Highlights of Attention Mechanisms for Model Interpretability

Khalil Mrini[1], **Franck Dernoncourt**[2], **Trung Bui**[2], **Walter Chang**[2], **and Ndapa Nakashole**[1]

[1] University of California, San Diego, La Jolla, CA 92093

khalil@ucsd.edu, nnakashole@eng.ucsd.edu

[2]Adobe Research, San Jose, CA 95110

{dernonco, bui, wachang}@adobe.com

1 Introduction

Neural networks have advanced the state-of-the-art in natural language processing (Wu et al., 2019; Joulin et al., 2017). Their performance has made their use ubiquitous, but their lack of interpretability has been a long-standing issue (Li et al., 2016).

Attention-based models offer a way to improve performance without sacrificing interpretability. Attention mechanisms were first introduced for machine translation (Bahdanau et al., 2014; Luong et al., 2015), and have since been extended to text classification (Yang et al., 2016), natural language inference (Chen et al., 2016) and language modeling (Salton et al., 2017).

Self-attention and transformer architectures (Vaswani et al., 2017) are now the state of the art in language understanding (Devlin et al., 2018), extractive summarization (Liu, 2019), semantic role labeling (Strubell et al., 2018) and machine translation for low-resource languages (Rikters, 2018; Rikters et al., 2018).

In this short survey, we first explain attention mechanisms and compare their interpretability in machine translation and text classification. Then, we explore self-attention and highlight the limits of its interpretability. Finally, we provide alternatives and challenges from the literature to attention for model interpretability, and outline suggestions for future work.

2 Attention Mechanisms

Cho et al. (2014) and Sutskever et al. (2014) introduce the RNN Encoder-Decoder architecture for machine translation. Bahdanau et al. (2014) augment the decoder part by introducing an attention mechanism to form a weighted context vector. The differences between the architectures are shown in Figure 1.

Bahdanau et al. (2014) define the attention weight of the i-th target language word w_i^t with regard to the j-th source language word w_j^s as:

$$\alpha(w_i^t, w_j^s) = \text{softmax}(a(h_{i-1}^t, h_j^s)) \quad (1)$$

where h_{i-1}^t is the output of the $(i-1)$-th cell of the decoder RNN, h_j^s is the output of the j-th cell of the encoder RNN, and a is the alignment model between them.

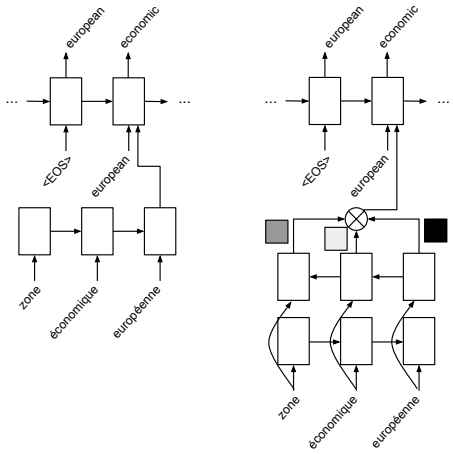

Figure 1: Comparison of the Decoder architectures of (a) Cho et al. (2014); Sutskever et al. (2014) and (b) Bahdanau et al. (2014). The colored squares on the right are the corresponding cells on the attention heatmap, with black being 0 and white being 1.

Yang et al. (2016) extend this attention mechanism to document classification. The architecture of their Hierarchical Attention networks consists of two levels: sentences and words. Each level is a bidirectional GRU encoder, followed by an attention mechanism. A sentence (resp. a document) is then an attention-weighted combination of GRU outputs of words (resp. sentences). Accordingly, this model is able to give an attention weight to each component of the document.

3 Self-Attention

Vaswani et al. (2017) introduce an encoder-decoder architecture. Each cell in the model contains a multi-head self-attention layer. For a sentence of length L, self-attention attributes attention weights to each word $w_1, w_2, ..., w_L$ in the sentence, with regard to a head: a given word w_i of the same sentence. As such, this process is *multi-head* when it is repeated with all words of the sentence taking turns at being the self-attention head.

More formally, given a head word w_{head} and a word w_i of the same sentence, we compute the corresponding attention weight as follows:

$$\alpha(w_{head}, w_i) = \text{softmax}\left(\frac{q_{head} \cdot k_i}{\sqrt{d}}\right) v_i \quad (2)$$

where q_{head} is the query vector of w_{head}, k_i and v_i are respectively the key vector and the value vector of w_i, and d is the dimension of both the query and key vectors. Given three learned matrices W_Q, W_K and W_V, we have:

$$q_{head} = w_{head} * W_Q \quad (3)$$

$$k_i = w_i * W_K \quad (4)$$

$$v_i = w_i * W_V \quad (5)$$

The multi-head self-attention offers as many attention distributions as there are words in the sentence. Therefore it becomes harder to visualise attention weights, and equally harder to interpret.

Devlin et al. (2018) introduce the BERT embedding model. Its architecture consists of multiple stacked transformer encoder layers, making it difficult to interpret. It has however performed strongly on benchmarks such as GLUE (Wang et al., 2018).

Liu and Lapata (2019) introduce a multi-document summarization model. It contains inter-paragraph attention using the same formulas as self-attention. The resulting context vectors are added to words for summarization worthiness prediction.

All of the cited papers in this section have beaten the previous state of the art. However, they have not provided visualisations of the learned attention distributions, nor have they attempted to interpret their models.

4 Alternatives to Attention

Jain and Wallace (2019) argue that it has not been proven that attention distributions can provide interpretations for a model's predictions. They provide examples where non-learned attention distributions get the same predictions as learned ones, and find that attention distributions rarely correlate with feature importance weights. Serrano and Smith (2019) erase intermediate representations by zeroing out their attention weights, and find that attention weights only noisily predict overall importance with regard to the model. Both studies focus on classification and omit language modeling and machine translation.

Attention has been proposed as a way to deal with the forgetfulness of LSTM networks. Haviv et al. (2019) study the memory cells of LSTMs by proposing the following problem: at the first step, an LSTM is shown an image of a number from the MNIST dataset. The LSTM is then shown noise until step t. At the t-th step, the LSTM is asked to predict what number it has seen at the first step. Their results show that the faster a representation moves across steps, the more likely it is to be forgotten, and that the forgetfulness grows as t grows.

5 Future Work

We suggest a number of future directions.

Providing Explainable Self-Attention. The number of distributions that self-attention produces makes it difficult to have interpretable predictions. A way to address that would be to back-propagate the attention weights through the layers to obtain global self-attention distributions. Self-attention can also provide weights for hierarchical combinations, similarly to Yang et al. (2016).

Memory in LSTM and GRU Networks. RNN networks with memory do not represent efficiently long text sequences, but how much do we know about their memory potential? A similar study to Haviv et al. (2019) for an NLP problem would help to show how words, or sentences, are represented in the hidden space, and how they are remembered or forgotten.

Information Learned in Text Embeddings. Another aspect of interpretability is to know what information models learn. Conneau et al. (2018) propose probing tasks for learned sentence embeddings, and Li et al. (2016) erase parts of the learned representations to observe the effects on predictions and interpret them.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $ &!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Doron Haviv, Alexander Rivkind, and Omri Barak. 2019. Understanding and controlling memory in recurrent neural networks. In *ICML*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *NAACL-HLT*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Matīss Rikters. 2018. Impact of corpora quality on neural machine translation. *arXiv preprint arXiv:1810.08392*.

Matīss Rikters, Mārcis Pinnis, and Rihards Krišlauks. 2018. Training and adapting multilingual nmt for less-resourced and morphologically rich languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Giancarlo Salton, Robert Ross, and John Kelleher. 2017. Attentive language models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 441–450.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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 *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Felix Wu, Tianyi Zhang, Amauri H. Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.