LingJing at SemEval-2022 Task 1: Multi-task Self-supervised Pre-training for Multilingual Reverse Dictionary

Bin Li[1]*, Yixuan Weng[2]*, Fei Xia[2,3]*, Bin Sun[1], Shutao Li[1], Shizhu He[2,3]
[1] College of Electrical and Information Engineering, Hunan University
[2] National Laboratory of Pattern Recognition, Institute of Automation, CAS
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences
{libincn, shutao_li, sunbin611}@hnu.edu.cn, wengsyx@gmail.com,
xiafei2020@ia.ac.cn, shizhu.he@nlpr.ia.ac.cn

Abstract

This paper introduces the result of Team LingJing's experiments in SemEval-2022 Task 1 Comparing Dictionaries and Word Embeddings (CODWOE)[1]. This task aims at comparing two types of semantic descriptions, including the definition modeling and reverse dictionary track. Our team focuses on the reverse dictionary track and adopts the multi-task self-supervised pre-training for multilingual reverse dictionaries. Specifically, the randomly initialized mDeBERTa-base model is used to perform multi-task pre-training on the multilingual training datasets. The pre-training step is divided into two stages, namely the MLM pre-training stage and the contrastive pre-training stage. As a result, all the experiments are performed on the pre-trained language model during fine-tuning. The experimental results show that the proposed method has achieved good performance in the reverse dictionary track, where we rank the 1-st in the Sgns targets of the EN and RU languages. All the experimental codes are open-sourced at `https://github.com/WENGSYX/Semeval`.

1 Introduction

The CODWOE shared task invites the participants to compare two types of semantic descriptions: dictionary glosses and word embedding representations. The intuitions come from the questions: "Are these two types of representation equivalent? Can we generate one from the other?". To study this question, the CODWOE proposes two sub-tracks: a definition modeling track (Noraset et al., 2017), where participants have to generate glosses from vectors, and a reverse dictionary track (Hill et al., 2016), where participants have to generate vectors from glosses. These two tracks are fairly challenging (Hill et al., 2016), where more efficient methods are required to be designed for implementation.

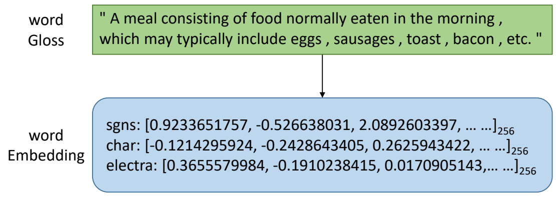
word
Gloss
" A meal consisting of food normally eaten in the morning ,
which may typically include eggs , sausages , toast , bacon , etc. "
word
Embedding
sgns: [0.9233651757, -0.526638031, 2.0892603397, … …]256
char: [-0.1214295924, -0.2428643405, 0.2625943422, … …]256
electra: [0.3655579984, -0.1910238415, 0.0170905143,… …]256

Figure 1: An example of the reverse dictionary task. Given the word gloss, it is required to generate the vectors of their corresponding Sgns, Char, and Electra, respectively.

These tasks are also useful for explainable AI, since they involve converting human-readable data into machine-readable data and back (Li et al., 2021). In this paper, we focus on the reverse dictionary track. As shown in Figure 1, given the gloss "A meal consisting of food normally eaten in the morning, which may typically include eggs, sausages, toast, bacon, etc.", the reverse dictionary task requires us to generate corresponding three sets of 256-dimensional word vectors. The Sgns (Mikolov et al., 2013) , char (Vakulenko et al., 2017), and Electra (González et al., 2020) are skip-gram with negative sampling embeddings, character-based embeddings, and Transformer-based contextualized embeddings, respectively.

It is noted that this task comprises datasets in 5 languages: English, Spanish, Italian, French, and Russian. The reverse dictionary task is difficult due to the significant inborn differences between word vectors and glosses and the vast differences between languages (Bosc and Vincent, 2018).

To solve the above problems, we use a multi-task self-supervised pre-training approach with Masked language modeling (MLM) (Taylor, 1953; Devlin et al., 2019) and contrastive learning (Reimers and Gurevych, 2019; Su et al., 2021). On the one hand, MLM can better capture the semantic representation of the input text (Liu et al., 2019). On the other hand, contrast learning can further improve the performance of downstream regression tasks (Jaiswal

*These authors contribute equally to this work.

[1] https://codwoe.atilf.fr/

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 29 - 35
July 14-15, 2022 ©2022 Association for Computational Linguistics

et al., 2020). Specifically, we use a randomly initialized mDeBERTa-base (He et al., 2021) model to perform MLM pre-training on five text datasets in different languages. Contrastive pre-training (Gao et al., 2021) is then performed using vectors with and without dropout (Srivastava et al., 2014). Afterward, the model is fine-tuned using the Reverse Dictionary dataset. The experimental results show that the proposed method has achieved good performance in the reverse dictionary track. We achieve the top three results on the Sgns evaluation metrics in all languages. Specifically, we get first place in English and Russian, second place in Spanish and French, and third place in Italian.

2 Main method

In this section, we will elaborate on the main methods for the reverse dictionary track. As the pre-training method can enhance the performance of semantic representation (Qiu et al., 2020), we adopt masked language modeling (MLM) task (Devlin et al., 2019) and contrastive pre-training task (Jaiswal et al., 2020) for implementing this regression task.

2.1 Masked language modeling task

Masked language modeling (MLM) task consists of giving the model a random masked sentence and optimizing the weights inside the model to output the unmasked sentence on the other side. We implement the MLM pre-training method with the same original setting as BERT (Devlin et al., 2019). What's more, we adopt the standard implementations of the MLM from the website[2].

2.2 Contrastive pre-training task

Our method follows the SimCSE (Gao et al., 2021) method, where the self-supervised model is adopted for the contrastive pre-training task. For the self-supervised part, we use dropout to add noise to the text twice, thus constructing a pair of positive samples, and pairs of negative samples are sentences processed with the dropout in the batch. The above processes can be formulated as the equation (1)

$$\mathcal{L}_{\mathrm{CL}} = -\log \frac{\exp\left(\mathrm{sim}\left(\tilde{h}_i, h_i\right)/\tau\right)}{\sum_{j=1}^{n} \exp\left(\mathrm{sim}\left(\tilde{h}_i, h_j\right)/\tau\right)}, \quad (1)$$

where the h_i represents the hidden feature of the positive sample, while the h_j is the hidden feature of the negative drop-out sample. The τ is a temperature hyper-parameter and sim(,) means the cosine similarity function.

2.3 Multi-task pre-training

Multi-task learning is known to fully enhance the performance of the single task with multiple related tasks to be designed and optimized (Sanh et al., 2021). We combine the above two pre-training task to the multi-task objectives, where the final loss function can be represented as follows

$$\mathcal{L} = \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{\mathrm{CL}}. \quad (2)$$

2.4 Downstream fine-tuning

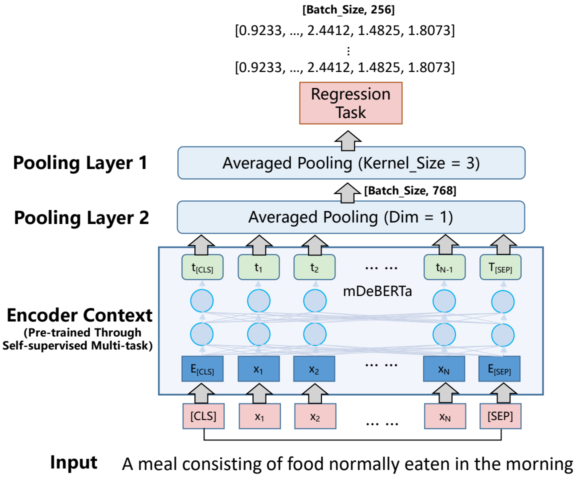

Figure 2: Main structure of the proposed method.

Concretely, given the input sentence, the semantic representation can be obtained through the context encoder with pre-training. As shown in the Figure 2, we use the pre-trained language model through self-supervised multi-task pre-training as the backbone for the regression task. Once obtaining the final representation of the pre-trained language model, two pooling layers (Lin et al., 2013) are designed to get the useful features with the probable size. The mean pooling layer is added on top of the pre-trained model for squeezing the features. Another pooling layer (with the kernel_size=3) is added before the final regression task.

3 Experimental setup

3.1 Data Description

The CODWOE shared task provides datasets in five different languages (EN, ES, FR, IT, RU). For these datasets of five languages, each dataset has 43,608

[2] https://github.com/lucidrains/mlm-pytorch

training sets, 6375 dev sets and 4208 test sets. Each language contains multiple embeddings containing “Char” and “Sgns”, while English, French and Russian have the embedding “Electra”. We will introduce these datasets as follows.

Char corresponds to character-based embeddings, computed with an auto-encoder on the spelling of a word. In addition, the “gloss” key in each dataset is the source in the reverse dictionary track. We need to use “gloss” to generate the associated embeddings.

Sgns corresponds to skip-gram with negative sampling embeddings (aka. word2vec (Mikolov et al., 2013)).

Electra corresponds to the Transformer-based (Vaswani et al., 2017) contextualized embeddings.

Moreover, the organizers want the shared task to be as linguistically relevant as possible and hope to provide a fair competition environment for all participants. The organizer forbids the use of external resources and pre-trained language models in CODWOE.

3.2 Evaluation metrics

In this task, the performance of the system is evaluated through three evaluation indicators (Mickus et al., 2022).

Mean squared error (MSE) between the submission’s reconstructed embedding and the reference embedding.

Cosine similarity (Cossim) between the submission’s reconstructed embedding and the reference embedding.

$$\text{MSE} = \frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{A_i - B_i}{\sigma_i}\right)^2$$

$$\text{Cossim} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

where the A and B refer to two matrices that need to be calculated.

Cosine-based ranking[3] between the submission’s reconstructed embedding and the reference embedding; i.e., how many other test items have a cosine similarity with the reconstructed embedding higher than that with the reference embedding.

3.3 Method introduction

The Baseline provided by the organizer[4] uses the encoder structure of the Transformer (Vaswani et al., 2017; Wolf et al., 2020) framework. After each token passes through the embedding layer, positional encoding will be added to indicate the location structure of the token. Then it will be input to the encoder based on the transformer and finally output to the linear layer to make the dimension of the matrix consistent with the label.

In addition, the organizer has made some improvements to the baseline.

1. The principled way of selecting hyperparameters (using Bayesian Optimization (Snoek et al., 2012; Frazier, 2018)).

2. A sentence-piece re-tokenization, to ensure the vocabulary is of the same size for all languages.

3. The beam-search (Wiseman and Rush, 2016; Freitag and Al-Onaizan, 2017) decoding for the definition modeling pipeline.

Our method uses the randomly initialized mDeBERTa (He et al., 2021) model. The mDeBERTa improves the BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models using disentangled attention and enhanced mask decoder. It shares the base model with 12 layers and 768 hidden size, which is pre-trained on the multilingual corpus. It has 86M backbone parameters with a vocabulary containing 250K tokens which introduce 190M parameters in the Embedding layer. It supports most languages around the world, since it is believed that there should be some shared semantic features between different languages[5].

3.4 Implementation details

We use the hugging-face[6] (Wolf et al., 2020) framework and train the model based on the Pytorch (Paszke et al., 2019). During training, we employ the AdamW optimizer (Loshchilov and Hutter, 2017). The default learning rate is set to 1e-5 with the warm-up (He et al., 2016). Four 3090 GPUs are used for all experiments.

[3] Specific implementations can refer to https://github.com/WENGSYX/Semeval.

[4] Specific implementation can refer to https://github.com/TimotheeMickus/codwoe/tree/main/baseline_archs

[5] Please refer: https://ai.glossika.com/blog/a-map-to-the-syntax-of-all-spoken-languages

[6] https://github.com/huggingface/transformers

Experimental Items	Baseline			Ours		
Language	MSE	Cosine	Ranking	MSE	Cosine	Ranking
English	0.91092	0.15132	0.49030	0.86239	0.24310	0.32907
Espana	0.92996	0.20406	0.49912	0.85770	0.35275	0.25101
French	1.14050	0.19774	0.49052	1.02968	0.32799	0.28213
Italian	1.12536	0.20430	0.47692	1.03945	0.35955	0.22995
Russian	0.57683	0.25316	0.49008	0.52827	0.42440	0.18711

Table 1: Results of the Sgns track.

Experimental Items	Baseline			Ours		
Language	MSE	Cosine	Ranking	MSE	Cosine	Ranking
English	0.14776	0.79006	0.50218	0.47103	0.00331	0.48599
Espana	0.56952	0.80634	0.49778	0.50121	0.85770	0.35275
French	0.39480	0.75852	0.49945	0.96678	0.00809	0.51862
Italian	0.36309	0.72732	0.49663	0.88129	-0.02992	0.49603
Russian	0.13498	0.82624	0.49451	0.47905	0.00479	0.47228

Table 2: Results of the Char track.

On the MLM pre-training task, we alternately carry out the pre-training tasks of long text and short text. After mixing the data sets of five different languages, we train them for 40 epochs. In detail, we classify all data sets with a text length of 30. In each epoch, firstly, samples with text length less than or equal to 30 are trained with a maximum length of 32 tokens (including <CLS> and <SEP>) and the batch size is set to 70. Then we change the maximum length to 160 tokens and set the batch size to 18 for training the remaining samples.

Referring to the settings of WWM (Cui et al., 2021; Joshi et al., 2020), we use the text mask rate with a probability of 20%, and adopt that the 1, 2, 3, 4 n-gram masking length with a probability of 85%, 5%, 5%, and 5%.

In contrastive pre-training, we repeatedly integrate a sample into the model twice. During this period, because our model has dropout, it will add noise to the input, so that the output of the two times is distinct. As a result, our method can be improved in the sentence representation ability through self-supervised.

Based on the pre-trained language model, we fine-tune with the maximum length of all samples to 100 tokens, the batch size to 50 (there will be 2 * 50 samples for each step to be calculated by the model at the same time). The number of training epochs is 40.

4 Results and discussions

In this section, we introduce the experimental results of the Sgns, the Char and the Electra tracks. The online results and further discussions are also presented.

4.1 Experimental results

The experimental results of the Sgns, Char and Electra can be found in the Table 1, 2 and 3. Specifically, for the Sgns track, we outperform the experiments of each baseline according to all the metrics. The reason may be that the pre-training method with MLM and contrastive learning can well provide well-formed vector space representations between samples. As for the Char and Electra track, the baseline is better than ours. It may be because the word and contextual character features are hard to be captured due to the smaller corpus. In the future, we will explore more efficient methods to perform well definition modeling in these tracks.

4.2 Official online results

As shown in Table 4, we achieve the top three results on the Sgns evaluation metrics in all languages. Specifically, we get first place in English and Russian, second place in Spanish and French, and third place in Italian. Our method is effective on the Electra evaluation metrics, but not the best. Our team ranks the second place, fourth and fourth place in Russian, English, and French, respectively. Our approach does not achieve good results on the char metric, which represents the character level. This result may be that it is difficult for the model to capture semantics while maintaining high precision letter-level fine-grained word vector learning.

Experimental Items	Baseline			Ours		
Language	MSE	Cosine	Ranking	MSE	Cosine	Ranking
English	1.41287	0.84283	0.49849	1.50876	0.84592	0.47773
French	1.15348	0.85629	0.49784	1.27066	0.85859	0.47762
Russian	0.87358	0.72086	0.49120	0.82773	0.73397	0.42020

Table 3: Results of the Electra track.

Online	Sgns					Char					Electra		
TEAM	EN	ES	FR	IT	RU	EN	ES	FR	IT	RU	EN	FR	RU
LingJing(ours)	1	2	2	3	1	7	5	5	6	5	4	4	2
pzchen	2	4	3	2	3	3	1	1	1	1	1	1	1
IRB-NLP	3	1	1	1	2	4	3	4	2	2	5	3	3
Locchi	4	/	/	4	/	1	/	/	4	/	3	/	/
Nihed_Bendahman_	5	5	4	6	4	2	2	2	3	4	2	2	4
zhwa3087	6	6	5	5	5	6	4	3	5	3	/	/	/
the0ne	7	/	/	/	/	5	/	/	/	/	6	/	/
tthhanh	8	7	6	7	6	/	/	/	/	/	/	/	/

Table 4: Results of the online official Rank.

5 Conclusion

In this paper, it is mainly introduced that in order to solve the reverse dictionary track in Semeval-22 CODWOE, the LingJing team makes the model have the ability of semantic understanding through the MLM task with contrastive learning in the randomly initialized mDeBERTa model. After that, we report the performance of our model in CODWOE, and obtain the best performance in English and Russian tasks of Sgns dataset, which proves that our method is effective. In the future, we will further study how to make full use of the characteristics of different languages and make the model embed the text into a more accurate vector space.

Acknowledgement

This work is supported by the National Key R&D Program of China (2018YFB1305200), the National Natural Science Fund of China (62171183).

References

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter I. Frazier. 2018. A tutorial on bayesian optimization.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *CoRR*, abs/1702.01806.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

José Ángel González, Lluís-F Hurtado, and Ferran Pla. 2020. Transformer based contextualization of pre-trained word embeddings for irony detection in twitter. *Information Processing & Management*, 57(4):102262.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *CoRR*, abs/2011.00362.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957, Online. Association for Computational Linguistics.

M. Lin, Q. Chen, and S. Yan. 2013. Network in network. *Computer Science*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Timothee Mickus, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2022. SemEval-2022 Task 1: Codwoe – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3259–3266. AAAI Press.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2951–2959, Red Hook, NY, USA. Curran Associates Inc.

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.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Svitlana Vakulenko, Lyndon Nixon, and Mihai Lupu. 2017. Character-based neural embeddings for tweet clustering. *SocialNLP 2017*, page 36.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *CoRR*, abs/1606.02960.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.