# Sparse Graph to Sequence Learning for Vision Conditioned Long Textual Sequence Generation

**Anonymous Authors**[1]

## Abstract

Generating longer textual sequences when conditioned on the visual information is an interesting problem to explore. The challenge here proliferate over the standard vision conditioned sentence-level generation (e.g., image or video captioning) as it requires to produce a brief and coherent story describing the visual content. In this paper, we mask this Vision-to-Sequence as Graph-to-Sequence learning problem and approach it with the Transformer architecture. To be specific, we introduce **S**parse **G**raph-to-**S**equence **T**ransformer (SGST) for encoding the graph and decoding a sequence. The encoder aims to directly encode graph-level semantics, while the decoder is used to generate longer sequences. Experiments conducted with the benchmark image paragraph dataset show that our proposed achieve 13.3% improvement on the CIDEr evaluation measure when comparing to the previous state-of-the-art approach.

## 1. Introduction

Most of the methods which address vision conditioned textual sequence generation have concentrated on shorter sequences (e.g., phrase or sentence). Usually, these methods employ a standard encoder-decoder framework (Cho et al., 2014; Bahdanau et al., 2014), where the encoder encodes an image into fixed vector representation and then the decoder decodes them into a textual sequence. Several improvements were seen in the recent years over earlier proposed methods where visual features are upgraded with bottom-up (Anderson et al., 2017) encoding, encoder-decoder architecture added with attention (Xu et al., 2015) and training is achieved with reinforcement for sequence decoding (Rennie et al., 2016). However, most of these methods fail to capture salient objects observed in the image and generate textual sequences which are generic and simple. A possible reason identified (Yao et al., 2018) is that visually grounded language generation is not end-to-end and largely attributed to the high-level symbolic reasoning. It is also observed that the high-level reasoning is natural for humans as we inherently incorporate *inductive bias* based on common sense or factual knowledge into language (Kennedy et al., 2007), however, this is ineffective for vision conditioned textual sequence generation due to gap between visual information and language composition. This gap widens more when longer textual sequences need to be generated when conditioned on visual information.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

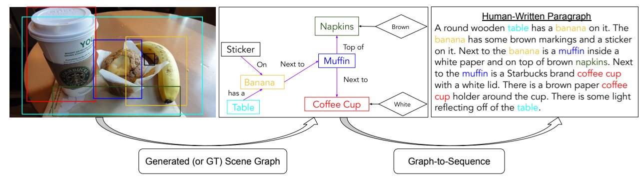


*Figure 1.* An example of graph-to-sequence from the image-paragraph dataset. Rectangles represent objects, connections between objects represent their relationships and diamond represent the attributes of the objects.

In NLP, structured inputs (e.g., graph structures, table data) are omnipresent as a representation of natural language. Recently, several works (Beck et al., 2018) have explored changing them into sequences for different applications (Song et al., 2018; Koncel-Kedziorski et al., 2019). Inspired from it, we propose to incorporate graph structure as an inductive bias for vision conditioned textual sequence generation. This is achieved by abstracting visual information (e.g., image) into a scene graph (Johnson et al., 2015) to add complementary strength of symbolic reasoning to multimodal learning. Scene graphs connect the visual objects, their attributes, and their relationships in an image by directed edges. Figure 1 presents the visualization of the overall idea.

However, the major challenge is embedding the scene graph structure into vector representations for seamless integration into the encoder-decoder learning framework. Also, such representation should facilitate the sequence decoder to generate longer sequences. Therefore, in this paper, we introduce **S**parse **G**raph-to-**S**equence **T**ransformer (SGST) for embedding scene graph by understanding structured sparsity and then decoding it into the textual sequence. This approach builds upon Transformer (Vaswani et al., 2017) encoder-decoder architecture as Transformer based decoders (Radford et al., 2018; Yang et al., 2019b) have already shown their ability to decode longer sequences, however, they are less explored in encoding graph structures. Nevertheless, there has been some interest recently (Yun et al., 2019; Zhu et al., 2019), but, many methods proposed earlier to encode graphs into vector representation are mostly based on Graph Convolutional Networks (GCN) (Kipf & Welling, 2016). We hypothize that SGST is a more effective approach for our problem than GCN as it performs global contextualization of each vertex than focused portions in GCN (e.g., adjacent vertices) allowing direct modeling of dependencies between any two nodes without regard to their distance in the input graph. Furthermore, SGST incorporates sparse attention mechanism (Peters et al., 2019) in the self-attention of Transformer architecture allowing it to assign zero probabilities for irrelevant graph vertices or tokens in a sequence. This aids SGST to effectively encode graphs and decode longer sequences.

# 2. Vision-to-Sequence as Graph-to-Sequence Learning

A standard approach for vision conditioned language generation is using a vision-language pair $(v, y) \in (\mathcal{V}, \mathcal{Y})$. We first modify it as the graph-language pair i.e., $(g, y) \in (\mathcal{G}, \mathcal{Y})$, where $g = (g_1, g_2, ..., g_m)$ is the set of $m$ nodes denoting visual objects, relationships and attributes, while $y = (y_1, y_2, ..., y_n)$ is the target description with $n$ tokens. For learning, a language model decoder is conditioned on a graph encoder to learn the parameter $\Theta$ for estimating the conditional probability $P(y|g;\Theta)$ using a log likelihood as the objective function. The conditional probability $P(y|g;\Theta)$ can be factorized according to the chain rule of probability: $P(y|g;\Theta) = \prod_{t=1}^{n} P(y_t|y_{<t}, g;\Theta)$, where $y_{<t}$ denotes the preceding tokens before the index $t$.

# 3. Sparse Graph-to-Sequence Learning

## 3.1. Sparse Graph Transformer as Encoder

Our Graph Encoder is inspired by the self-attention use of the Transformer on the sequential data. It can be seen resembling GNN by replacing the token sequence as an unlabeled directed acyclic graph (DAG). To ensure, all scene graphs generated for images are unlabeled DAGs, we replace relations representing labels between vertices by new vertices. Further, the new vertices are connected with the object vertices such that the directionality of the former edge is maintained. We also introduce a global vertex to connect the entire graph.

Therefore, in the final graph $G = (V, A)$, $V$ embeds objects, relationships, attributes and the global vertex into dense vector representations (combined with their positional encodings), resulting in a matrix $\mathbf{V}^0 = [\mathbf{v}_i], \mathbf{v}_i \in \mathbb{R}^d$, given as input to the encoder. $A$ denote the adjacency matrix showing the connection between vertices. Now, each vertex representation $\mathbf{v}_i$ is self-attended with the *sparse graph multi-head attention* over the other neighbourhood vertices to which $v_i$ is connected in the $G$. We use an $N$-headed self attention setup, where $N$ independent attentions are calculated.

The self-attention in the Transformer is densely connected. Given $n$ query contexts and $m$ sequence items under consideration, attention computes, for each query, a weighted representation of the items i.e., *scaled dot-product attention* given in Equation 1.

$$\mathsf{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathsf{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \tag{1}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times d}$ contains representations of the queries, $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{m\times d}$ are the *keys* and *values* of the items attended over, and $d$ is the dimensionality of these representations. For multiple heads ($\mathsf{H}_i$), Att is calculated separately.

$$\mathsf{H}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathsf{Att}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{2}$$

However, we hypothize that adding sparsity to our graph Transformer encoder at different levels is beneficial. There are several advantages to it: (1) eliminate unnecessary vertices that are still taken into consideration to a certain extent for calculation of attention weights (Martins & Astudillo, 2016; Correia et al., 2019) (2) reduce memory and computational requirements with factorizations of the attention matrix (Child et al., 2019). We concentrate on the first scenario and present more details in the following.

**Sparse Graph Self-Attention** For $\mathcal{N}_i$, the neighborhood of $v_i$ in $G$, we compute the self-attention of a single head using each vertex $v_i$ with vertices $v_j$ in a single-hop using Equation 3.

$$\mathsf{Att}_G = \sum_{j\in\mathcal{N}_i} \beta_{ij}\mathbf{W}^V\mathbf{v}_j \tag{3}$$

where $\mathbf{W}^V \in \mathbb{R}^{d\times d}$ and $\beta_{ij}$ is given by Equation 4.

$$\beta_{ij} = \mathsf{Normalize}(\mathbf{v}_i, \mathbf{v}_j) \tag{4}$$

Further, to introduce sparse attention, we modify the Normalize by simply replacing softmax with $\alpha$-entmax in the attention heads. That is, softmax in Equation 1 is modified as follows.

$$\mathsf{Normalize}(\mathbf{q}_i, \mathbf{k}_j) = \alpha\text{-}\mathsf{entmax}(\mathbf{z}) \tag{5}$$

The $\alpha$-$\mathsf{entmax}(\mathbf{z})$ is given by Equation 6 (Blondel et al., 2019) and $\mathbf{z}$ is provided by Equation 7.

$$\alpha\text{-}\mathsf{entmax}(\mathbf{z}) = \mathsf{ReLU}[(\alpha-1)\mathbf{z} - \tau\mathbf{1}]]^{\frac{1}{\alpha-1}} \tag{6}$$

$$\mathbf{z} = \frac{\mathbf{W}_i^Q \mathbf{q}_i (\mathbf{W}^K \mathbf{k}_j)^\top}{\sqrt{d}} \tag{7}$$

where $\mathbf{q}_i, \mathbf{k}_j$ are query and key of $v_i$ and $v_j$ respectively. $\tau$ is unique threshold and $\mathbf{1}$ is vector of all ones. For the experiments, following (Peters et al., 2019), we fixed $\alpha$=1.5 and also used sparse attention with a different learned $\alpha = 1 + \mathsf{sigmoid}(\mathsf{att\_scalar}) \in [1, 2]$ for the each attention head (Correia et al., 2019). We tie all $\alpha$ values between encoder-decoder and $\mathsf{att\_scalar} \in \mathbb{R}$ is a parameter per attention head.

**Encoder** Output ($\mathsf{Att}_G^N$) of $N$ attention heads is concatenated and added to $\mathbf{v}_i$ to attain $\hat{\mathbf{v}}_i$. Further, $\hat{\mathbf{v}}_i$ is passed through different computations in the encoder to transform into $\mathbf{h}_i^{enc}$ given as follows.

$$\mathbf{h}_i^{enc} = \mathsf{LayerNorm}(\tilde{\mathbf{v}}_\mathbf{i} + \mathsf{LayerNorm}(\hat{\mathbf{v}}_i)) \tag{8}$$

$$\tilde{\mathbf{v}}_i = \mathsf{TransFunction}(\mathsf{LayerNorm}(\hat{\mathbf{v}}_i)) \tag{9}$$

Where $\mathsf{TransFunction}(\mathbf{x})$ is a two layer feedforward network with a non-linear transformation between layers. To increase the depth of network, blocks are stacked $\mathsf{L}$ times, with the output of layer $l-1$ taken as the input to layer $l$, so that $\mathbf{v}_i^l = \mathbf{h}_i^{enc(l-1)}$. Stacking multiple blocks allows information to propagate through the graph.

### 3.2. Sparse Sequence Transformer as Decoder

Our sequence decoder is built on the principle of sequential Transformer decoder. It predicts the next token $y_t$ given all the previous tokens $y_{<t} = y_1, ..., y_{t-1}$. Context attention $\mathbf{C}_{att}$ is computed by performing *sparse context attention* for single head over the output ($\mathbf{h}_i^{enc(l)}$) of the graph encoder and decoder hidden state ($\mathbf{h}_t^{dec}$) at each timestep $t$ given as follows.

$$\mathsf{C}_{att} = \sum_{j \in \mathcal{N}_i} \gamma_j \mathbf{W}^G \mathbf{h}_j^{enc(l)} \tag{10}$$

where $\mathbf{W}^G \in \mathbb{R}^{d\times d}$ and $\gamma_j$ is given by Equation 11 and is further modified according to Equation 5.

$$\gamma_j = \mathsf{Normalize}(\mathbf{h}_t^{dec}, \mathbf{h}_j^{enc(l)}) \tag{11}$$

We also modify the masked self-attention into masked *sparse sequence self-attention* in the similar manner as *sparse graph self-attention*. Due to space limitation, we present overall architecture in the supplemental material.

## 4. Training and Inference

We use Transformer with $\mathsf{L} = 6$ layers and $\mathsf{H} = 8$ heads both for encoder and decoder. To optimize, we use Adam (Kingma & Ba, 2014) with $\beta 2$=0.98. Input embeddings and hidden size is set to 512 and batch size of 2048. The $\mathsf{TransFunction}$ has an intermediate size of 2048. All models are trained between 8 and 12 epochs. During inference, beam search is used with beam size = 5.

## 5. Experimental Setup

### 5.1. Dataset and Evaluation Metrics

To evaluate our proposed approach, we used the image paragraph dataset (Krause et al., 2017) containing images aligned with the textual sequences that are longer than an usual sentence-level caption (e.g., MSCOCO (Lin et al., 2014)) i.e., paragraph. The dataset contains 19,551 image-paragraph pairs. On average, each paragraph has 67.5 words and each sentence in the paragraph consists of 11.91 words. Following the settings of (Krause et al., 2017), we split the dataset into 14,575 images for training, 2,487 for validation and 2,489 for testing. We evaluated image paragraph generation with the widely used language generation metrics such as BLEU (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), and CIDEr (Vedantam et al., 2015).

### 5.2. Image Scene Graph Generation

Since all images from the image-paragraph dataset are part of the Visual Genome (VG) dataset (Krishna et al., 2017). We could have directly used the ground truth (GT) scene graph annotations, i.e., objects and their pairwise relationships along with attributes. However, to show that our method can be applied to any image, we generated a scene graph for the images using a trained model that can do object proposal detection (to detect and classify objects), relationship classification (classify relationships between objects), and the attribute classification. To overcome the noisy annotations present in the dataset, similar to (Yang et al., 2019a) we filter and keep 305 objects, 103 attributes, and 64 relationships to train our detector and classifiers.

To be specific, to train the object detector we used Faster-RCNN (Ren et al., 2015) for extracting 36 RoI features in similar manner as (Anderson et al., 2017). The detected objects are further used as the input to the relationship classifier (Zellers et al., 2018) to predict a relationship between two objects and also attribute classifier to attain top-3 attributes per object. Now, for each image using the predicted objects, relationships, and attributes, an image scene graph can be built.

## 6. Results and Discussion

Table 1 presents the automatic metrics comparison, while Table 2 shows the language analysis performed to know the choice of the vocabulary used by our models in paragraph generation. From the automatic metrics presented in the Table 1 we observe that the BLEU score which is dependent on word overlap has only seen an improvement of around 8% when SGST ($\alpha$-entmax) (w/ GT scene graphs) compared against (Wang et al., 2019), while CIDEr that cares about overall semantics of the generated paragraph has a gain of 13.3%. Similarly, when we try to analyze the impact of Table 2, we observe that it depicts the usage of grammar in the generated paragraphs. In contrast to single sentence generation, the paragraphs should have a smooth transition between sentences having high coherence. In general, pronouns capture such a transition well while verbs provide the actions observed in the scene. We observe that our proposed models produce more verbs and pronouns while generating concise paragraphs (i.e., minimal average length).

Figure 2 shows an example paragraph generated by the baseline SGST (softmax) and best approach SGST ($\alpha$-entmax) using both GT and the generated scene graph. It is interesting to observe that paragraph generated by both models are highly relevant to the image. However, SGST ($\alpha$-entmax) generated a more coherent and brief paragraph with lesser words.

| Method | C | M | B-4 |
|---|---|---|---|
| (Karpathy & Fei-Fei, 2015) | 11.06 | 12.82 | 7.71 |
| (Krause et al., 2017) | 13.52 | 15.95 | 8.69 |
| (Liang et al., 2017) | 20.36 | 18.39 | 9.21 |
| (Chatterjee & Schwing, 2018) | 20.93 | 18.62 | 9.43 |
| (Wang et al., 2019) | 25.15 | 18.82 | 9.67 |
| w/ generated scene graphs | | | |
| SGST (softmax) | 25.22 | 18.95 | 9.84 |
| SGST (1.5-entmax) | 25.46 | 19.01 | 9.86 |
| SGST ($\alpha$-entmax) | 26.01 | 19.16 | 10.05 |
| w/ GT scene graphs | | | |
| SGST (softmax) | 26.89 | 19.16 | 10.03 |
| SGST (1.5-entmax) | 27.51 | 19.20 | 10.13 |
| SGST ($\alpha$-entmax) | **28.50** | **19.25** | **10.45** |

*Table 1.* Performance of our proposed methods in comparison with other state-of-the-art using CIDEr (C), METEOR (M), and BLEU-4 (B-4) measures on the image paragraph dataset. All values are reported as percentage (%).

| Method | AvgLen (words) | StdDev (words) | N | V | P |
|---|---|---|---|---|---|
| Baseline | 70.47 | 17.67 | 24.77 | 13.53 | 2.13 |
| w/ generated | | | | | |
| (softmax) | 57.33 | 14.01 | 26.00 | 14.33 | 3.67 |
| (1.5-entmax) | 59.66 | 18.71 | 27.00 | 15.00 | 3.33 |
| ($\alpha$-entmax) | 54.66 | 11.71 | 25.33 | 15.00 | 3.33 |
| w/ GT | | | | | |
| (softmax) | 62.00 | 14.93 | 27.00 | 15.67 | 4.00 |
| (1.5-entmax) | 61.33 | 16.19 | 27.33 | 14.67 | 4.00 |
| ($\alpha$-entmax) | 56.33 | 11.15 | 25.67 | 15.67 | 3.00 |

*Table 2.* Language Analysis is performed to comprehend the choice of vocab used by our models in generation. Regions-Hierarchical model from (Krause et al., 2017) is the baseline, while AvgLen and StdDev denote the average number of words in the paragraph and standard deviation of them respectively. N, V and P are Nouns, Verbs and Pronouns observed in the paragraph.

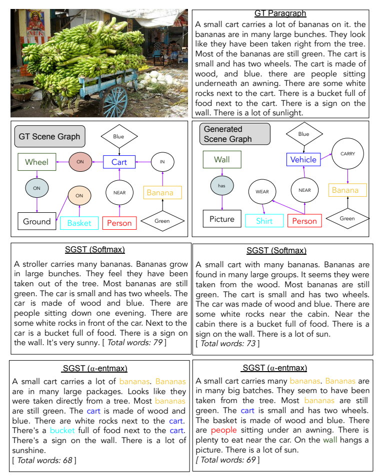


*Figure 2.* Qualitative results of generated paragraphs (only partial graph is shown w/o global vertex).

## 7. Conclusion and Future Work

We have presented SGST, treating vision-to-sequence as graph-to-sequence learning. We encode images into scene graphs and condition on them for long textual sequence generation. Our experiments show that our proposed approach can effectively encode scene graphs for generating paragraphs. In future, we plan to investigate the impact of leveraging graph reasoning while encoding scene graph constituents into vectors. Further, we also aim to find the impact of sparse attention on the attention heads and compare the performance with GCN encoders.

## References


Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Beck, D., Haffari, G., and Cohn, T. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

Blondel, M., Martins, A., and Niculae, V. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 606–615, 2019.

Chatterjee, M. and Schwing, A. G. Diverse and coherent paragraph generation from images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 729–744, 2018.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Correia, G. M., Niculae, V., and Martins, A. F. Adaptively sparse transformers. 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)*, pp. 2174–2184, 2019.

Denkowski, M. and Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 376–380, 2014.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., and Fei-Fei, L. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015.

Kennedy, W. G., Bugajska, M. D., Marge, M., Adams, W., Fransen, B. R., Perzanowski, D., Schultz, A. C., and Trafton, J. G. Spatial representation and reasoning for human-robot collaboration. In *AAAI*, volume 7, pp. 1554–1559, 2007.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., and Hajishirzi, H. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342*, 2019.

Krause, J., Johnson, J., Krishna, R., and Fei-Fei, L. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 317–325, 2017.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Liang, X., Hu, Z., Zhang, H., Gan, C., and Xing, E. P. Recurrent topic-transition gan for visual paragraph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3362–3371, 2017.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pp. 1614–1623, 2016.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Peters, B., Niculae, V., and Martins, A. F. Sparse sequence-to-sequence models. *arXiv preprint arXiv:1905.05702*, 2019.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

Song, L., Zhang, Y., Wang, Z., and Gildea, D. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vedantam, R., Lawrence Zitnick, C., and Parikh, D. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Wang, J., Pan, Y., Yao, T., Tang, J., and Mei, T. Convolutional auto-encoding of sentence topics for image paragraph generation. *IJCAI*, 2019.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057, 2015.

Yang, X., Tang, K., Zhang, H., and Cai, J. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10685–10694, 2019a.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019b.

Yao, T., Pan, Y., Li, Y., and Mei, T. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 684–699, 2018.

Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J. Graph transformer networks. In *Advances in Neural Information Processing Systems*, pp. 11960–11970, 2019.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, 2018.

Zhu, J., Li, J., Zhu, M., Qian, L., Zhang, M., and Zhou, G. Modeling graph structure in transformer for better amr-to-text generation. 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)*, pp. 5462–5471, 2019.