
Visual Analysis of Abstractive Document Summarization

Dujian Ding
dujian@cs.ubc.ca

Raymond Li
raymond1@cs.ubc.ca

Abstract

Recent state-of-the-art abstractive summarization models have been dominated by the usage of pre-trained transformer-based encoders in a sequence-to-sequence fashion. As saliency maps are often a common technique for the explanation and interpretation of AI models, we want to apply similar techniques to abstractive summarization models in this project. We propose a visual framework for investigating the relevancy paid to source tokens at each output step, as well as performing experiments in assessing the interpretation methods quantitatively. We adopted a summarization model based on the popular BERT encoder and discuss the interpretation results for the three methods: attention, layer-wise relevance propagation (LRP), and input gradients.

1 Introduction

Summarization has and continues to be a hot research topic in the NLP and more recently the ML domain. Concisely, text summarization is the task of outputting a fluent and coherent summary while preserving the key content of the original documents or articles. Its application ranges from social media monitoring, market forecasting, legal document analysis and many more. Generally, there are two main approaches in summarization: extractive and abstractive. The former method uses probabilistic models to identify key sentences and phrases before generating them verbatim to construct a concise summary. While the abstractive method mainly deals with the utilization of natural language generation (NLG) models to output an abstract conditioned on the original text.

As recent abstractive models have nearly all been variants of the attention-based sequence-to-sequence models [Bahdanau et al., 2014], one common built-in method is the visualization of attention scores as explanation of the output. This, however, has been criticized in recent studies specifically in the work by [Jain and Wallace, 2019], where they concluded that attention vectors are not necessarily useful for explanation in a transparent sense, as attention weights can often be substantially changed without altering model predictions. However, their experiments were conducted on RNNs based on LSTMs [Hochreiter and Schmidhuber, 1997], where in contrast to transformers [Vaswani et al., 2017], the information received by the decoder and the importance it “allocates” to each attention state might be very low. In this project, we look at attention in a slightly different manner, instead of solely focusing on explanation, we are also interpreting whether the information learned by the model are relevant and useful.

Visual analysis of NLP models has also been a rapid growing sub-field at the intersection of information visualization and text analytics. Many techniques and frameworks have been introduced in recent years to help users and researchers with a wide range of tasks. With the recent advances of deep neural models in NLP, there has been a growing demand in such tools due to black-boxes nature of such models during predictions. Various tools have been proposed in different domains, such as machine translation [Strobel et al., 2018], adversarial text generation [Laughlin et al., 2019], question answering [Lee et al., 2019], as well as general frameworks for a variety of prediction tasks [Wallace et al., 2019]. To help facilitate the visualization of different saliency maps, we developed a

visual framework for the specific purpose of document-level summarization that allows the user to easily interact with individual tokens to obtain saliency maps a different steps of the output. We were able to observe several interesting features associated with our model using our visual framework for which we will discuss in a later section.

A large portion of the current state-of-the-art results posted for abstract summarization models have all been achieved by transformer based encoder-decoder models. Due the recent trend for large-scale pre-training of language models popularized by BERT [Devlin et al., 2018], many recent summarization models have adapted for a similar approach. In our experiments, we use a recent BERT-based summarization model [Liu and Lapata, 2019] that reported state-of-the-art results on the CNN/Daily Mail dataset. The model consists of the pre-trained BERT encoder and a 6-layer transformer decoder fine-tuned on the downstream summarization dataset with different optimizers for the encoder and decoder. Although similar proposed models with different pre-training objectives have reported better quantitative results [Lewis et al., 2019] [Zhang et al., 2019], we decided on the current model due to popularity of BERT. To summarize, our contribution is two-fold. First, we propose a visual framework for developers to interactively analyze different data samples using an overview-detail view. Second, we perform experiments with three-interpretation methods (attention, LRP, and input gradients) and use a quantitative measure of extractive summarization to measure the results of the saliency map generated by each method.

2 Background

In this section, we briefly discuss three building blocks of the abstractive summarization model and interpretation methods adopted in this project, which are Transformer, BERT, and LRP.

2.1 Transformer

Transformer [Vaswani et al., 2017] is a sequence-to-sequence (seq2seq) model with an encoder-decoder structure. Distinguished from the RNN-based architecture (Bahdanau et al. [2014]), Transformer is implemented with stacked self-attention and feed-forward neural network layers. Instead of focusing on the input-to-output dependency, self-attention allows the network to learn dependencies between different positions within the input sequence. In practice, the self-attention is computed through a query matrix Q , a key matrix K , and a value matrix V :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where d_k is the dimensionality for queries and keys. Besides, Transformer utilizes the multi-head attention to jointly learn from different representation subspaces for different positions. Typically, multi-head attention is achieved by first concatenating multiple attention matrices, and then projecting the concatenation to the final values through parameter matrices W^O and W_i :

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_n)W^O \quad (2)$$

$$\text{where } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

In Transformer, the encoder comprises of multiple layers, where each layer consists of two components: 1) the aforementioned self-attention, 2) and one feed-forward neural network. The input embedding first is augmented by positional embedding and then flows into the self-attention and feed-forward network in order, whose outputs are fed into the next layer of the encoder or the decoder. In the decoder side, additional encoder-decoder attention layer is added to capture the dependency between input and output sequences. The final outputs of the decoder is then mapped to a probability distribution on the next possible token through a softmax layer.

2.2 BERT

BERT (Devlin et al. [2018]), which stands for Bidirectional Encoder Representations from Transformers, is a language representation model pre-trained on a large corpus of 3300 million of words. Before BERT, researchers have realized the importance of contextual meaning and various techniques have been applied to train sentence representations: sequential autoencoders (Hill et al. [2016]), the left-to-right generation model (Kiros et al. [2015]), and the shallow concatenation of left-to-right and

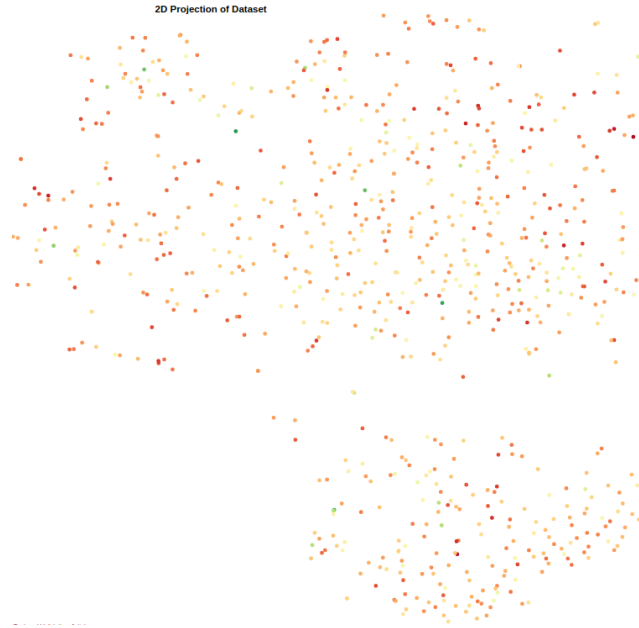


Figure 1: 2D projection visualization for t-SNE

right-to-left representations (Peters et al. [2018]). BERT further extends the idea of learning word embedding from the contexts by applying a masked language modelling. In the original paper, the authors masked some percentage of the input tokens at random, and trained the model by asking it to predict the masked words. With the masked language modelling, BERT successfully captures the bidirectional contextual information for different positions without letting each word indirectly "see itself". In addition, BERT is also capable to understand the relationship between sentences, given it was originally pretrained for a *next sentence prediction* task, which can be generated from any monolingual corpus.


2.3 LRP

Layer-wise Relevance Propagation (LRP; Bach et al. [2015]) was originally proposed to visualize the contributions of single pixels to predictions of the image classifier. By recursively computing relevance from the output layer to the input layer, LRP is demonstrated to be useful in unravelling the inference process of neural networks and has been adopted in recent work on neural network analysis (Voita et al. [2019]). The intuition behind LRP is that, each neuron of the network is contributed by neurons in the previous layer, and the total amount of contributions for each layer should be a constant during back-propagating, which is called the *conservation principle*. LRP offers flexibility to design propagation rules to explain various deep neural networks, one example propagation rule is shown as follows (Montavon et al. [2019]),

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k \quad (4)$$

where R_j and R_k are relevance scores of two neurons in consecutive layers, a_j is the respective activation for neuron j , and w_{jk} is the weight between neuron j and k .

3 Our Work

In this section, we describe the implementation of our visual framework  as well as documenting the findings of our experiments with the three interpretation methods: attention, LRP, and input gradients.

3.1 Visual Framework

To provide model developers and user with an easy-to-use framework in visualizing the results and saliency maps produced by our summarization model, we implemented an overview-detail view that facilitates user-interaction. In the main view, the framework provides an overview of the test set where individual marks represents an article sample projected onto a 2D plane, and color encodes the ROUGE score from the output of summarization model. When a user clicks on any individual article, a pop-up window will display a article-to-summary view for visualizing the model output as well as the saliency maps generated from the various interpretation techniques. We also implemented a secondary linked view, where user can select an area on the projection space and the vocal distribution of the data samples are rendered on a bar chart histogram. However, we found no correlation between the vocab distribution and the samples within different clusters, and therefore removed this view for the demo of this project.

To obtain the features used for our 2D projection overview, we extract the output from the BERT encoder and average across each steps of the input to obtain a 768-dimensional vector. We also implemented BOW and TFIDF representations, but removed them from the demo as we found poor visual results due to the lack of semantic context. We then considers three dimension reduction methods for projection: PCA, SVD and t-SNE [Goodfellow et al., 2016], and provide a selection menu in the main view that allows the users to select the projection method (Figure 1). From visual results, we found the projections obtained from PCA and SVD formed a single dense elliptical cluster with high variance along a single axis, while t-SNE projections were much sparser where there were several clusters with different densities. This is due to the fact the t-SNE is optimizing for two different distributions in both low and high dimension space, which leads to better visualization of different clusters. We found that nearby data samples often correlates to similar topics, this is similar to the visualization from variational autoencoders used in topic modelling as shown in the work by [Wang et al., 2019]. However, we did not observe any trivial patterns among samples in the same clusters, this is due to the lack of Gaussian constraints placed on the hidden states that encourages interpolation in the latent space.

The article-to-summary view can be activated when a user clicks on any data samples from the main view. In the pop-up window, the source and prediction tokens are displayed in vertically "side-by-side" boxes. Note that since byte pair encoding [Sennrich et al., 2015] is used by the BERT tokenizer to deal with rare and unseen words, the "##" prefix implies that the token is sub-word of the preceding token. When a user clicks on a token from either the source or predict, the corresponding saliency attributed to that token is visualized using a heatmap in the opposing box where color intensity encodes how much saliency is attributed. For a source token, this can be interpreted as the amount of saliency that the source token is responsible for generating. While for a prediction token, this is interpreted as the saliency map used for generating that token. We use a drop-down menu that allows the user to select the saliency map generated from different methods (Figure 2). We represent the heatmap as a $n \times m$ vector for a data sample with n source tokens and m predicted tokens. There are two features that we forgot to implement but are useful for the context of our visualization: the visualization of saliency for tokens in the same domain representing the self-attention, and the concatenation of sub-words for a word-level saliency map using the approach described in the following section.

3.2 Experiments

3.2.1 Saliency Maps

We generate our saliency maps using three methods: attention, LRP and input gradients. For attention, we extract the attention vectors after the softmax layer for the encoder-decoder attention in the decoder layers of each attention head to obtain a $l \times h \times n \times m$ vector for l layers and h heads. In our visualization, we allow the selection of individual attention heads as well as the average and sum of the attention heads across each layers. In LRP, we only need to consider fully-connected linear layers as relevance conservation ensures that relevance is just inherited in a one-to-one manner for activation layers such as softmax and gelu. We use the epsilon rule for all linear layers using the following formula for neuron i of layer l from neuron j of layer $l + 1$ from [Bach et al., 2015]:

$$R_{i \leftarrow j}^{l, l+1} = \frac{w_{i \rightarrow j}^{l, l+1} z_i^l + \frac{\epsilon \text{sign}(z_j^{l+1}) + b_j^{l+1}}{D_i}}{z_j^{l+1} + \epsilon \text{sign}(z_j^{l+1})} R_j^{l+1} \quad (5)$$

Visualization of Article-Summary Saliency Map



Figure 2: Saliency map visualization for individual data samples

Where $w_{i \rightarrow j}^{l, l+1}$ is weight unit for the corresponding neurons, z_i^l is the activation value of neuron i of layer l , D_l is the number of dimension at layer l and R_j^{l+1} is the relevance propagated from the previous layer. Finally, we obtain the gradients w.r.t the output token as

$$\frac{\partial y_j^k}{\partial x_i} \quad (6)$$

Where x_i is the average embeddings values of input token i , and y_j^k is the prediction of token $k \in [0, \text{vocab_size} - 1]$ at time-step j .

3.2.2 Quantitative Results

To assess the effectiveness of our saliency map, we propose a quantitative measure by extracting the most salient sentences from the input to form an extractive summary and evaluating it against the prediction generated by our model. This is under the intuition that the quality of the saliency map could be determined by whether the tokens attended to have a high similarity with the model’s predictions, where high quality saliency map attributes high values to sentences that are most relevant for the summary. The ROUGE score of the extracted sentences against the model’s prediction is listed in Table 1. This is in contrast to the method proposed by [Arras et al., 2017] where “important” words were removed from the input text and measure the changes in the resulting predictions. We choose to perform extraction on a sentence level due to the fact that our summarization model [Liu and Lapata, 2019] inserts external [CLS] tokens to represent the start of each sentence segments. We found that several attention heads were specialized in attending to such segmentation tokens, and therefore utilized them in computing the average sentence saliency across all prediction steps.

Since we want to evaluate saliency at a word-level, we convert token-level saliency maps to word-level saliency maps. For attention to a split-up word, we sum up the attention weights over its tokens. While for attention from a split-up word, we take the mean of the attention weights over its tokens [Clark et al., 2019b]. For each individual decoder layers, we take both the max and average of each saliency value across all heads as well as the average and max across all layers and heads and report the results for each method. From the results, we can see that the attention from layer 3 and 4 of the decoder attended to sentences with the highest similarity with the generated output. LRP significantly under-performed against all other methods as they attended mostly to segmentation tokens and the variation in saliency map varied little across different prediction steps. This is consistent with the results obtained from [Tuckey et al., 2019], where they generated LRP saliency heatmap across each prediction step using the LSTM-based Pointer-Generator Network [See et al., 2017].

Table 1: Rouge scores of extracted salient sentences against model predictions

Layer	Approach	Unweighted			Weighted		
		R1	R2	RL	R1	R2	RL
all	Attention_Avg	71.08	63.60	70.25	69.44	61.68	68.52
all	Attention_Max	67.70	59.42	66.68	66.43	57.94	65.33
1	Attention_Avg	56.86	45.18	55.03	53.80	41.44	51.81
1	Attention_Max	64.26	54.82	62.95	64.16	54.81	62.86
2	Attention_Avg	61.81	51.45	60.32	59.68	48.87	58.10
2	Attention_Max	65.19	55.85	63.96	64.65	55.23	63.38
3	Attention_Avg	69.85	61.99	68.95	69.45	61.56	68.52
3	Attention_Max	67.78	59.47	66.76	67.37	59.03	66.33
4	Attention_Avg	69.16	61.12	68.22	68.95	60.91	67.98
4	Attention_Max	67.27	58.79	66.22	66.69	58.17	65.61
5	Attention_Avg	68.04	59.67	67.02	68.01	58.72	66.98
5	Attention_Max	66.42	57.68	65.30	66.11	57.40	64.97
6	Attention_Avg	61.41	51.4	59.97	60.94	50.94	59.49
6	Attention_Max	59.48	49.04	57.92	59.32	48.96	57.77
	LRP	36.85	21.08	33.78	36.85	21.08	33.78
	Input Gradients	30.42	11.70	27.18	31.65	13.11	28.39

Our unweighted approach assumes each prediction step contributes equally to the final input saliency by averaging the saliency map across all output tokens (obtaining a n dimensional vector from a $n \times m$ vector by averaging across all m). However, since the transformer decoder uses self-attention for attending to previous generated output tokens, we hypothesize that not all output tokens are equally relevant for interpretation. To test our hypothesis, we obtain a weighting vector for each prediction step using the self-attention vector extracted from the decoder. The self-attention vector is averaged across all layers and heads and normalized by the amount of steps each token was attended to. This is due to the fact that decoder self-attention can only attend to tokens that it has generated at previous time steps, and normalizing the attention vector help us obtain the average relevance of each output token with respect to future predictions. As the last token was never attended to, we take the mean of all previous tokens to be the values for the last EOS token generated by the decoder to signify the end of summary. Additionally, we follow the same approach for the first generated token as we found that it had an unreasonable high attention value across all data samples.

From the results the max across all heads had little variance across different layers (with the exception of the final layer). This could be due to the fact that the few heads e dedicated for attending to matching tokens in the input will dominate, resulting in similar saliency maps across different layers. However, since more specialized heads are observed in the upper layers, averaging the saliency across all heads will result in vastly different saliency maps for different layers. The middle layers yielded the best performance with respect to averaging the attention heads. We also note that our weighted approach using decoder self-attention actually resulted in poorer quality saliency maps with the exception of input gradients. We hypothesize that this is due to the large amount of noise introduced while averaging the self-attention across all layers and heads before naively applying it to the saliency map, this only provided additional information to the input gradients method as it had poor quality to begin with. We think that further investigation into the usage of self-attention as weights for the saliency maps might yield improvements across different methods. Lastly, we note that although gradients-based methods are more suitable for the explanation of model predictions, they result in poor saliency maps for the purpose of general interpretation. We will leave the experiments with individual attention heads such as pruning as possible follow-up work.

3.2.3 Qualitative Observations

While visualizing the saliency maps using our visual framework, we make some interesting qualitative observations. For the attention from the first two layers, prediction tokens will almost exclusively attend to the [CLS] token marking the beginning of a sentence. Some heads will attend to some tokens earlier in the article, while other attends to later sentences. Although there will be one or two dedicated attention heads that are exclusive for attending to the identical token when present in the

input, while little or no attentions are attributed to the rest. As we get to higher layers, less heads be dedicated to attending to [CLS] tokens, as the variance in the attention distribution will also increase, as saliency becomes sparser across input tokens. Positional words such as "in" and "to" will begin attending to variety of tokens that are uncorrelated to the context that it's generated. In the upper layers, different heads begin to diverge in different specializations. Some heads were dedicated for attending to the subsequent context within the input (ex. prediction "start in Europe", "start" token attends to "in Europe" within input), while others will attend to antecedent tokens that entails the predictions. This is also observed in the work by [Clark et al., 2019a], where self-attention heads are observed to be specialized in a variety of different tasks.

4 Related Work

There has been substantial recent work focusing on visualization and interpretability of neural networks. [Ribeiro et al., 2016] proposed an approach to give local interpretable model-agnostic explanations (LIME) to individual predictions generated by any classifiers. [Ding et al., 2017] attempted to visualize and interpret neural machine translation through layer-wise relevance propagation. In [Wallace et al., 2019], an open-source toolkit was implemented to facilitate the evaluation of interpretation methods across a wide range of NLP topics and models, which continues to evolve nowadays.

More recently, Transformer [Vaswani et al., 2017] and BERT [Devlin et al., 2018] emerged to be a driving force in the research of natural language processing. One line of work focuses on why either Transformer or BERT achieving the state-of-the-art performance on a wide range of tasks, and attempts to reduce the model size without significantly impacting performance. In [Vig, 2019], the authors characterized the dependency between input and output sequences of neural networks by visualizing the intermediate attention in attention-head level, model level, and neuron level. This work was demonstrated on both BERT and Open-AI GPT-2, and shed light on how attention transmitting through layers to influence the final outputs. Besides, several work [Voita et al., 2019] [Clark et al., 2019b] [Michel et al., 2019] were inspired by the functionality of different attention heads. Particularly, it was found that attention heads do not equally contribute to the overall model performance, and a vast majority of heads can be pruned without seriously affecting performance. In addition, those functional attention heads were demonstrated to correspond well to linguistic notions and can be further categorized into three types: positional heads, syntactic heads, and heads responsible to rare words recognition.

Our work can be regarded as an extension to aforementioned interpretation analysis on Transform and BERT. In addition to visualizing attention heads for different layers, we also applied several aggregation strategies to compute the dependency between each output token and its corresponding input sequence, which can be used to improve model performance on natural language processing tasks, e.g., text summarization.

5 Conclusion

In this project, we implemented a visual framework for analyzing abstract document summarization, and performed various experiments in identifying the best technique for generating saliency maps based on a metric that measures the similarity between the most salient sentences of the input with our model's predictions. Evidence from our experiments with transformer-based sequence-to-sequence model. We concluded while gradients based methods such as LRP and input gradients may be better suited for a direct explanation of the model's predictions, attention is better at visualizing the learned saliency of the model w.r.t to its predictions.

References

L. Arras, G. Montavon, K.-R. Müller, and W. Samek. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5221. URL <https://www.aclweb.org/anthology/W17-5221>.

- S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*, 10(7), 2015.
- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of bert’s attention. *CoRR*, abs/1906.04341, 2019a. URL <http://arxiv.org/abs/1906.04341>.
- K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*, 2019b.
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Y. Ding, Y. Liu, H. Luan, and M. Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1106. URL <https://www.aclweb.org/anthology/P17-1106>.
- I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- F. Hill, K. Cho, and A. Korhonen. Learning distributed representations of sentences from unlabelled data. *CoRR*, abs/1602.03483, 2016. URL <http://arxiv.org/abs/1602.03483>.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- S. Jain and B. C. Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.
- R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015. URL <http://arxiv.org/abs/1506.06726>.
- B. Laughlin, C. Collins, K. Sankaranarayanan, and K. El-Khatib. A visual analytics framework for adversarial text generation. *arXiv preprint arXiv:1909.11202*, 2019.
- G. Lee, S. Kim, and S.-w. Hwang. Qadiver: Interactive framework for diagnosing qa models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9861–9862, 2019.
- M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.
- Y. Liu and M. Lapata. Text summarization with pretrained encoders. 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 3721–3731, 2019.
- P. Michel, O. Levy, and G. Neubig. Are sixteen heads really better than one? *CoRR*, abs/1905.10650, 2019. URL <http://arxiv.org/abs/1905.10650>.
- G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_10. URL https://doi.org/10.1007/978-3-030-28954-6_10.
- M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018. URL <http://arxiv.org/abs/1802.05365>.
- M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL <http://arxiv.org/abs/1602.04938>.

- A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.
- R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- H. Strobel, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018.
- D. Tuckey, K. Broda, and A. Russo. Saliency maps generation for automatic text summarization. *arXiv preprint arXiv:1907.05664*, 2019.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL <http://arxiv.org/abs/1706.03762>.
- J. Vig. Visualizing attention in transformer-based language representation models. *CoRR*, abs/1904.02679, 2019. URL <http://arxiv.org/abs/1904.02679>.
- E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *CoRR*, abs/1905.09418, 2019. URL <http://arxiv.org/abs/1905.09418>.
- E. Wallace, J. Tuyls, J. Wang, S. Subramanian, M. Gardner, and S. Singh. Allennlp interpret: A framework for explaining predictions of nlp models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 2019. doi: 10.18653/v1/d19-3002.
- W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, and L. Carin. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*, 2019.
- J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*, 2019.