Understanding Attention for Text Classification

Xiaobing Sun and **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
xiaobing_sun@mymail.sutd.edu.sg, luwei@sutd.edu.sg

Abstract

Attention has been proven successful in many natural language processing (NLP) tasks. Recently, many researchers started to investigate the interpretability of attention on NLP tasks. Many existing approaches focused on examining whether the local *attention weights* could reflect the importance of input representations. In this work, we present a study on understanding the internal mechanism of attention by looking into the gradient update process, checking its behavior when approaching a local minimum during training. We propose to analyze for each word token the following two quantities: its *polarity score* and its *attention score*, where the latter is a global assessment on the token's significance. We discuss conditions under which the attention mechanism may become more (or less) interpretable, and show how the interplay between the two quantities may impact the model performance.[1]

1 Introduction

Attention mechanism (Bahdanau et al., 2015) has been used as an important component across a wide range of NLP models. Typically, an attention layer produces a distribution over input representations to be attended to. Such a distribution is then used for constructing a weighted combination of the inputs, which will then be employed by certain downstream modules.

Recently, several research efforts on investigating the interpretability of attention on tasks such as text classification, question answering, and natural language inference (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Arras et al., 2019) have been conducted. One of their important arguments was whether the attention distribution could adequately reflect the significance of inputs. To answer this question, they designed a series of metrics and conducted corresponding experiments. In their approaches, they were mainly observing how the attention may impact the outputs on the pre-trained models by changing some elements in the inputs. While such approaches have resulted in interesting findings, the attention mechanism itself remains a *black box* to us – it is still largely unclear what are the underlying factors that may have an impact on the attention mechanism.

When analyzing the results of a typical model with attention on the text classification tasks, we noticed that in some instances, many of the word tokens with large attention weights were adjectives or adverbs which conveyed explicit signals on the underlying class label. On the other hand, in some other instances, we also noticed that such useful words may not always be able to receive significant attention weights, especially under certain configurations of hyperparameters, making the attention mechanism less interpretable.

Such observations lead to several important questions. First, the attention weight for a word token appears to be the relative measurement to its significance, and is largely local and instance specific. Would there be an instance-independent quantity to assess the corpus-level importance of a word token? And if so, what role would such a quantity play in terms of interpreting the overall attention mechanism? Second, when the attention mechanism appears to be less interpretable, how would the underlying model be affected in terms of performance?

In this work, we focus on answering the above questions. We argue that the *attention scores* (rather than *attention weights*) are able to capture the *global*, *absolute* importance of word tokens within a corpus. We present a study to figure out the underlying factors that may influence such attention scores under a simple neural classification model. Inspired by Qian (1999), we analyzed the gradients as well as the updates of intermediate variables in the process of gradient descent, and

[1] Supplementary material and code at https://github.com/richardsun-voyager/UAFTC

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3418–3428

found that there exist some implicit trends on the intermediate variables related to attention: the degree of association between a word token and the class label may impact their attention scores. We argue that when certain hyperparameters are properly set, tokens with strong *polarity* – high degree of association with specific labels, would likely end up with large attention scores, making them more likely to receive large attention weights in a particular sentence. While in such scenarios, the attention mechanism would appear to be more interpretable, we also discuss scenarios where the attention weights may become less interpretable, and show how the *polarity scores*, another important token-level quantity, will play their roles in the overall model in terms of contributing towards the model performance.

2 Related Work

Research on interpretability of neural models has received significant attention recently. One approach was using visualization to explore patterns that exist in the intermediate representations of neural networks. Simonyan et al. (2013) visualized the image-specific class saliency on image classification tasks using learnt ConvNets, and displayed the features captured by the neural networks. Li et al. (2016a,b) proposed visualization methods to look into the neural representations of the embeddings from the local composition, concessive sentences, clause composition, as well as the saliency of phrases and sentences, and illustrated patterns based on the visualizations. An erasure method was also adopted to validate the importance of different dimensions and words. Vig and Belinkov (2019) analyzed the attention structure on the Transformer (Vaswani et al., 2017) language model as well as GPT-2 (Radford et al., 2019) pre-trained model.

Another approach to understanding neural approaches is to conduct theoretical analysis to investigate the underlying explanations of neural models. One example is the work of Levy and Goldberg (2014), which regarded the word embedding learning task as an optimization problem, and found that the training process of the skip-gram model (Mikolov et al., 2013a,b) can be explained as implicit factorization of a *shifted positive PMI* (pointwise mutual information) matrix.

Recently, several research efforts have focused on the interpretability of the attention mechanism. Jain and Wallace (2019) raised the question on the explainability of feature importance as captured by the attention mechanism. They found the attention weights may not always be consistent with the feature importance from the human perspective in tasks such as text classification and question answering. Serrano and Smith (2019) also carried out an analysis on the interpretability of the attention mechanism, with a focus on the text classification task. They conducted their study in a cautious way with respect to defining interpretability and the research scope. The paper concluded that the attention weights are noisy predictors of importance, but should not be regarded as justification for decisions. Wiegreffe and Pinter (2019) suggested that the notion of explanation needs to be clearly defined, and the study of the explanation requires taking all components of a model into account. Their results indicated that prior work could not disprove the usefulness of attention mechanisms with respect to explainability. Moreover, Michel et al. (2019) and Voita et al. (2019) examined the multi-head self-attention mechanism on Transformer-based models, particularly the roles played by the heads.

Our work and findings are largely consistent with such findings reported in the literature. We believe there are many factors involved when understanding the attention mechanism. Inspired by Qian (1999), which investigated the internal mechanism of gradient descent, in this work we focus on understanding attention's internal mechanism.

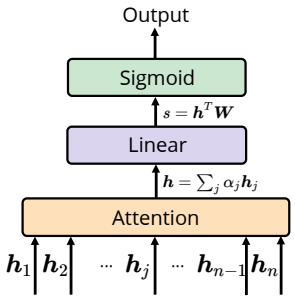


Figure 1: Classification architecture with attention

3 Classification Model with Attention

We consider the task of text classification, with a specific focus on binary classification.[2] The architecture of the model is depicted in Figure 1.

There are various attention mechanisms introduced in the field (Luong et al., 2015). Two commonly used mechanisms are the additive attention (Bahdanau et al., 2015) and scaled dot-product attention (Vaswani et al., 2017). In this work, we will largely focus our analysis on the latter approach (but we will also touch the former approach later).

[2]Extending to multi-class classification is possible. See the supplementary material for detailed analysis and discussion.

Consider an input token sequence of length n: $x = e_1, e_2, \ldots, e_n$, where e_j is the j-th input token whose representation before the attention layer is $\boldsymbol{h}_j \in \mathbb{R}^d$. The *attention score* for the j-th token is:

$$a_j = \frac{\boldsymbol{h}_j^\top \boldsymbol{V}}{\lambda}, \tag{1}$$

where the hyperparameter λ is the scaling factor (typically set to a large value, e.g., $\sqrt{d}$ is often used in the literature (Vaswani et al., 2017)), and $\boldsymbol{V} \in \mathbb{R}^d$ is the context vector that can be viewed as a fixed query asking for the "most informative word" from the input sequence (Yang et al., 2016). The token representation $\boldsymbol{h}_j$ can be the word embedding, or the output of an encoder.

The corresponding *attention weight* would be:

$$\alpha_j = \frac{\exp(a_j)}{\sum_{j'} \exp(a_{j'})}. \tag{2}$$

The complete input sequence is represented as:

$$\boldsymbol{h} = \sum_j \alpha_j \boldsymbol{h}_j, \tag{3}$$

and the output of the linear layer is:

$$s = \boldsymbol{h}^\top \boldsymbol{W}, \tag{4}$$

which we call *instance-level polarity score* of the input sequence. Here, $\boldsymbol{W} \in \mathbb{R}^d$ is the weight vector for the linear layer.

When we make predictions, if the resulting polarity score s is positive, the corresponding input sequence will be classified as positive (i.e., $y = +1$, where y is the output label). Otherwise, it will be classified as negative (i.e., $y = -1$).

During training, assume we have a training set $\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$ with m labeled instances. Our overall loss is:

$$\ell = \frac{1}{m} \sum_{t=1}^{m} \ell^{(t)} = -\frac{1}{m} \sum_{t=1}^{m} \log\Big(\sigma(y^{(t)} s^{(t)})\Big). \tag{5}$$

where $y^{(t)}$ and $s^{(t)}$ are the gold output label and the instance-level polarity score for the t-th instance respectively, and σ is the sigmoid function.

The instance-level polarity score s can also be written as:

$$s = \sum_j \alpha_j \boldsymbol{h}_j^\top \boldsymbol{W} = \sum_j \alpha_j s_j. \tag{6}$$

Here, we have introduced the *token-level polarity score* s_j for the input token representation $\boldsymbol{h}_j$:

$$s_j = \boldsymbol{h}_j^\top \boldsymbol{W}. \tag{7}$$

From here we can observe that the instance-level polarity score of the input sequence can be interpreted as the weighted sum of the token-level polarity scores, where the weights are given by the attention weights (α_j for $\boldsymbol{h}_j$). Such attention weights measure the *relative* importance of the token within a specific input sequence.

On the other hand, the attention score a_j captures the *absolute* importance of the token. We believe such absolute measurements to the significance of words may be playing a more crucial role (than attention weights) when understanding the attention mechanism. Thus, unlike many previous research efforts, we will instead focus on the understanding of attention scores in this work.

In this paper, we will mainly investigate a simple neural model where $\boldsymbol{h}_j = \boldsymbol{e}_j$. Here $\boldsymbol{e}_j$ is the word embedding for the j-th input token. In other words, we assume the word embeddings are used as the inputs to the attention layer. Detailed discussions on other assumptions on $\boldsymbol{h}_j$ can be found in the supplementary material.

4 Analysis

We conduct some analysis in this section to understand how the attention mechanism works for the task of text classification. First, let us consider the following 3 different types of tokens:

- *positive tokens*: tokens that frequently appear in positive training instances only,
- *negative tokens*: tokens that frequently appear in negative training instances only, and
- *neutral tokens*: tokens that appear evenly across both positive and negative training instances.

We also call the first two types of tokens *polarity tokens*. For the ease of analysis and discussion, we assume each token belongs to either of these 3 types, and we assume the dataset is balanced and symmetric[3]. While some of these assumptions may seem strong, having them would significantly simplify our analysis. As we will see later in experiments, even though some of the above assumptions do not hold in some real datasets, our findings are still valid in practice.

The gradient descent algorithm that minimizes a loss ℓ could be interpreted as the integration of

[3] In other words, if we flip the signs of the y labels for all documents in the training set, we arrive at exactly the same training set (under a particular mapping between tokens).

the gradient flow equation using Euler's Method (Scieur et al., 2017; Qian, 1999), written as:

$$\frac{d\boldsymbol{z}(\tau)}{d\tau} = -\nabla \ell(\boldsymbol{z}(\tau)), \boldsymbol{z}(0) = \boldsymbol{z}_0, \tag{8}$$

where $\boldsymbol{z}$ is the parameter vector, and $\boldsymbol{z}_0$ is its initialization, and τ is the time step. We assume that all parameters have initializations, and will omit such initializations in the subsequent differential equations. We will not seek to solve the differential equations directly but to find out whether there exist some trends and patterns for certain variables during training.

4.1 Polarity Score

Consider the token e in the vocabulary whose vector representation is $\boldsymbol{e}$. Let us have an analysis on the polarity score s_e for the token e. This token may appear somewhere in the training set. We write $e_j^{(t)} \equiv e$ if and only if this token e appears as the j-th token in the t-th instance.

Gradient update iteration will be represented as:

$$\frac{ds_e(\tau)}{d\tau} = (\frac{d\boldsymbol{e}(\tau)}{d\tau})^\top \boldsymbol{W}(\tau) + \boldsymbol{e}^\top(\tau)\frac{d\boldsymbol{W}(\tau)}{d\tau}, \tag{9}$$

where $\boldsymbol{W}(\tau)$ is the linear layer weight vector at the time τ. Its update can be represented by another ordinary differential equation:

$$\frac{d\boldsymbol{W}(\tau)}{d\tau} = -\frac{\partial \ell}{\partial \boldsymbol{W}}(\tau), \tag{10}$$

Similarly, we have:

$$\frac{d\boldsymbol{e}(\tau)}{d\tau} = -\frac{\partial \ell}{\partial \boldsymbol{e}}(\tau). \tag{11}$$

For simplicity, we will omit the time step τ in the equations. The derivative of the token level polarity score will be written as:

$$\frac{ds_e}{d\tau} = \underbrace{-\left(\frac{\partial \ell}{\partial \boldsymbol{e}}\right)^\top \boldsymbol{W}}_{\Delta s'_e} + \underbrace{\left(-\boldsymbol{e}^\top \frac{\partial \ell}{\partial \boldsymbol{W}}\right)}_{\Delta s''_e}. \tag{12}$$

The two partial derivatives can be calculated as[4]:

$$\frac{\partial \ell}{\partial \boldsymbol{e}} = -\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left[\frac{\boldsymbol{V}(\boldsymbol{e}-\boldsymbol{h}^{(t)})^\top}{\lambda} + \mathbf{I}\right]\boldsymbol{W}, \tag{13}$$

$$\frac{\partial \ell}{\partial \boldsymbol{W}} = -\frac{1}{m}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\boldsymbol{h}^{(t)}, \tag{14}$$

[4]See the supplementary material for details.

where $(t,j) : e_j^{(t)} \equiv e$ means we are selecting such tokens from the t-th instance at the j-th position that are exactly e, and $\alpha_j^{(t)}$ is the attention weight for that j-th token in the selected t-th instance. The vector $\boldsymbol{h}^{(t)}$ is the representation of the t-th instance, and $\beta^{(t)}$ is defined as $\beta^{(t)} = 1 - \sigma(y^{(t)}s^{(t)})$.

The first term in Equation 12 can be written as:

$$\Delta s'_e = \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{\left(s_e - s^{(t)}\right)}{\lambda}\boldsymbol{V}^\top\boldsymbol{W} + \frac{1}{m}||\boldsymbol{W}||_2^2 \sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}. \tag{15}$$

The sign of the second term above depends on:

$$\pi(e) = \sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}. \tag{16}$$

This term has the following property: it is positive if e is a positive token, negative if e is negative, and close to 0 if e is neutral.

The second term in Equation 12 is:

$$\begin{aligned}\Delta s''_e &= \frac{1}{m}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\boldsymbol{e}^\top\boldsymbol{h}^{(t)} \\ &= \frac{1}{m}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\sum_j \alpha_j^{(t)}\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)} \\ &= \frac{1}{m}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}.\end{aligned} \tag{17}$$

Equation 17 involves dot-products between embeddings. During training, certain trends and patterns will be developed for such dot-products. Near a local minimum, we can show that it is desirable to have $\boldsymbol{e}_i^\top\boldsymbol{e}_j > 0$ when e_i and e_j are both positive tokens or both negative tokens, and $\boldsymbol{e}_i^\top\boldsymbol{e}_j < 0$ when one is a positive token and the other is a negative token. More details and analysis on the desirability of these properties can be found in the supplementary material.

Now let us look at the last term in Equation 17. This term can be re-written as:

$$\frac{1}{m}\sum_{(t,j):y^{(t)}=+1}\beta^{(t)}\alpha_j^{(t)}\left(\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\right) + \frac{1}{m}\sum_{(t,j):y^{(t)}=-1}\beta^{(t)}\alpha_j^{(t)}\left(-\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\right). \tag{18}$$

where we split the term into two based on the polarity of the training instances.

In the first term, each e_j token would be either a positive or a neutral token; in the second term, each e_j would be either a negative or a neutral token, and again under the assumption on the dataset, all the terms involving neutral e_j tokens would roughly sum to a value close to 0 (regardless of e). So we may assume there are no neutral e_j tokens. Now, if e is a positive token, we can see it is desirable for both terms to be positive. If e is negative, it is desirable for both terms to be negative. If e is neutral, likely this term is close to 0.

Overall, the update of s_e is:

$$\begin{aligned}\frac{ds_e}{d\tau} =& \underbrace{\frac{1}{m}\left(\boldsymbol{V}^\top \boldsymbol{W}/\lambda\right)\rho(e)}_{(A)} \\ &+ \underbrace{\frac{1}{m}||\boldsymbol{W}||_2^2\ \pi(e)}_{(B)} \\ &+ \underbrace{\frac{1}{m}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)}}_{(C)}, \end{aligned} \tag{19}$$

where

$$\rho(e) = \sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\Big(s_e - s^{(t)}\Big). \tag{20}$$

Under the assumption that $\boldsymbol{V}^\top \boldsymbol{W}/\lambda$ is reasonably small (for example, we may set λ to an appropriate value, which is reasonably large), we have $A \approx 0$. We then have the following results:

- For positive tokens, we have $B > 0$ and $C > 0$. The corresponding polarity scores will likely increase after each update when approaching the local minimum, and may end up with relatively large positive polarity scores eventually.
- For negative tokens, we have $B < 0$ and $C < 0$. The corresponding polarity scores will likely decrease after each update when approaching the local minimum, and may end up with relatively large negative polarity scores eventually.
- For neutral tokens, we have $B \approx 0$ and $C \approx 0$. Their polarity scores will likely not change significantly after each update when approaching the local minimum, and may end up with polarity scores that are neither significantly positive nor significantly negative eventually.

Based on the above results, we can also quickly note that $\rho(e)$ has the following property: it is positive if e is a polarity token, and close to zero if e is neutral.

These results are desirable as the token-level polarity scores will be used for defining the instance-level polarity scores, which are in term useful for prediction of the final polarity of the sentence containing such tokens.

However, we note that the above results depend on the assumption that term A is small. As we mentioned above, we may assume λ is large to achieve this. When $\boldsymbol{V}^\top \boldsymbol{W}/\lambda$ is not small enough, the term A may lead to a gap in the polarity scores between the positive and negative tokens, depending on the sign of $\boldsymbol{V}^\top \boldsymbol{W}$ – a term that will appear again in the next section when examining the attention scores.

4.2 Attention Score

Now let us have an analysis on the attention score for each token. Again given a token e, the corresponding attention score is $a_e = \frac{\boldsymbol{e}^\top \boldsymbol{V}}{\lambda}$. Note that this is a global score that is independent of any instance. The update of a_e is:

$$\frac{da_e(\tau)}{d\tau} = \frac{1}{\lambda}\left(\frac{d\boldsymbol{e}(\tau)}{d\tau}\right)^\top \boldsymbol{V}(\tau) + \frac{1}{\lambda}\boldsymbol{e}^\top(\tau)\frac{d\boldsymbol{V}(\tau)}{d\tau}. \tag{21}$$

Similarly, let us rewrite the equation as:

$$\frac{da_e}{d\tau} = \underbrace{-\frac{1}{\lambda}\left(\frac{\partial \ell}{\partial \boldsymbol{e}}\right)^\top \boldsymbol{V}}_{\Delta a'_e} + \underbrace{\left(-\frac{1}{\lambda}\boldsymbol{e}^\top \frac{\partial \ell}{\partial \boldsymbol{V}}\right)}_{\Delta a''_e}. \tag{22}$$

We have

$$\frac{\partial \ell}{\partial \boldsymbol{V}} = -\frac{1}{m\lambda}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}_j^{(t)}\Big(s_j^{(t)} - s^{(t)}\Big). \tag{23}$$

The first term can be calculated as:

$$\begin{aligned}\Delta a'_e =& \frac{1}{m\lambda^2}||\boldsymbol{V}||_2^2 \sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\Big(s_e - s^{(t)}\Big) \\ &+ \frac{1}{m\lambda}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{W}^\top \boldsymbol{V}. \end{aligned} \tag{24}$$

The second term is:

$$\Delta a''_e = \frac{1}{m\lambda^2}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)}\Big(s_j^{(t)} - s^{(t)}\Big). \tag{25}$$

Similarly, this can be re-written as:

$$\begin{aligned}&\frac{1}{m\lambda^2}\sum_{(t,j):y^{(t)}=+1} \beta^{(t)}\alpha_j^{(t)}\Big(s_j^{(t)} - s^{(t)}\Big)\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)} \\ &+ \frac{1}{m\lambda^2}\sum_{(t,j):y^{(t)}=-1} \beta^{(t)}\alpha_j^{(t)}\Big(s^{(t)} - s_j^{(t)}\Big)\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)}. \end{aligned} \tag{26}$$

This term shall be close to zero initially, regardless of e. However, this term may become positive for a polarity token e as learning progresses.[5]

The update of a_e is (note that $\boldsymbol{W}^\top \boldsymbol{V} = \boldsymbol{V}^\top \boldsymbol{W}$):

$$\frac{da_e}{d\tau} = \underbrace{\frac{1}{m\lambda^2}\left(\boldsymbol{V}^\top \boldsymbol{W} \cdot \lambda\right)\pi(e)}_{(D)} + \underbrace{\frac{1}{m\lambda^2}||\boldsymbol{V}||_2^2\,\rho(e)}_{(E)} + \underbrace{\frac{1}{m\lambda^2}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right)}_{(F)}. \quad (27)$$

Let us now understand the influence of these terms respectively:

- Term D. When $\boldsymbol{V}^\top \boldsymbol{W} > 0$, the positive tokens will receive a positive update whereas the negative tokens will receive a negative update from this term after each step. When $\boldsymbol{V}^\top \boldsymbol{W} < 0$, the influence is the other way around. It does not influence the attention scores of the neutral tokens much as the corresponding $\pi(e)$ is approximately zero. When it is not close to zero, this term can lead to a gap between the final attention scores of the positive tokens and negative tokens.
- Terms E and F. Based on our analysis, $E > 0$, and $F \geq 0$ for polarity tokens, and $E \approx 0$ and $F \approx 0$ for neutral tokens. This means for the positive tokens and negative tokens, their attention scores will likely receive a positive value from this term after each update when approaching a local minimum. Their corresponding attention scores may end up with large positive scores eventually. For the neutral tokens, this term does not have much influence on their attention scores.

From here we can observe that when $\boldsymbol{V}^\top \boldsymbol{W} \cdot \lambda$ is small, the polarity tokens will likely end up with larger attention scores than the neutral tokens. This is actually a desirable situation – polarity tokens are likely more representative when used for predicting the underlying class labels, and therefore shall receive more "attention" in general.

However, we note that if the scaling factor λ is too large, the term D may be significant. This means the sign of $\boldsymbol{V}^\top \boldsymbol{W}$ will then play a crucial role – when it is non-zero and when λ is very large, positive tokens and negative tokens will likely have attention scores of opposite signs. This may not be a very desirable situation as the attention scores would be less interpretable in that case. On the other hand, as we have discussed in the previous section, the scaling factor λ should not be too small too. Otherwise term A in Equation 19 would not be close to 0 – as a result the conclusions on the polarity scores for the tokens stated at end of Sec 4.1 may not hold.

In conclusion, if we would like to observe the desirable behavior as discussed for the attention mechanism, it is important for us to choose an appropriate λ value or we shall possibly find ways to control the value of $\boldsymbol{V}^\top \boldsymbol{W}$[6]. We will conduct experiments on real datasets to verify our findings.

Besides the above analysis, we have also analyzed polarity scores and attention scores from the model with additive attention, the model with an affine input layer and the model for multi-class classification respectively. There are terms that have similar effects on polarity and attention scores during training. Due to space limitations, we provide such details in the supplementary material.

[5] See the supplementary material for more details.

Dataset	AvgLength	VocabSize	Size		
			Train	Dev	Test
SST	18	16174	3610/3310	444/428	909/912
IMDB	183	63311	8539/8673	2113/2191	2174/2189
20News I	185	17584	624/612	156/154	195/192
20News II	187	29433	794/790/716	91/70/79	84/100/90

Table 1: Datasets are all split into training, dev and test sets, respectively and are all balanced. The first 3 datasets are for binary classification (*positive/negative*), and the last is for 3-class classification (*rec.motorcycles/sci.med/talk.politics.guns*).

5 Experiments

We conducted experiments on four text classification datasets[7]. The statistics of the datasets are shown in Table 1. We followed the work of Jain and Wallace (2019) for pre-processing of the datasets[8], and lower-cased all the tokens.

- Stanford Sentiment Treebank (SST) (Socher et al., 2013). The original dataset that consists of 10,662 instances with labels ranging from 1 (most negative) to 5 (most positive). Similar to the work of Jain and Wallace (2019), we removed neutral instances (with label 3), and regarded instances with label 4 or 5 as positive and instances with the label 1 or 2 as negative.
- IMDB (Maas et al., 2011). The original dataset

[6] We have further discussions on $\boldsymbol{V}^\top \boldsymbol{W}$ in the supplementary material.

[7] We also conducted analysis on synthetic datasets. The results can be found in the supplementary material.

[8] https://github.com/successar/AttentionExplanation

λ	SST				λ	20News I			
	DP	DP-L	DP-A	AD		DP	DP-L	DP-A	AD
0.001	55.3	79.8	67.9	62.8	0.001	54.8	88.6	78.6	49.4
1	74.4	81.2	73.4	73.4	1	88.4	93.0	85.3	87.6
10	82.2	81.7	80.8	80.3	10	92.8	91.2	92.8	92.0
20	81.4	80.9	81.0	81.2	20	93.5	92.2	93.5	91.2
50	80.8	82.0	81.5	79.9	50	93.3	92.3	92.2	91.7
100	81.2	81.1	80.7	80.8	100	92.8	91.2	92.8	93.3
10000	79.6	81.4	79.3	80.8	10000	92.8	92.0	93.0	92.0

λ	IMDB				λ	20News II			
	DP	DP-L	DP-A	AD		DP	DP-L	DP-A	AD
0.001	55.5	87.7	73.3	69.8	0.001	31.8	90.1	64.6	59.2
1	79.5	88.2	85.4	83.7	1	85.4	92.3	88.3	86.7
10	89.2	87.8	89.6	88.2	10	93.4	93.4	91.7	90.0
20	89.6	88.1	89.6	89.6	20	94.9	94.2	93.3	92.1
50	89.8	87.2	89.1	88.5	50	94.9	92.3	92.9	93.8
100	89.3	88.3	89.2	88.8	100	94.9	93.1	92.9	92.9
10000	89.3	88.4	88.9	88.9	10000	94.5	93.8	92.5	92.9

Table 2: Test set results in accuracy (%). Models were chosen based on the highest accuracy on the dev sets. L_2-regularization was adopted on DP-L, DP-A and AD.

that consists of 50,000 movie reviews with positive or negative labels.

- 20Newsgroup I (20News I). The original dataset that consists of around 20,000 newsgroup correspondences. Similar to the work of Jain and Wallace (2019), we selected the instances from these two categories: "*rec.sport.hockey*" and "*rec.sport.baseball*", and regarded the former as positive instances and the latter negative.
- 20Newsgroup II (20News II). This is a dataset for 3-class classification. We selected instances from these three categories: "*rec.motorcycles*" , "*sci.med*" and "*talk.politics.guns*".

Our analysis focused on the ideal case (e.g., positive tokens only appear in positive documents). To be as consistent as possible with our analysis, we only examined the tokens of strong association with specific labels and the tokens that could be seen almost evenly across different types of instances based on their frequencies (note that we only selected these tokens for examination after training, but no tokens were excluded during the training process). We defined a metric γ_e to measure the association between the token e and instance labels[9]:

$$\gamma_e = \frac{f_e^+ - f_e^-}{f_e^+ + f_e^-}, \tag{28}$$

where f_e^+ and f_e^- refer to the frequencies in the positive and in the negative instances respectively. If $\gamma_e \in (0.5, 1)$ and $f_e^+ > 5$, the token will be regarded as a "positive token". If $\gamma_e \in (-1, -0.5)$ and $f_e^- > 5$, the token will be regarded as a "negative token". If $\gamma_e \in (-0.1, 0.1)$ and $|f_e^+ - f_e^-| < 5$, the token will be regarded as a "neutral token".[10]

We ran the experiments using different scaling factors λ on the models with the scaled dot-product attention (DP) and additive attention (AD) respectively. For the former, we also investigated the performances on the models with a LSTM (DP-L) or an affine transformation layer (DP-A) as the input encoder.[11] The Adagrad optimizer (Duchi et al., 2011) was used for gradient descent. Dropout (Srivastava et al., 2014) was adopted to prevent overfitting. All the parameters were learned from scratch to avoid the influence of prior information. For the same reason, while we may be able to use pretrained word embeddings, we chose to initialize word embeddings with a uniform distribution from -0.1 to 0.1 with a dimension $d = 100$.

The results are shown in Table 2. For the scaled dot-product attention, which is our focus in this work, it can be observed that when the scaling factor λ is small (1 or 0.001), the test set results appear to be worse than the case when λ is set to a larger value. The optimal results may be obtained when λ is set to a proper value. However, setting λ to a very large value does not seem to have a significant impact on the performance – in this case, from Equations 1 and 2 we can see that the attention weights will be close to each other for all input tokens, leading to an effect similar to mean pooling. Results on using LSTM or the affine transformation layer as the input encoder are similar – setting a proper value for λ appears to be crucial.

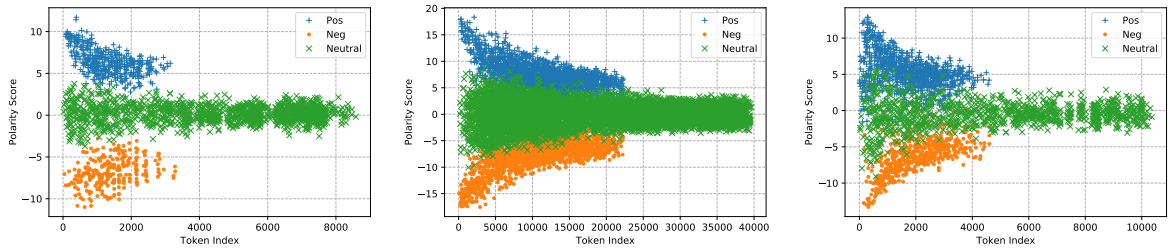
Figure 2 shows the results for polarity scores and attention scores for the first 3 datasets, when λ is set to a moderate value of 10 (i.e., $\sqrt{d}$). These results are consistent with our analysis. It can be observed that generally positive tokens have positive polarity scores while negative tokens have negative polarity scores. Neutral tokens typically have polarity scores around zero. It can also be observed that both the positive and negative tokens generally have larger attention scores than the neutral tokens.

We also examined whether there would be an obvious gap between the attention scores of the polarity tokens when λ is large. As we can see from Figure 3b, when λ is set to 100, the resulting attention scores for the positive tokens are smaller than those of the neutral (and negative) tokens. In

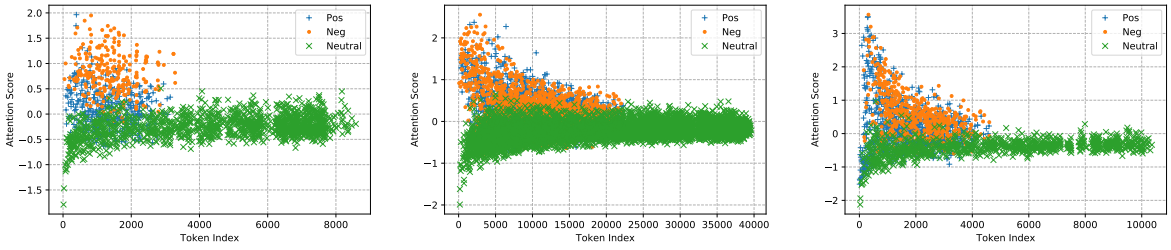
[9]For multi-class classification, we determined the polarity of each token based on the relative frequency of each token with respect to each label. For each token, we calculated the frequency distribution across the labels that they appear in. If the largest element of the distribution is above a given threshold, we will regard the token as a polarity one.

[10]Example selected tokens from these datasets can be found in the supplementary material.

[11]More results from these models can be found in the supplementary material. For each model, we only reported one set of the results with a random initialization as we found the patterns were similar with different initializations.

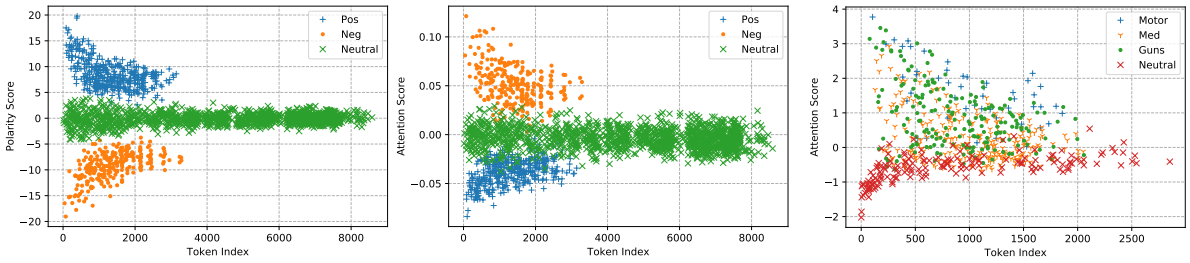


(a) SST: polarity scores ($\lambda = 10$) (b) IMDB: polarity scores ($\lambda = 10$) (c) 20News I: polarity scores ($\lambda = 10$)



(d) SST: attention scores ($\lambda = 10$) (e) IMDB: attention scores ($\lambda = 10$) (f) 20News I: attention scores ($\lambda = 10$)

Figure 2: Polarity (top) and attention scores (bottom). Scaled dot product attention is used with $\lambda = 10$.



(a) SST: polarity scores ($\lambda = 100$) (b) SST: attention scores ($\lambda = 100$) (c) 20News II: attention scores ($\lambda = 10$)

Figure 3: Polarity (left) and attention (middle) scores for SST with scaling factor λ set to 100. Attention scores (right) for 20News II, with scaling factor λ set to 10. Scaled dot product attention is used.

this case, the resulting attention scores appear to be less interpretable. However, as we discussed above, when λ is very large, the attention mechanism will effectively become mean pooling (we can also see from Figure 3b that attentions scores of all tokens are now much smaller), and the overall model would be relying on the average polarity scores of the word tokens in the sentence for making prediction. Interestingly, on the other hand, as we discussed before at the end of Section 4.1, when λ is large, the polarity tokens will likely end up with polarity scores of large magnitudes – a fact that can also be empirically observed in Figure 3a. It is because of such healthy polarity scores acquired, the model is still able to yield good performance in this case even though the attention scores do not appear to be very interpretable.

We also tried to set a constraint on $\boldsymbol{V}^\top \boldsymbol{W}$ by introducing a regularization term to minimize it in the learning process. We found doing so will generally encourage the attention model to produce more interpretable attention scores – for example, even when λ was large, both the positive and negative tokens ended up with positive attention scores that were generally larger than those of the neutral tokens. However, empirically we did not observe a significant improvement in test performance. See the supplementary material for details.

We examined the attention scores on the 20News II dataset which consists of 3 labels. As shown in Figure 3c, polarity tokens that are strongly associated with specific labels are still likely to have larger attention scores than those of neutral tokens.

To understand whether there are similar patterns for the polarity and attention scores when using the additive attention models, we replaced the scaled dot-product attention layer with the additive attention layer and ran experiments on the SST dataset. The results are shown in Figure 4, which are similar to those of our scaled dot-product attention model.

Furthermore, we analyzed the relationship between the global attention scores and the local attention weights. We collected all the attention weights on the test set of SST for the positive, negative and

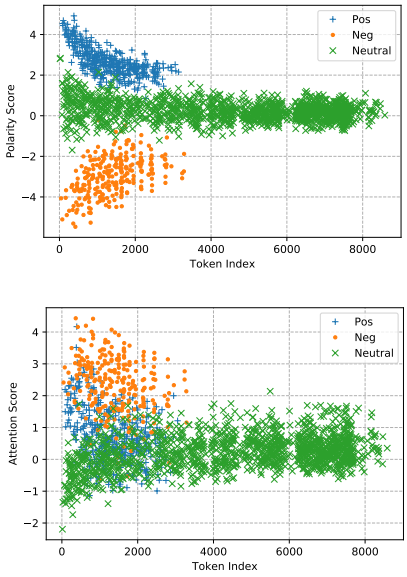


Figure 4: Polarity and attention scores when additive attention is used (on SST, $\lambda = 10$).

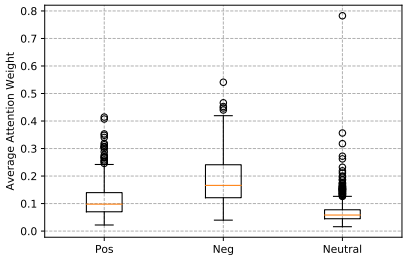


Figure 5: Distributions of average attention weights for positive, negative and neutral tokens. The minimum, maximum, median, first and third quartiles are displayed for tokens of each type. Circles are outliers.

neutral tokens, and calculated the average weight for each token. Next we plot in Figure 5 the distribution of such average attention weights for tokens of these three types separately. As we can observe, generally, the polarity tokens are more likely to have larger attention weights than the neutral tokens. However, the positive tokens seemed to receive lower scores than the negative tokens in terms of the attention weights. This is consistent with the attention scores shown in Figure 2d: the attention scores of the positive tokens were generally lower than those of the negative tokens. Meanwhile, we could see that there were some outliers of large weights for the neutral tokens (circles that appear outside the boxes are outliers). We looked into the case, it was due to that all of the three tokens in the short instance "is this progress" had negative attention scores, and the last token "progress" somehow had a relatively larger one, making its corresponding attention weight the largest amongst the three. This can be explained by the fact that attention weights only capture relative significance of tokens within a local context.

These empirical results support our analysis as well as our belief on the significance of the attention scores. When certain hyperparameters are properly set, the attention mechanism tends to assign larger attention scores to those tokens which have strong association with instances of a specific label. Meanwhile, the polarity scores for such tokens tend to yield large absolute values (of possibly different signs, depending on the polarity of the tokens), which will be helpful when predicting instance labels. By contrast, neutral tokens that appeared evenly across instances of different labels are likely assigned small attention scores and polarity scores, making them relatively less influential.

6 Conclusions

In this work, we focused on understanding the underlying factors that may influence the attention mechanism, and proposed to examine *attention scores* – a global measurement of significance of word tokens. We focused on binary classification models with dot-product attention, and analyzed through a gradient descent based learning framework the behavior of attention scores and *polarity scores* – another quantity that we defined and proposed to examine.

Through the analysis we found that both quantities play important roles in the learning and prediction process and examining both of them in an integrated manner allows us to better understand the underlying workings of an attention based model. Our analysis also revealed factors that may impact the interpretability of the attention mechanism, providing understandings on why the model may still be robust even under scenarios where the attention scores appear to be less interpretable. The empirical results of experiments on various real datasets supported our analysis. We also extended to and empirically examined additive attention, multi-label classification and models with an affine input layer, and observed similar behaviors.

There are some future directions that are worth exploring. Specifically, we can further examine the influence of using pre-trained word embeddings – whether similar words can help each other boost their polarity and attention scores. Moreover, we can also examine the influence of using deep contextualized input encoders such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018).

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. We also thank Rui Qiao for his help on proofreading. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-012), and Ministry of Education, Singapore, under its Academic Research Fund (AcRF) Tier 2 Programme (MOE AcRF Tier 2 Award No: MOE2017-T2-1-156). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and AI Singapore or the views of the Ministry of Education, Singapore.

References

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. Evaluating recurrent neural network explanations. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

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

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of NAACL*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NeurIPS*.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. *Proceedings of NAACL*.

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

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Proceedings of NeurIPS*.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*.

Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Technical report, OpenAI*.

Damien Scieur, Vincent Roulet, Francis Bach, and Alexandre d'Aspremont. 2017. Integration methods and accelerated optimization algorithms. In *Proceedings of NeurIPS*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of EMNLP*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Elena Voita, David Talbot, F. Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of ACL*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of EMNLP*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL*.