RUC_AIM3 at TRECVID 2020: Ad-hoc Video Search & Video to Text Description

Yida Zhao, Yuqing Song, Shizhe Chen and Qin Jin*
School of Information, Renmin University of China
{zyiday, syuqing, cszhe1, qjin}@ruc.edu.cn

Abstract

In this report, we present our solutions for the two tasks of TRECVID 2020 [1]: Ad-hoc Video Search (AVS) and Video to Text Description (VTT). For the AVS task, we adopt a two-branch framework including a global matching branch and a fine-grained matching branch. In the global matching branch, we employ VSE++ [2] and Dual Encoding [3] models to capture the global information of video and text. In the fine-grained matching branch, we adopt the hierarchical matching model HGR [4] to match the video and text at more fine-grained level. For the VTT Matching and Ranking subtask, we use the same two-branch model as the AVS task and further improve it with hubness mitigation as [5] at inference time. For the VTT Description Generation subtask, we employ a two-layer LSTM as the language decoder to generate video descriptions at both scene-level and object-level and late fuse them with hybrid reranking. Our team RUC_AIM3 finally ranks the 1st place on both AVS and VTT tasks in TRECVID 2020.

1 Ad-hoc Video Search

1.1 Approach

Ad-hoc video search task aims to retrieve video clips with a text query. Given the query, the AVS task requires to retrieve the most relevant top 1000 video clips from the V3C vimeo collection [6] which contains 1,082,659 video clips.

The main challenge of this task is the semantic matching between video and text. Most recent works learn a joint visual-semantic embedding to measure the cross-modal similarities [2, 3]. They first encode the video and text as global feature vectors respectively and then map them into a joint embedding space. We call such models as global matching models, which are shown good abilities to capture the global information of video and text for the overall matching and achieve promising results in cross-modal video-text retrieval tasks.

However, the single global encoding vector is insufficient to represent complicated details of video and text, such as scenes, objects, actions and their compositions. In order to capture both global and local details, we propose a fine-grained matching model called Hierarchical Graph Reasoning (HGR) [4]. The HGR model decomposes video-text matching into global-to-local levels. It takes the advantage of global and local matching approaches and makes up their deficiencies.

Since the global matching models and fine-grained matching models are complementary, our Ad-hoc Video Search System combines these two branches through a late fusion strategy to achieve better performance. We will introduce our system in details in the following subsections.

*Qin Jin is the corresponding author.

1.1.1 Global Matching

In the global matching branch, we adopt two effective models: VSE++ [2] and Dual Encoding [3].

The VSE++ takes the mean pooling of frame-level features as the global features of video clips and concatenates the forward and backward hidden states of bidirectional GRU (biGRU) as the global features of text queries. A fully connected layer is adopted to map them into the joint embedding space. The main contribution of the VSE++ model is the proposed triplet loss function with hard negative mining, which is used in each model in our retrieval system.

The Dual Encoding improves the video encoder and text encoder of VSE++. Given a sequence of input features, three levels encoder (mean pooling, biGRU, and biGRU-CNN) are used to encode global, temporal and local information respectively. The encoded features from three levels are then concatenated into a single feature vector and mapped into the joint embedding space.

We train the VSE++ and Dual Encoding models respectively and late fuse them by averaging the similarity scores during inference time. Furthermore, we also train the above two models using BERT as the text encoder instead.

1.1.2 Fine-grained Matching

We employ the HGR model [4] in our fine-grained matching branch. The HGR model disentangles text query into a hierarchical semantic graph including three levels of events, actions, entities. Then it generates hierarchical textual embeddings via attention-based graph reasoning. The three levels are responsible to capture global events, local actions, entities respectively. Different levels of text are used to guide the learning of diverse and hierarchical video representations. Cross-modal matchings at all three levels are aggregated to compute the final cross-modal similarities. More details of the HGR model can be found in our previous work [4].

1.2 Experiments

We employ the MSRVTT [7], TGIF [8] and VATEX [9] video captioning datasets as our training set, and TRECVID VTT 2016 as the validation set. Besides these video captioning datasets, to improve the generalization ability of our system, we further adopt the image captioning dataset MSCOCO [10] to train the global matching models. We extract video features with ResNeXt-101 [11] pre-trained on billion scale weakly-supervised data [12] and irCSN-152 [13] pre-trained on IG-65M [14]. For the model trained on image captioning dataset, only ResNeXt-101 is used to extract image features.

For ad-hoc video search task, we submit four runs as follows:

- Run 4: Global matching branch (Ensemble of VSE++ and Dual Encoding models trained on video captioning datasets).
- Run 3: Run 4 + global matching models trained on image captioning datasets.
- Run 2: Run 3 + Fine-grained matching branch (HGR).
- Run 1: Run 2 + global matching models with BERT as the text encoder.

Table 1: Results of different runs on TRECVID 2019 and 2020 AVS Main Task.

Runs	Method	Training Data	Results	
			2019	2020
Winner in 2019 [15]	-	-	0.163	-
Run 4	Global	Video	0.177	0.354
Run 3	Global	Video + Image	0.193	0.350
Run 2	Global & Fine-grained	Video + Image	0.195	0.357
Run 1	Global & Fine-grained, +BERT	Video + Image	**0.196**	0.359
Run 5*	Global & Fine-grained, +BERT	Video	0.181	**0.361**

Table 1 presents the infAP performances of four submitted runs on TRECVID 2019 & 2020 AVS Main Task, which contains 30 and 20 text queries respectively. All of our four runs significantly outperform the winner solution in 2019 [15]. Run 4 is our baseline model which only contains global matching branch and is trained on video captioning datasets. The performances of Run 4 to Run 1 on

TRECVID 2019 dataset are gradually improved with additional components added, which includes the models trained on image captioning dataset, fine-grained matching branch and the models using BERT as the text encoder. However, unlike the trend on 2019 dataset, the performance of Run 3 on 2020 dataset decreases after adding models trained on image captioning dataset. Nevertheless, our Run 1 achieves the best result among all participating teams with the infAP of 0.359. Since the ground-truth of this year has been released, we re-tested a new submission called Run 5, which removed models trained on image captioning dataset, and achieved a better performance with the infAP of 0.361. Since the queries are different in 2019 and 2020 datasets, the contribution of using the additional image dataset varies a lot.

Table 2: Results of TRECVID 2020 AVS Progress Subtask (10 queries).

Runs	Results
Winner in 2019 [15]	0.177
Run 4	**0.235**
Run 3	0.208
Run 2	0.220
Run 1	0.223

Table 2 shows the infAP performance of the four submitted runs on TRECVID 2020 AVS Progress Subtask, which contains 10 text queries. The performances of our 4 runs are all better than the winner solution in last year [15]. The Run 4 achieves the best performance among the 4 runs, because all other 3 runs utilize image captioning dataset in training, which might not be suitable for queries in 2020 as shown in Table 1.

2 Video-to-Text Description

2.1 Matching and Ranking

The VTT matching and ranking subtask aims to rank a list of sentences for a given video based on their semantic relevance. In this year, there are 1,700 videos selected from V3C vimeo collection [6] and 5 sentence sets with 1,720 sentences in each of them. It is similar to the AVS task, except that it is video-to-text retrieval while the AVS task is text-to-video retrieval.

2.1.1 Approach

For VTT matching and ranking subtask, we adopt two-branch model similar to our Ad-hoc Video Search System, including global matching branch and fine-grained matching branch. We employ Dual Encoding [3] in the global matching branch and HGR [4] in the fine-grained matching branch.

The hubness problem [5] is common in high-dimensional space learning, which means that some texts can be the nearest neighbors for multiple videos. However, we want to retrieve different texts rather than the same "hub" text to different video queries. We follow [5] to employ Inverted Softmax [16] to mitigate the hubness problem. It scales down the similarity $s(v,t)$ between video v and text t if t is also close to other video queries.

$$s'(v,t) = \frac{e^{\beta s(v,t)}}{\sum_{\overline{v} \in V \setminus \{v\}} e^{\beta s(\overline{v},t)}} \tag{1}$$

where V denotes all video queries and β is a hyperparameter temperature which is set as 30.

2.1.2 Experiments

For VTT matching and ranking subtask, we submit four runs as follows:

- Run 4: Global matching branch with hubness mitigation. (Single Dual Encoding model)
- Run 3: Fine-grained matching branch with hubness mitigation. (Single HGR model)
- Run 2: Global matching branch with hubness mitigation (Ensemble).
- Run 1: Global matching branch and Fine-grained matching branch with hubness mitigation (Ensemble).

Table 3: Results of TRECVID 2020 VTT matching and ranking subtask.

Ours	SetA	SetB	SetC	SetD	SetE
Run 4	0.606	0.611	0.621	0.618	0.636
Run 3	0.627	0.621	0.620	0.620	0.641
Run 2	0.683	0.692	0.691	0.696	0.711
Run 1	**0.714**	**0.711**	**0.707**	**0.721**	**0.731**

Table 3 presents the mean inverted rank metric results of four submitted runs on TRECVID 2020 VTT matching and ranking subtask. Run 4 and Run 3 are single models that use global matching branch and fine-grained matching branch, respectively. The performance of fine-grained matching is better than global matching. Run 2 and Run 1 show that using ensemble of multiple models in each branch can significantly improve the performance. Run 1 combines the global matching branch and fine-grained matching branch and achieves the best results.

2.2 Description Generation

2.2.1 Approach

Compared with selecting descriptions from the corpus through matching [2], the description generation subtask is more challenging which aims to automatically generate a natural language sentence to describe the video content [17]. Following with previous works [17, 18], we employ the encoder-decoder architecture [19] for this subtask. Considering the complexities of videos at both spatial and temporal structures, we encode the video at both scene-level and object-level to capture abundant video information for the description generation. For the language decoder, we employ a two-layer LSTM to generate descriptions with temporal and spatial attentions on the above two kinds of encoding features and late fuse them via hybrid reranking. In the following, we will introduce each component of our model in details.

Scene-level and Object-level Video Encoding. In order to comprehensively encode videos, we extract two types of video features for temporal and spatial attention respectively. In the temporal branch, we represent the video as a sequence of segment-level multi-modal features $V^T = \{v_1^T, \ldots, v_n^T\}$. Each segment-level feature is the concatenation of video features from 2D (ResNeXt-101 [11]) and 3D (irCSN [13]) CNNs. In the spatial branch, we employ Faster-RCNN [20] pretrained on Visual Genome [21] to extract grounded region features for each frame of the video, and select the top-K region features $V^S = \{v_1^S, \ldots, v_K^S\}$ according to their predicted scores. The V^T and V^S are then used as the scene-level and object-level video encoding features respectively.

Language Decoder with Temporal and Spatial Attentions. Based on the encoded video features, we can generate video descriptions with temporal and spatial attentions. We employ a two-layer LSTM [22] as the language decoder to generate description words based on ctx_t^T and ctx_t^S respectively. The decoder includes an attention LSTM and a language LSTM. The attention LSTM takes the previous word embedding w_{t-1} and previous output from language LSTM h_{t-1}^l as input to compute an attentive query h_t^a as follows:

$$h_t^a = \mathrm{LSTM}([w_{t-1}; h_{t-1}^l], h_{t-1}^a; \theta^a) \tag{2}$$

where $[;]$ is vector concatenation and θ^a are parameters.

With the computed attention query h_t^a, the captioning model learns to focus on the relevant temporal frames and spatial regions for each word's generation as follows:

$$z_t^T = \mathrm{softmax}(h_t^a W^T (V^T)^T) V^T \tag{3}$$

$$z_t^S = \mathrm{softmax}(h_t^a W^S (V^S)^T) V^S \tag{4}$$

Then the language LSTM is fed with z_t^* and h_t^a to generate words sequentially:

$$h_t^l = \mathrm{LSTM}([z_t^*; h_t^a], h_{t-1}^l; \theta^l), * \in [T, S] \tag{5}$$

$$p(y_t|y_{<t}) = \mathrm{softmax}(W_p h_t^l + b_p) \tag{6}$$

where θ^l, W_p and b_p are parameters.

We train the whole model with cross entropy (XE) loss and further improve it via reinforcement learning (RL) [23] with CIDEr [24] as the sequence-level reward function to address the exposure bias and target mismatch [25] problems in MLE. The XE loss and RL loss for a single ground-truth pair (v, y^*), where $y^* = \{y^*_1, \ldots, y^*_L\}$, are:

$$\mathcal{L}_{xe} = -\frac{1}{L}\sum_{t=1}^{L}\log p(y^*_t|y^*_{<t}, v) \tag{7}$$

$$\mathcal{L}_{rl} = -\frac{1}{L}r(y^s)\sum_{t=1}^{L}\log p(y^s_t|y^s_{<t}, v) \tag{8}$$

where $y^s = \{y^s_1, \cdots, y^s_L\}$ is a paragraph sampled from the model and $r(\cdot)$ is the reward function, which is defined with CIDEr.

Hybrid Reranking. Considering that the scene-level and object-level captioning models are complementary, we late fuse the two models with hybrid reranking. Another language model and video-text matching model are trained to evaluate the generated descriptions from language fluency and visual relevance perspectives. The language model is another LSTM pre-trained on the ground-truth caption corpus, which can be used to evaluate the language fluency of generated ones. The cross-modal semantic matching model is trained as in Matching and Ranking subtask, and be fixed to evaluate the visual relevance of generated captions. We can rerank the captions generated from the two models by the weighted sum of fluency score and relevancy score, and choose the best description for the video.

2.2.2 Experiments

We employ the TGIF [8], MSRVTT [7], VATEX [9], TRECVID VTT 2016-2018 video captioning datasets as our training set, and TRECVID VTT 2019 as our validation set. To verify the effectiveness of the proposed captioning model, we first conduct experiments on the MSRVTT dataset with different captioning models, including the AoANet [26] with attention on attention, LSTM version of X-LAN [27] with infinity order feature interaction, Transformer [28] and our two-layer LSTM model.

Table 4: Performance comparison of different captioning models on MSRVTT validation set.

Models	BLEU@1	BLEU@2	BLEU@3	BLEU@4	CIDEr	METEOR	SPICE
Trained with Cross-Entropy Loss							
AoANet	83.20	69.88	55.98	42.73	51.97	29.54	7.01
X-LAN	84.07	**71.95**	**58.94**	**46.65**	**58.63**	30.48	7.41
Transformer	**85.10**	70.98	56.78	43.63	51.61	**30.94**	7.60
Ours	82.14	67.58	53.08	40.68	53.32	30.50	**7.87**
Trained with Reinforcement Learning							
AoANet	85.83	71.89	57.15	43.62	60.39	30.41	7.71
X-LAN	**87.96**	**74.91**	**60.77**	**47.46**	59.69	**31.68**	7.92
Transformer	85.51	71.02	55.72	42.11	53.88	30.08	7.85
Ours	87.89	74.42	59.54	45.61	**62.06**	31.41	**8.00**

The results in Table 4 show that the LSTM-based models perform better than the Transformer on the captioning task, which might come from two reasons. Firstly, the transformer structure has the advantage in long text generation, however, the short video captioning task usually generates descriptions of no longer than 20 words. Secondly, for the captioning task, visual understanding and grounding is more important than the textual context modeling. Therefore, the transformer model does not show its advantages as in machine translation task [28]. Our two-layer LSTM model and the X-LAN model are the best two models, which are adopted in the following experiments.

Table 5 shows the results of the above two models on scene-level and object-level video features. Our model achieves competitive results with the X-LAN model on scene-level video features, while outperforms it on the object-level features. It also shows that models with spatial attention alone are inferior to the temporal attention models, which infers the temporal information is more important than the spatial information in videos. Combining our models on different features with hybrid

Table 5: Results with different visual features on TRECVID VTT 2019 dataset.

Models	Loss	BLEU@1	BLEU@2	BLEU@3	BLEU@4	CIDEr	METEOR	SPICE
Trained with Scene-level Video Features								
X-LAN	XE	57.67	39.87	26.35	16.78	30.23	16.16	10.94
Ours	XE	59.53	38.93	24.81	15.47	30.29	15.47	10.71
X-LAN	RL	**66.67**	**45.13**	29.21	18.13	36.01	17.30	11.60
Ours	**RL**	66.52	45.01	**29.24**	**18.19**	**36.30**	**17.37**	**11.63**
Trained with Object-level Video Features								
X-LAN	XE	57.81	39.55	26.06	16.62	28.85	15.95	10.64
Ours	XE	61.18	40.02	25.50	15.80	32.17	17.00	**11.64**
X-LAN	RL	65.85	44.40	28.64	17.78	32.96	17.02	11.18
Ours	**RL**	**65.87**	**44.84**	**29.13**	**18.04**	**35.15**	**17.29**	11.62
Hybrid Reranking								
Ours	**RL**	**67.75**	**46.48**	**30.30**	**18.80**	**38.45**	**17.96**	**12.32**

reranking shows significant improvements due to the complementarity of the temporal and spatial models.

Finally, we submit four runs as follows, and their final evaluation results on TRECVID VTT 2020 dataset are shown in Table 6.

- Run 4: Our single best model.
- Run 3: Ensemble of the captioning models trained on object-level visual features.
- Run 2: Ensemble of the captioning models trained on scene-level visual features.
- Run 1: Ensemble of run2 and run3 by captions reranking.

Table 6: Results of the submitted four runs on TRECVID VTT 2020 dataset.

Runs	BLEU@4	CIDEr	METEOR	SPICE
4	5.11	28.40	29.64	10.20
3	5.27	27.70	29.65	10.30
2	5.42	28.90	30.28	10.70
1	**5.56**	**30.30**	**31.02**	**11.00**

3 Conclusions

In this report, we present our systems for the Ad-hoc Video Search (AVS) and Video to Text Description (VTT) tasks in TRECVID 2020 challenge. For the AVS task, we adopt a two-branch architecture which includes a global matching branch and a fine-grained matching branch to match videos and texts at both global and fine-grained levels. For the VTT task, we propose to integrate temporal and spatial attentions for the captioning model based on scene-level and object-level video features. Hybrid reranking is employed to ensemble different models according to the language fluency and visual relevance qualities of generated captions. Our systems achieve the best performance on both tasks in the TRECVID 2020 challenge.

4 Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 61772535) and Beijing Natural Science Foundation (No. 4192028).

References

[1] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020.

[2] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, 2018.

[3] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Fangyu Liu and Rongtian Ye. A strong and robust baseline for text-image matching. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019.

[6] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A Butt. V3c–a research video collection. In *International Conference on Multimedia Modeling*. Springer, 2019.

[7] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] Wang Xin, Wu Jiawei, Chen Junkun, Li Lei, Wang Yuan-Fang, and Wang William Yang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *IEEE International Conference on Computer Vision*, 2019.

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, 2014.

[11] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision*, 2018.

[13] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *IEEE International Conference on Computer Vision*, 2019.

[14] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[15] Xiang Wu, Da Chen, Yuan He, Hui Xue, Mingli Song, and Feng Mao. Hybrid sequence encoder for text based video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*, 2017.

[17] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. In *IEEE International Conference on Computer Vision*, 2015.

[18] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014.

[20] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Annual Conference on Neural Information Processing Systems*, 2015.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1), 2017.

[22] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[23] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

[24] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[25] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[26] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019.

[27] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[28] 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*, 2017.