CS410 – Tech Review (NetID: Wenxif2)

The Application of Attention Models in Text Classification

Attention mechanism has been broadly understood as the technique to help the model give more weight to the more relevant inputs in the way that helps make more accurate prediction. It has revolutionized the deep learning field, especially in computer visioning an NLP.

Dzmitry Bahdanau et.al proposed the 1st attention model[1], and since then, we have seen the growing use in NLP, we have seen that it was widely used in various types of tasks including text classification, text summarization, topic modeling, machine translation, question-answering and so on.

Text classification has been one of the most common NLP tasks and has been a widely discussed topic in the research field. In this review, I will summarize and discuss the attention mechanism and its use in the classification tasks.

Mechanism of Attention

The intuition behind attention mechanism is similar to biological attention, in the sense that its leveraging sources to give better context by assigning more accurate weight to each piece of input, at each step of training/prediction or at difference (stages) of the tasks. Depending on the task and model choice, there could be various sources that include but not limited to inputs, targets, background documents.

1. Inputs to the attention models

The most common inputs into the attention models has been embeddings, which is the representation of input through which the features and context of inputs could be extracted. This form of pre-processing gives the subsequent models a good basis that not only transform the inputs into numerical form, but also extract the features to certain advantages. For example, one of the most discussed advantage of word embeddings is dimension reduction. Compared with bag of words representation, this could reduce the dimension from vocabulary size to the selected dimension.

There are various forms of embeddings in addition to the word embeddings discussed above, with examples being the position embeddings, segment embeddings. These different forms of embeddings could be used separate or combined to add information from different sources/aspects of inputs.

Other forms of inputs to attention models could be vectorized representation (e.g. bag of words), or modified version that considers of characters of inputs such as TF-IDF. These forms of input however turn to capture less of context and structure of input. For example, bag of words representation tends to lose the order of the inputs.

In addition, it was also observed that the inputs vectors or word embeddings are transformed in various ways before being fed into the attention calculation. Examples of such transformations are linear, concatenation, pooling, etc. [2]

2. How attention is generated

While different attention models tend to have different architecture, speaking generally at the high level, the attention is calculated as probability distribution over inputs, that signifies how important the specific input is at the certain stage of training/prediction.

CS410 – Tech Review (NetID: Wenxif2)

Most of the models or architectures tend to have key, value pairs that are generated from inputs (for example embeddings). In addition, there is also query, to which the key is compared to generate a score that implies the potential level of attention. Depending on the task, the query could be from the various sources. When the query is from the same source as the key, such scenarios are called self-attention models that turns to work well on capturing the context information. One significance of the model is that one could choose the boundary of context, allowing users to capture not only dynamically but also flexibly over the horizon of inputs, either a chunk of the sequence, or the whole input depending on the setting. One could also learn this setting through training and choose the one that performs the best for the certain task. In addition, the query could be from the other sources. For example, for machine translation tasks, the query is often the embeddings of the target/translate word from one word/sequence back. This serves as the guide for the attention model to identify the context and thus calculate the probability distribution specific to the local steps.

Most of the models and architectures tend to calculate the attention probability distribution based off the similarity of key and query. For example, the straightforward approaches could be cosine and Euclidean distances. In addition, some other approaches tend to use neural networks (e.g. CNN, RNN) to calculate such distribution. Once the probability distribution was calculated, there are also various ways to use such distribution to produce the consolidated output that severs as the processed input to be passed to the later phase of the model for the specific task. Approaches such as average, weighted average and pooling were quite widely used [2]. For example, once the probability distribution regarding each input is known, then the consolidated input could be calculated by weighting each piece of the input by its corresponding probability.

It should also be noted that there could be several layers of attention, the interaction between which could vary. For example, different attention scores could be generated from different key/value and query pairs that capture the different aspects of input, combining which to get the final processed input. Or the output of one attention model could be served as the query input to the other/parallel attention model, guiding the context of the other attention layer. [3]

Connecting the dot – how attention helps text classification

Form the discussion above, one could tell that attention models could serve the purpose of feature selection and extraction well for text classification tasks. Processing the inputs through attention models, especially when the attention model is trained with the later steps of the classification model, could guide the training process to identity which parts of the inputs contributes the most to differentiating between the classes. It also acts to reduce noise in this sense.

Three significance I would like to bring up here. One is that the attention mechanism could help differentiate context to an extent where same/similar words would have different embedding and attention scores in different context settings. This would overcome some significant challenges on semantic analysis, where basic embeddings would give same weight to the same word even in different context. The second significance is that the fact that attention could incorporate various source of information from various locations could help in great extent in collecting information beyond traditional horizon. For example, in addition to studying the similarity on the word level, which attention models could do, the input structure information (e.g. positional of words, etc.) can also be fed into attention models from which to extract useful features to guide the assignment of weight. The third significance is that attention could be constructed in various level – character, sentence, and documents. [2] This allows interactive learning process that extracts richer information that resides in different levels of the inputs, providing more insights on the contexts.

Conclusion

In summary, attention models could contribute to the text classification through feature selection and extraction. It has great potential to improve the classifier by helping extract richer information and identifying the information that is more relevant to the task. It could also act to reduce noise in this sense. There are various attention models, either pre-trained or not, that has been proved with improved performance. However, the attention models could also come with some computational expense, thus one will always need to weigh the cost with its benefits.

Reference:

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," ICLR, 2015, pp. 1–15

[2] Andrea Galassi, Marco Lippi, and Paolo Torroni, "Attention in Natural Language Processing", IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

[3] https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/?utm_source=blog&utm_medium=comprehensive-guide-attention-mechanism-deep-learning

[4] https://nlp.stanford.edu/projects/glove/

[5] https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/?utm_source=blog&utm_medium=demystifying-bert-groundbreaking-nlp-framework