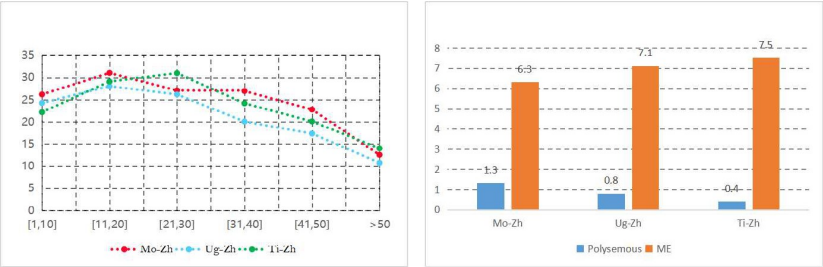

Figure 5: Analysis of sentence length and polysemy.

4.3 Case study

We mainly verify the BLEU scores of different sentence lengths and the ability of the model to deal with polysemous words. According to figure 5, when the sentence length is about 20 tokens, the model has better performance. With the increase of the length, BLEU scores also gradually decreases, when the sentence length is greater than 50, BLEU scores is the lowest.

For the polysemy phenomenon in the text, we mainly sample 300 sentences from each of the three target tasks, and manually detect the polysemous words and whether it causes ambiguity in the translation. The details are shown in figure 5.

According to figure 5, the average number of sentences with polysemy per 100 sentences sampled in the three target tasks is 1.3, 0.8 and 0.4, respectively. Since BLEU evaluation index is not sensitive to the polysemous words, we use manual evaluation(ME), and the total score is set at 10, which is used to score sentences. The scores of ME scores in three target tasks is 6.3, 7.1 and 7.5, respectively.

5 Conclusion

This paper presented a syntactic relevance XLNet word embedding pre-training method and evaluated its performance in various neural machine translation models. The conclusion is that the dynamic word embedding with context representation ability improved the performance of the model. When different sampling strategies are used to obtain the factorization sequence, the model extracted word embedding with rich syntactic relations from the disordered sequence of the permutation language model. In addition, we employed the fragment loop mechanism to alleviated the vanishing gradient in hyper-long sequence modeling. In the future, we will research the application of different word embedding generation methods in low-resource neural machine translation tasks.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL http://arxiv.org/abs/1409.0473.

Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. Federated learning of n-gram language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pp. 121–130, 2019. doi: 10.18653/v1/K19-1012. URL https://doi.org/10.18653/v1/K19-1012.

Kenneth Ward Church. Word2vec. *Nat. Lang. Eng.*, 23(1):155–162, 2017. doi: 10.1017/S1351324916000334. URL https://doi.org/10.1017/S1351324916000334.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 2978–2988, 2019. doi: 10.18653/v1/p19-1285. URL https://doi.org/10.18653/v1/p19-1285.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1243–1252, 2017. URL http://proceedings.mlr.press/v70/gehring17a.html.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. ISBN 978-0-521-87415-1. URL http://www.statmt.org/book/.

Jingyun Liu, Jackie Chi Kit Cheung, and Annie Louis. What comes next? extractive summarization by next-sentence prediction. *CoRR*, abs/1901.03859, 2019. URL http://arxiv.org/abs/1901.03859.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1412–1421, 2015. doi: 10.18653/v1/d15-1166. URL https://doi.org/10.18653/v1/d15-1166.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, 2014. doi: 10.3115/v1/d14-1162. URL https://doi.org/10.3115/v1/d14-1162.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 2699–2712, 2020. URL https://www.aclweb.org/anthology/2020.acl-main.240/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on*

Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 5754–5764, 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.