Supplementary Material for "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

This is the supplementary material for the paper Understanding Attention for Text Classification (Sun and Lu, 2020).

A Gradient Calculation

The gradients will be derived based on the model with scaled dot-product attention as discussed in the main paper. We consider the case of a single training instance for simplicity and clarity. In other words, we assume $m = 1$. The gradient on the entire training set can be calculated as the average of the gradients on all training instance.

A.1 Gradients of s, $\boldsymbol{h}$, $\boldsymbol{W}$

$$\begin{aligned}\frac{\partial \ell}{\partial s} &= -y\beta, \\ \frac{\partial \ell}{\partial \boldsymbol{W}} &= \frac{\partial \ell}{\partial s}\frac{\partial s}{\partial \boldsymbol{W}} \\ &= -y\beta \boldsymbol{h}, \\ \frac{\partial \ell}{\partial \boldsymbol{h}} &= \frac{\partial \ell}{\partial s}\frac{\partial s}{\partial \boldsymbol{h}} \\ &= -y\beta \boldsymbol{W},\end{aligned} \tag{29}$$

where $\beta = 1 - \sigma(ys) = \sigma(-ys)$.

A.2 Gradient of $\boldsymbol{e}$

In our model with scaled dot-product attention, the input of the attention layer consists of word embeddings. The input can be generalized to the output of an affine layer. Here, we will use $\boldsymbol{h}_i$ instead of $\boldsymbol{e}_i$ to denote the general representation of the token at the i-th position in the instance. The partial derivative of $\boldsymbol{h}$ with respect to the representation of the i-th token $\boldsymbol{h}_i$ can be calculated as:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \frac{\partial \sum_j \alpha_j \boldsymbol{h}_j}{\partial \boldsymbol{h}_i} \\ &= \sum_j \frac{\partial \alpha_j}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I},\end{aligned} \tag{30}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is an identity matrix. Note that, $\boldsymbol{h}_i$ can be the embedding of the i-th token, or the i-th output of an affine layer (discussed in Section D). The partial derivative of α_j with respect to $\boldsymbol{h}_i$ can be represented as:

$$\frac{\partial \alpha_j}{\partial \boldsymbol{h}_i} = \frac{\partial \alpha_j}{\partial a_i}\frac{\partial a_i}{\partial \boldsymbol{h}_i} = \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}, \tag{31}$$

where δ_{ji} is the Kronecker delta function:

$$\delta_{ji} = \begin{cases} 1 & j = i \\ 0 & else. \end{cases} \tag{32}$$

Based on Equations 31 and 32 , we can obtain the partial derivative of $\boldsymbol{h}$ with respect to $\boldsymbol{h}_i$ as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \sum_j \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I} \\ &= \alpha_i \left[\frac{\partial a_i}{\partial \boldsymbol{h}_i}(\boldsymbol{h}_i - \boldsymbol{h})^\top + \mathbf{I}\right].\end{aligned} \tag{33}$$

On the model with dot-product attention, the attention score and corresponding partial derivative will be written as:

$$\begin{aligned}a_i &= \frac{\boldsymbol{h}_i^\top \boldsymbol{V}}{\lambda}, \\ \frac{\partial a_i}{\partial \boldsymbol{h}_i} &= \frac{\boldsymbol{V}}{\lambda},\end{aligned} \tag{34}$$

where $\boldsymbol{V} \in \mathbb{R}^d$ is the context vector. With Equation 33 and 34, we can have:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \sum_j \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I} \\ &= \alpha_i \left[\frac{\boldsymbol{V}(\boldsymbol{h}_i - \boldsymbol{h})^\top}{\lambda} + \mathbf{I}\right].\end{aligned} \tag{35}$$

With Equations 29 and 35, we can obtain the partial derivative of the loss with respect to $\boldsymbol{h}_i$:

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{h}_i} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i}\frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= -y\beta\alpha_i \left[\frac{\boldsymbol{V}(\boldsymbol{h}_i - \boldsymbol{h})^\top}{\lambda} + \mathbf{I}\right]\boldsymbol{W}.\end{aligned} \tag{36}$$

As the inputs of the attention layer are embeddings directly, given a token e (in other words, $\boldsymbol{h}_i = \boldsymbol{e}$), the partial derivative on the entire training set can be represented as:

$$\frac{\partial \ell}{\partial \boldsymbol{e}} = -\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \left[\frac{\boldsymbol{V}(\boldsymbol{e} - \boldsymbol{h}^{(t)})^\top}{\lambda} + \mathbf{I} \right] \boldsymbol{W}, \tag{37}$$

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.

A.3 Gradient of $\boldsymbol{V}$

The partial derivative of ℓ with respect to $\boldsymbol{V}$ can be written as:

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{V}} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{V}} \frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= \sum_j \frac{\partial \alpha_j \boldsymbol{h}_j}{\partial \boldsymbol{V}} \frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= \sum_j \frac{\partial \alpha_j}{\partial \boldsymbol{V}} \boldsymbol{h}_j^\top \frac{\partial \ell}{\partial \boldsymbol{h}}. \end{aligned} \tag{38}$$

Note that the attention weight α_j is the function of $\boldsymbol{V}$. The partial derivative of α_j with respect to $\boldsymbol{V}$ is:

$$\begin{aligned} \frac{\partial \alpha_j}{\partial \boldsymbol{V}} &= \sum_i \frac{\partial \alpha_j}{\partial a_i} \frac{\partial a_i}{\partial \boldsymbol{V}} \\ &= \sum_i \alpha_j (\delta_{ji} - \alpha_i) \frac{\boldsymbol{h}_i}{\lambda} \\ &= \frac{\alpha_j}{\lambda} (\boldsymbol{h}_j - \boldsymbol{h}). \end{aligned} \tag{39}$$

Plugging Equation 39 into Equation 38, we will obtain the following:

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{V}} &= -\frac{y\beta}{\lambda} \sum_j \alpha_j (\boldsymbol{h}_j - \boldsymbol{h}) \boldsymbol{h}_j^\top \boldsymbol{W} \\ &= -\frac{y\beta}{\lambda} \left(\sum_j \alpha_j \boldsymbol{h}_j \boldsymbol{h}_j^\top - \boldsymbol{h}\boldsymbol{h}^\top \right) \boldsymbol{W} \\ &= -\frac{y\beta}{\lambda} \sum_j \alpha_j \boldsymbol{h}_j (\boldsymbol{h}_j - \boldsymbol{h})^\top \boldsymbol{W}. \end{aligned} \tag{40}$$

For the simple model in our paper, $\boldsymbol{h}_j = \boldsymbol{e}_j$, the gradient of $\boldsymbol{V}$ on the entire training set can be calculated as:

$$\begin{aligned} &\frac{\partial \ell}{\partial \boldsymbol{V}} = \\ &-\frac{1}{m\lambda} \sum_{t=1}^{m} y^{(t)} \beta^{(t)} \sum_j \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(\boldsymbol{e}_j^{(t)} - \boldsymbol{h}^{(t)} \right)^\top \boldsymbol{W} \\ &= -\frac{1}{m\lambda} \sum_{t=1}^{m} y^{(t)} \beta^{(t)} \sum_j \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(s_j^{(t)} - s^{(t)} \right). \end{aligned} \tag{41}$$

B Analysis of Embeddings

Given a token e, the gradient of the corresponding embedding can be calculated as:

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{e}} = &-\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \frac{\left(s_e - s^{(t)}\right)}{\lambda} \boldsymbol{V} \\ &-\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \boldsymbol{W}. \end{aligned} \tag{42}$$

Let us use Ω_{ik} to denote the *embedding dot-product* between two tokens e_i and e_k. Similar to what we have done for the polarity score and attention score, we can get the derivative of Ω_{ik} with respect to τ:

$$\frac{d\Omega_{ik}}{d\tau} = -\left(\frac{\partial \ell}{\partial \boldsymbol{e}_i} \right)^\top \boldsymbol{e}_k - \left(\frac{\partial \ell}{\partial \boldsymbol{e}_k} \right)^\top \boldsymbol{e}_i, \tag{43}$$

The first term in Equation 43 can be expanded as:

$$\begin{aligned} -\left(\frac{\partial \ell}{\partial \boldsymbol{e}_i} \right)^\top \boldsymbol{e}_k = &\underbrace{\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e_i} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \frac{\left(s_i - s^{(t)}\right) a_k}{\lambda}}_{A_{ik}} \\ &+ \underbrace{\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e_i} y^{(t)} \beta^{(t)} \alpha_j^{(t)} s_k}_{B_{ik}} \end{aligned} \tag{44}$$

The second term can be expanded similarly.

B.1 Analysis of Embedding Dot-products

In this part, we focus on the analysis of the signs of embedding dot-products between different types of tokens.

Assume the scaling factor λ is sufficiently large, then A_{ik} in Equation 44 will be negligible, and the gradient can be approximated as:

$$-\left(\frac{\partial \ell}{\partial \boldsymbol{e}_i}\right)^\top \boldsymbol{e}_k \approx \pi(e_i) s_k. \tag{45}$$

And the derivative of Ω_{ik} with respect to τ will be described as:

$$\begin{aligned}\frac{d\Omega_{ik}}{d\tau} &\approx \pi(e_k)\boldsymbol{e}_i^\top \boldsymbol{W} + \pi(e_i)\boldsymbol{e}_k^\top \boldsymbol{W} \\ &= \pi(e_k)s_i + \pi(e_i)s_k.\end{aligned} \tag{46}$$

Therefore, the derivative depends on the token-level polarity scores s_i and s_k. According to Equation 19 in the main paper, we have that:

$$\begin{aligned}\frac{ds_k}{d\tau} &\approx \underbrace{\frac{1}{m}||\boldsymbol{W}||_2^2\ \pi(e_k)}_{(B)} \\ &\quad + \underbrace{\frac{1}{m}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}_k^\top \boldsymbol{e}_j^{(t)}}_{(C)} \\ &= \underbrace{\frac{1}{m}||\boldsymbol{W}||_2^2\ \pi(e_k)}_{(B)} \\ &\quad + \underbrace{\frac{1}{m}\sum_{(t,j)} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\Omega_{kj}^{(t)}}_{(C)}.\end{aligned} \tag{47}$$

The update of polarity scores also depend on the embedding dot-products. We have that $\pi(e) > 0$ for a positive token, $\pi(e) < 0$ for a negative token, and $\pi(e) \approx 0$ for a neutral token.

In the beginning, as the parameters were randomly initialized, Part C in Equation 47 would be close to zero. The update of the polarity scores would rely more on Part B. Therefore, the polarity scores would likely be increasing for positive tokens, be decreasing for negative tokens and not change significantly for neutral tokens in the beginning.

After sufficient time steps, the polarity scores will likely become significantly positive for positive tokens, significantly negative for negative tokens and sufficiently small for neutral tokens. The derivative of Ω_{ik} with respect to τ will be positive for both positive-positive and negative-negative token pairs ($\pi(e_k)s_i > 0$, $\pi(e_i)s_k > 0$) in Equation 46, be negative for positive-negative token pairs

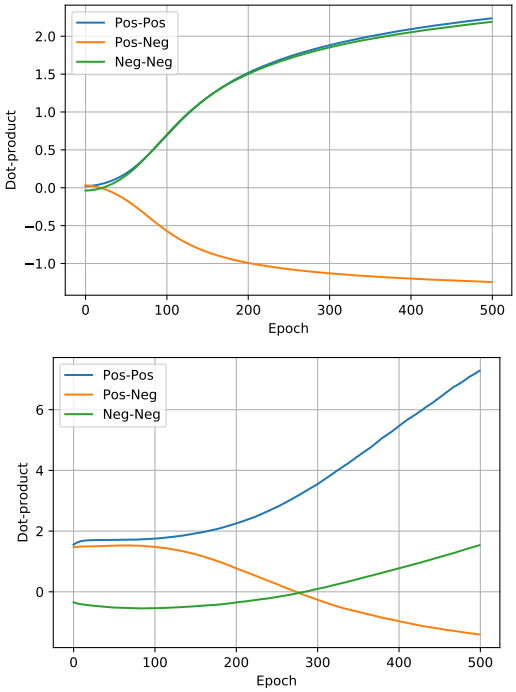


Figure 1: Top: embeddings with uniform initializations between -0.1 and 0.1, the embedding dot-product between two positive (or two negative) tokens are increasing during training whereas the one between a positive and a negative token are decreasing. Bottom: embeddings with uniform initializations between -1 and 1, similar trend but the dot-product between two negative tokens have some struggles in the beginning. Model trained on the synthetic dataset (introduced later).

($\pi(e_k)s_i < 0$, $\pi(e_i)s_k < 0$). For token pairs involving neutral tokens, assume e_k is a neutral token, then we have $\pi(e_k) \approx 0$ and $s_k \approx 0$. The derivative will likely be sufficiently small ($\pi(e_k)s_i \approx 0$, $\pi(e_i)s_k \approx 0$) .

This will make the embedding dot-products change in a direction that positive-positive or negative-negative token pairs will likely have significantly positive embedding dot-products, positive-negative token pairs will likely have significantly negative embedding dot-products and neutral-positive/negative/neutral token pairs will likely remain sufficiently small. In turn this trend of embedding dot-products will make the polarity scores change in a desirable direction and keep the signs of them consistent afterwards.

The trends of the embedding dot-products between different types of tokens during training are shown in Figure 1. It seems initializing embeddings with a smaller range will make the model develop towards a desirable direction more quickly. This may indicate with a less significant Part C

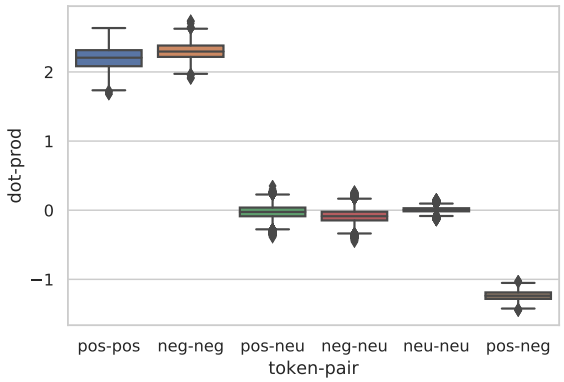


Figure 2: Boxplot embedding dot-products from different token pairs. Each box shows the distribution of the dot-products for token pairs from a specific group. 'pos-pos': positive-positive token pairs, 'neg-neg': negative-negative token pairs, 'pos-neu': positive-neutral token pairs, 'neg-neu': negative - neutral token pairs, 'neu-neu': neutral-neutral tokens pairs, 'pos-neg': positive-negative token pairs. A SGD optimizer is used with random initializations, and the scaling factor is 10. Model trained on the synthetic dataset.

in Equation 47, the model behaves in a way more consistent with our analysis.

B.2 Discussion on F in Equation 27

Now let us take a look how learning progress changes F. We focus on the magnitudes of embedding dot-products that have impacts on it.

In the beginning of the training the term F is close to 0, the update of the attention scores will be dominated by term E in Equation 27 assuming $\boldsymbol{V}^\top \boldsymbol{W}$ to be sufficiently small. As a result, the attention scores for polarity tokens will likely increase and become positively large after sufficient time steps.

Assume we have a polarity token e_* (we have already discussed the case where e_* is a neutral token in the previous section). Since the dataset is symmetric, for any positive token e_+, there is always a corresponding negative token e_-. We will discuss the updates of the embedding dot-products Ω_{*+} and Ω_{*-}

Using Equation 43 and Equation 44, we can obtain the derivatives of Ω_{*+} and Ω_{*-} with respect to τ respectively. It is important to note that we have the following property: $A_{*+}, A_{+*} > 0$ and $A_{*-}, A_{-*} > 0$.

If e_* is a positive token, for the derivative of Ω_{*+}, we can have $s_* - s^{(t)} > 0$, $a_+ > 0$, $s_+ > 0$ (and $y^{(t)}\beta^{(t)}\alpha_j^{(t)} > 0$ for all j). Therefore, for the two terms defined in Equation 44, we have $A_{*+} > 0$ and $B_{*+} > 0$. We can obtain the following inequality:

$$-\left(\frac{\partial \ell}{\partial \boldsymbol{e}_*}\right)^\top \boldsymbol{e}_+ = A_{*+} + B_{*+} > B_{*+} > 0. \quad (48)$$

Similarly, we can have:

$$-\left(\frac{\partial \ell}{\partial \boldsymbol{e}_+}\right)^\top \boldsymbol{e}_* > B_{+*} > 0. \quad (49)$$

Therefore, we can obtain:

$$\frac{d\Omega_{*+}}{d\tau} > B_{*+} + B_{+*} > 0. \quad (50)$$

For the derivative of Ω_{*-}, similarly we can have $s_* - s^{(t)} > 0$, $a_- > 0$, $s_- < 0$. Therefore, in Equation 44, $A_{*-} > 0$, $B_{*-} < 0$ ($A_{*-} < -B_{*-}$ as λ is sufficiently large). The inequalities will be:

$$0 > -\left(\frac{\partial \ell}{\partial \boldsymbol{e}_*}\right)^\top \boldsymbol{e}_- = B_{*-} + A_{*-} > B_{*-}, \quad (51)$$

$$0 > -\left(\frac{\partial \ell}{\partial \boldsymbol{e}_-}\right)^\top \boldsymbol{e}_* = B_{-*} + A_{-*} > B_{-*}. \quad (52)$$

We can obtain:

$$0 > \frac{d\Omega_{*-}}{d\tau} > B_{*-} + B_{-*}. \quad (53)$$

As e_+ and e_- forms a pair, they will have attention scores, polarity scores with opposite signs but the same magnitude, namely $-B_{*-} = B_{*+}$, $-B_{-*} = B_{+*}$. Then we can have:

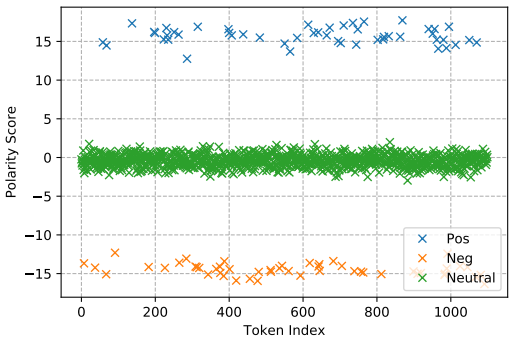
$$\frac{d\Omega_{*+}}{d\tau} > B_{*+} + B_{+*} = -B_{*-} - B_{-*} > -\frac{d\Omega_{*-}}{d\tau}. \quad (54)$$

Similarly, if e_* is a negative token, we can have the inequality:

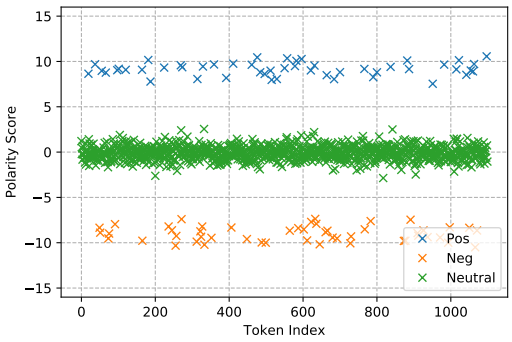
$$\frac{d\Omega_{*-}}{d\tau} > B_{*-} + B_{-*} = -B_{*+} - B_{+*} > -\frac{d\Omega_{*+}}{d\tau}. \quad (55)$$

Therefore, for a positive token e_*, Ω_{*+} will likely receive a larger magnitude update than Ω_{*-} during training and end up with a larger magnitude, that is, $\Omega_{*+} + \Omega_{*-} > 0$. For a negative token e_*, we can have $\Omega_{*-} + \Omega_{*+} > 0$ similarly.

This can be observed in the experiment on the synthetic dataset as shown in Figure 2: positive-positive token pairs and negative-negative token

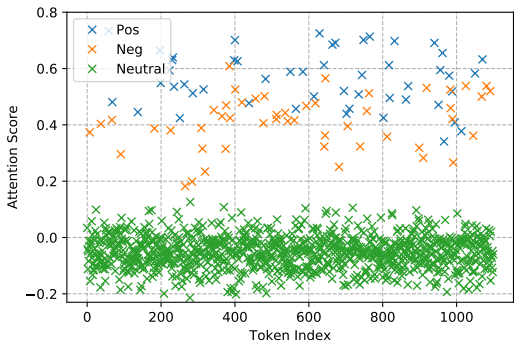


(a) Data I

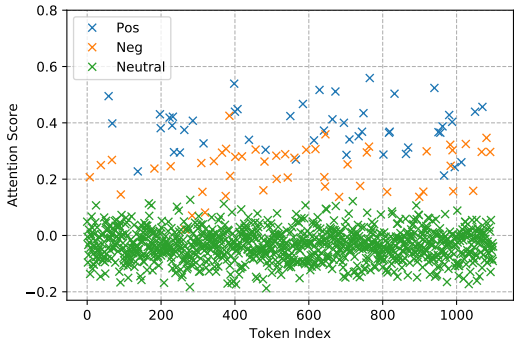


(b) Data II

Figure 3: Polarity scores for polarity tokens and neutral tokens on Data I and Data II respectively. The scaling factor λ is 10.



(a) Data I



(b) Data II

Figure 4: Attention scores for polarity tokens and neutral tokens on Data I and Data II respectively. The scaling factor λ is 10.

pairs generally have larger embedding dot-product magnitudes than positive-negative token pairs.

Let us go back to the term F in Equation 27 which can be viewed as the weighted sum of embedding dot-products. For a positive token e_*, each dot-product Ω_{*+} will be corresponding a dot-product Ω_{*-} with the same weight, the overall value of F will be significantly positive. For a negative token e, the overall value of F will also be significantly positive. The embedding dot-products involving neutral tokens are sufficiently small and negligible as analysed .

C Experiments on Synthetic Datasets

We conducted experiments on two synthetic datasets. We created a vocabulary consisting of three types of tokens: positive, negative and neutral, with size 50, 50, 1,000 respectively. The instances were generated based on combinations of those tokens. Each instance had 12 tokens, namely 2 random positive tokens (or negative tokens) and 10 random neutral tokens. In Data I, the positive/negative tokens only appeared in the positive/negative instances. Data II a noisy version of Data I, with positive/negative tokens appearing a few times in negative/positive instances. The datasets were balanced and symmetric.

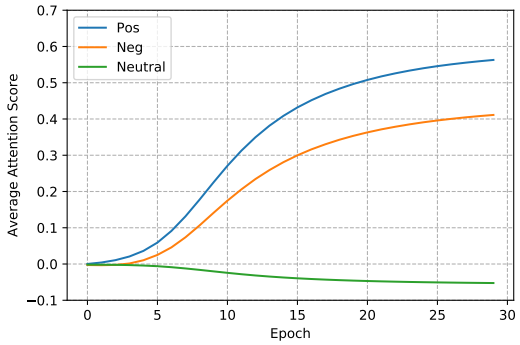
A SGD optimizer was adopted during training, all of the parameters were learned from scratch. The learning rate was fixed and the embeddings were uniformly initialized in the range -0.1 to 0.1. We ran multiple trials for each model, and found the trends and patterns were similar. Hence, only one set of the results were reported for each model.

C.1 Polarity&Attention Score

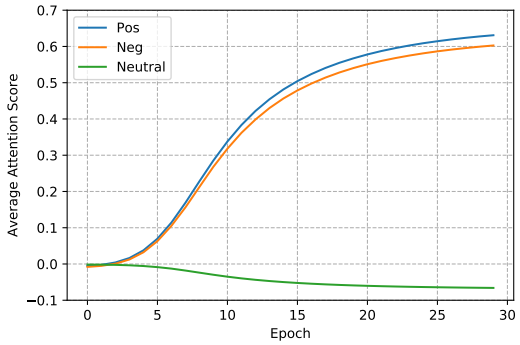
Our analysis on the polarity scores and attention scores was supported by the results shown in Figure 3, Figure 4. Note that, even with certain noise on Data II, the patterns were still robust.

C.2 Properties of Embedding Dot-products

The trends of embedding dot-products support our analysis as shown in Figure 6: the dot-product between the embeddings of either two positive tokens or two negative tokens is positively large, the dot-product between the embeddings of a positive and a negative tokens is negatively large. However, the dot-product between a neural token and another token is small. Furthermore, we could see that both the positive-positive, negative-negative token pairs



(a) Without Regularization



(b) With Regularization

Figure 5: Average attention score for positive tokens, negative tokens and neutral tokens. Model trained on Data I.

had larger dot-products than the positive-negative token pairs here, which was consistent with our aforementioned analysis.

C.3 Influence of $\boldsymbol{V}^\top \boldsymbol{W}$

We also examined the influence of the dot-product $\boldsymbol{V}^\top \boldsymbol{W}$ and found an apparent gap between the average attention scores of the positive tokens and that of the negative tokens on Data I as shown in Figure 5a. We placed regularization on $\boldsymbol{V}^\top \boldsymbol{W}$, and the gap almost disappeared as shown in Figure 5b, which indicated $\boldsymbol{V}^\top \boldsymbol{W}$ did have an impact on attention scores.

Additionally, we examined the dot-product term on synthetic datasets with different positive/negative instance ratios. It seems such a ratio has an impact on this term as shown in Figure 7: $\boldsymbol{V}^\top \boldsymbol{W}$ will be generally small on the balanced dataset compared to skewed datasets.

D Model with an Affine Input Layer

Let us consider a model with an affine input layer. The affine layer is added between the embedding layer and the attention layer. The variables will be

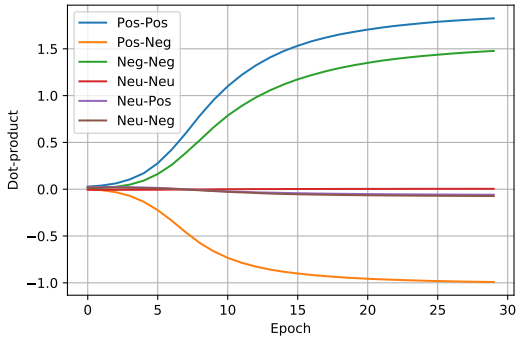


Figure 6: Six token pairs are selected to observe the trend of their dot-products during training on Data II: positive-positive pair, positive-negative pair, negative-negative pair, neutral-neutral pair, neutral-positive pair, neutral-negative pair. Tokens are randomly selected.

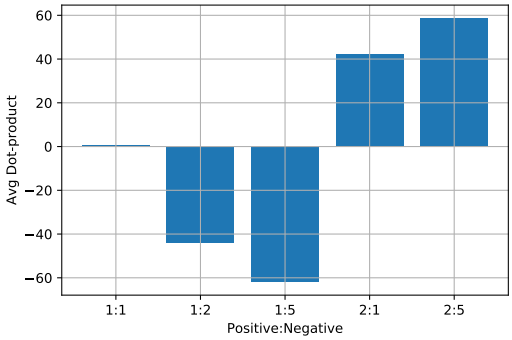


Figure 7: The dot-product of $\boldsymbol{V}^\top \boldsymbol{W}$ is averaged on the results from four random initializations. Five synthetic training sets with different positive/negative ratios are created.

described as:

$$\boldsymbol{h}_i = \mathbf{M}\boldsymbol{e}_i, \tag{56}$$

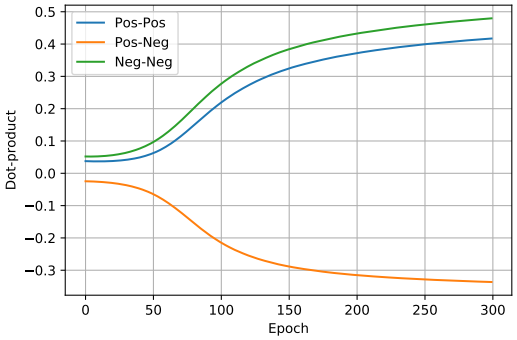
$$a_i = \frac{\boldsymbol{h}_i^\top \boldsymbol{V}}{\lambda}, \tag{57}$$

$$s_i = \boldsymbol{h}_i^\top \boldsymbol{W}, \tag{58}$$

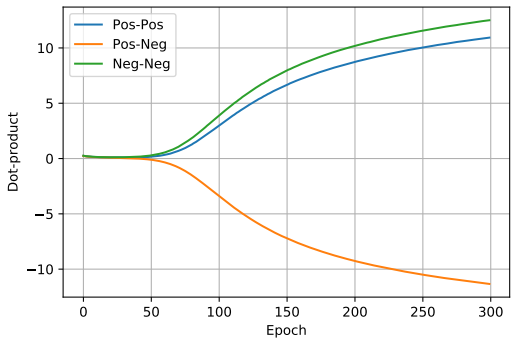
where $\mathbf{M} \in R^{d \times d_e}$ is the weight matrix of the affine layer, $\boldsymbol{e}_i \in R^{d_e}$. The gradients of the parameters $\boldsymbol{e}$,$\boldsymbol{V}$ and $\mathbf{M}$ on the entire dataset will be calculated as:

$$\frac{\partial \ell}{\partial \boldsymbol{e}} = -\frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \left(s_e - s^{(t)} \right) \frac{\mathbf{M}^\top \boldsymbol{V}}{\lambda} - \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \mathbf{M}^\top \boldsymbol{W}. \tag{59}$$

$$\frac{\partial \ell}{\partial \boldsymbol{V}} = -\frac{1}{m} \sum_{t=1}^{m} \sum_{j} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \left(s_j^{(t)} - s^{(t)} \right) \frac{\boldsymbol{h}_j^{(t)}}{\lambda}. \tag{60}$$



(a) Embedding Dot-product



(b) Affine Representation Dot-product

Figure 8: Embedding dot-products and affine representation dot-products between different type of tokens. The experiment is conducted on synthetic datasets with a SGD optimizer.

$$\frac{\partial \ell}{\partial \mathbf{M}} = -\frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right)\frac{\boldsymbol{V}(\boldsymbol{e}_j^{(t)})^\top}{\lambda} - \frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{W}(\boldsymbol{e}_j^{(t)})^\top. \tag{61}$$

The update of the polarity score can be represented as:

$$\begin{aligned}\frac{ds_e}{d\tau} &= (\frac{d\boldsymbol{h}_e}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau} \\ &= (\frac{d\mathbf{M}\boldsymbol{e}}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau} \\ &= (\frac{d\boldsymbol{e}}{d\tau})^\top (\mathbf{M})^\top \boldsymbol{W} + \boldsymbol{e}^\top (\frac{d\mathbf{M}}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau}.\end{aligned} \tag{62}$$

We know the properties of embedding dot-products in the aforementioned analysis. We also assume such properties exist for the token representations from the affine layer. Let us call them *affine representations* for tokens. We can see such patterns as shown in Figure 8.

The update of s_e can be described as below:

$$\begin{aligned}\frac{ds_e}{d\tau} &= \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}||\boldsymbol{W}^\top\mathbf{M}||_2^2 \\ &+ \frac{1}{m}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\boldsymbol{h}_e^\top\boldsymbol{h}^{(t)} \\ &+ \frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}||\boldsymbol{W}||_2^2 \\ &+ \frac{1}{m\lambda}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left(s_e - s^{(t)}\right)\boldsymbol{W'}^\top\boldsymbol{V'} \\ &+ \frac{1}{m\lambda}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right)\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}Q.\end{aligned} \tag{63}$$

where $\boldsymbol{W'} = \mathbf{M}^\top\boldsymbol{W}$, $\boldsymbol{V'} = \mathbf{M}^\top\boldsymbol{V}$, $Q = \boldsymbol{V}^T\boldsymbol{W}$.

Similar to the analysis in our paper, for positive tokens, the first three terms will likely be positive. For negative tokens, the first three terms will likely be negative. When λ is properly set and sufficiently large, the last two terms will be viewed as negligible. The first three terms will likely be dominant during update. Therefore, the polarity scores of positive tokens will likely end up with positively large values and the polarity scores of negative tokens will likely end up with negatively large values as shown in Figure 9. For neutral tokens, the polarity scores will likely end up being sufficiently small.

The update of the attention score can be written as:

$$\begin{aligned}\frac{da_e}{d\tau} &= \frac{1}{\lambda}(\frac{dh_e}{d\tau})^\top\boldsymbol{V} + \frac{1}{\lambda}h_e^\top\frac{d\boldsymbol{V}}{d\tau} \\ &= \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{||\boldsymbol{V}^\top\mathbf{M}||_2^2}{\lambda^2}(s_e - s) \\ &+ \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{\boldsymbol{W'}^\top\boldsymbol{V'}}{\lambda} \\ &+ \frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\frac{\boldsymbol{W}^\top\boldsymbol{V}}{\lambda} \\ &+ \frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right)\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\frac{||\boldsymbol{V}||_2^2}{\lambda^2} \\ &+ \frac{1}{m}\sum_{t=1}^{m}\sum_{j} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right)\frac{\boldsymbol{h}_e^\top\boldsymbol{h}_j^{(t)}}{\lambda^2},\end{aligned} \tag{64}$$

where $\boldsymbol{W'} = \mathrm{M}^\top \boldsymbol{W}$, $\boldsymbol{V'} = \mathrm{M}^\top \boldsymbol{V}$. From Equation 64, it can be seen that, if e is a polarity token with a strong polarity score, the term will likely be positively large as $y\beta\alpha_e(s_e - s) > 0$; the term will be reasonably small if e is a neutral token. The second and third terms have opposite effects on the positive and negative tokens with respect to attention scores. As for the last two terms, positive and negative instances have opposite effects which may lead to an offset, let us assume them to be reasonably small on a balanced and symmetric training set. Therefore, if the second and the third terms are sufficiently small, the attention scores of strong polarity tokens will likely receive positive updates during training and end up with large positive values as shown in Figure 9.

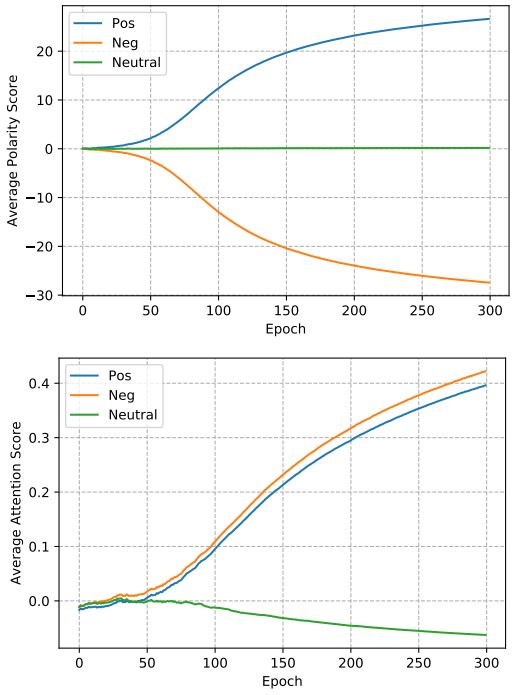


Figure 9: Top: the average polarity scores for positive, negative and neutral tokens respectively. Bottom: the average attention scores for positive, negative and neutral tokens respectively. The experiment is conducted on synthetic datasets with a SGD optimizer. The learning rate is fixed during training and regularization is placed upon $\boldsymbol{W'}^\top \boldsymbol{V'}$.

E Model with Additive Attention

Let us replace the scaled dot-product attention in the model discussed in the paper with additive attention. To make it more general, we also consider a scaling factor for the additive attention. The attention score at the i-th token of an instance will be calculated as:

$$\boldsymbol{x}_i = \mathbf{M}^a \boldsymbol{e}_i,\ a_i = \frac{\tanh(\boldsymbol{x}_i^\top)\boldsymbol{V}}{\lambda}. \tag{65}$$

where $\boldsymbol{e}_i \in R^{d_e}$, $\mathbf{M}^a \in R^{d \times d_e}$ is the weight matrix. The polarity score is calculated as $s_i = \boldsymbol{e}_i^\top \boldsymbol{W}$.

The derivative of the loss with respect to the embedding $\boldsymbol{e}_i$ will be calculated as:

$$\frac{\partial \ell}{\partial \boldsymbol{e}_i} = -y\beta\alpha_i \left[\frac{(\mathbf{M}^a)^\top \mathbf{D}_i \boldsymbol{V}(\boldsymbol{e}_i - \boldsymbol{h})^\top}{\lambda} + \mathbf{I} \right] \boldsymbol{W}, \tag{66}$$

where $\mathbf{D}_i = \mathrm{Diag}(1 - \tanh^2 \boldsymbol{x}_i)$, which is a diagonal matrix.

The derivative of the loss with respect to $\boldsymbol{V}$ will be calculated as:

$$\frac{\partial \ell}{\partial \boldsymbol{V}} = -\frac{y\beta}{\lambda} \sum_j \alpha_j s_j \left(\tanh \boldsymbol{x}_j - \sum_i \alpha_i \tanh \boldsymbol{x}_i \right). \tag{67}$$

And the derivative of the loss with respect to $\mathbf{M}^a$ will be:

$$\frac{\partial \ell}{\partial \mathbf{M}^a} = -\frac{y\beta}{\lambda} \sum_j \alpha_j s_j (\boldsymbol{e}_j \boldsymbol{V}^\top \mathbf{D}_j - \sum_i \alpha_i \boldsymbol{e}_i \boldsymbol{V}^\top \mathbf{D}_i)^\top. \tag{68}$$

The update of the polarity score can be written as:

$$\frac{ds_e}{d\tau} = (\frac{d\boldsymbol{e}}{d\tau})^\top \boldsymbol{W} + \boldsymbol{e}^\top \frac{d\boldsymbol{W}}{d\tau}. \tag{69}$$

This is similar to the scenario in the main paper, and we can have similar conclusions on polarity scores.

The update of the attention score will be described below:

$$\frac{da_e}{d\tau} = \frac{d\tanh(\boldsymbol{x}_e^\top)}{d\tau} \frac{\boldsymbol{V}}{\lambda} + \frac{\tanh(\boldsymbol{x}_e^\top)}{\lambda} \frac{d\boldsymbol{V}}{d\tau} \tag{70}$$

It is complex to handle the function $\tanh$ directly. We will approximate the function $\tanh$ using first-order Taylor series, namely $\tanh(x) \approx x$. The update of attention score on the entire training set

can be approximated as:

$$\frac{da_e}{d\tau} \approx \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \frac{||(\mathbf{M}^a)^\top \boldsymbol{V}||_2^2}{\lambda^2} (s_e - s^{(t)}) + \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} y^{(t)} \beta^{(t)} \alpha_j^{(t)} \frac{\boldsymbol{W}^\top (\mathbf{M}^a)^\top \boldsymbol{V}}{\lambda} + \frac{1}{m} \sum_{t=1}^{m} \sum_j y^{(t)} \beta^{(t)} \alpha_j^{(t)} \left(s_j^{(t)} - s^{(t)} \right) \boldsymbol{e}^\top \boldsymbol{e}_j^{(t)} \frac{||\boldsymbol{V}||_2^2}{\lambda^2} + \frac{1}{m} \sum_{t=1}^{m} \sum_j y^{(t)} \beta^{(t)} \alpha_j^{(t)} \left(s_j^{(t)} - s^{(t)} \right) \frac{\boldsymbol{x}^\top \boldsymbol{x}_j}{\lambda^2}. \tag{71}$$

We can have similar analysis on the terms. The first term will likely be positive when approaching a local minimum for polarity tokens. The second term can have opposite effects on positive and negative tokens. We assume that $\boldsymbol{x}^\top \boldsymbol{x}_j$ has similar trends and patterns to the embedding dot-products. The last two terms will likely be reasonably small as there will be offsets between positive and negative instances. Therefore, when the second term is small or λ is appropriately set, the derivative of the attention score will likely be dominated by the first term. This indicates the tokens with strong polarity scores will likely gain incremental values during training and end up with large positive values as shown in Figure 10.

F Multi-class Classification

The architecture for multi-class classification is similar to the one in the main paper. But the last linear layer will be modified to project the hidden states into a K-dimension vector, the sigmoid layer will be replaced by a *softmax* layer.

The instance-level polarity scores[1] will be calculated as:

$$\boldsymbol{s} = (\mathbf{W}^h)^\top \boldsymbol{h}, \tag{72}$$

where $\boldsymbol{h}$ is the instance representation, $\mathbf{W}^h \in R^{d \times K}$ is the weight matrix of the final linear layer, K is the class number. $\boldsymbol{s}$ is a vector consisting of K elements, each of which corresponds to a specific label.

We can get the probability distribution for all the labels:

$$\boldsymbol{p} = \text{Softmax}(\boldsymbol{s}), \tag{73}$$

[1]For simplicity, we do not consider the bias in the linear layer as we did in the paper.

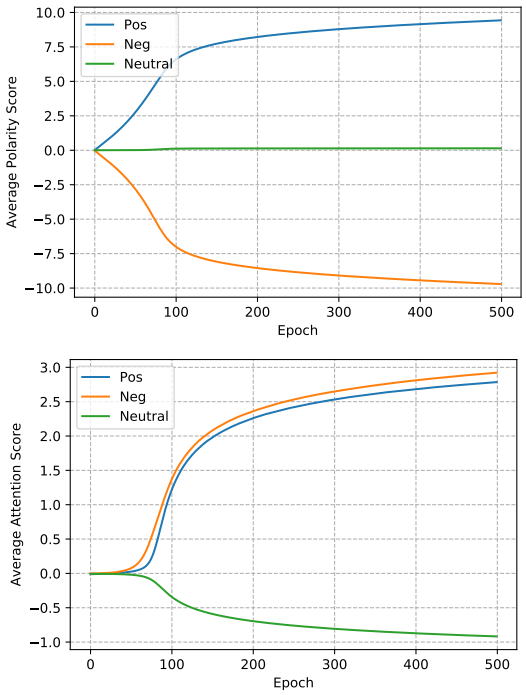


Figure 10: Top: the average polarity scores for positive, negative and neutral tokens respectively. Bottom: the average attention scores for positive, negative and neutral tokens respectively. The experiment is conducted on synthetic datasets with a SGD optimizer. The learning rate is fixed during training. Additive attention is used here.

where $\boldsymbol{p} \in R^K$.

Cross-entropy loss will be used, and the loss on an instance with the label $k \in (0, 1, .., K-1)$ can be calculated as:

$$\ell = -\log p_k, \tag{74}$$

where p_k refers to the corresponding probability for the label k.

The partial derivative of the loss ℓ with respect to $\boldsymbol{h}$ will be calculated as:

$$\frac{\partial \ell}{\partial \boldsymbol{h}} = \mathbf{W}^h \hat{\boldsymbol{p}}, \quad \hat{\boldsymbol{p}} = \boldsymbol{p} - \boldsymbol{I}_k, \tag{75}$$

where $\boldsymbol{I}_k \in R^K$ is the one-hot encoding for the k-th label. Note that the elements of the vector $\hat{\boldsymbol{p}} \in R^K$ are positive except the one that corresponds to the ground truth label.

Given an instance with the label k, we can have such partial derivatives:

$$\frac{\partial \ell}{\partial \boldsymbol{s}} = \hat{\boldsymbol{p}}, \tag{76}$$

$$\frac{\partial \ell}{\partial \mathbf{W}^h} = \boldsymbol{h} \hat{\boldsymbol{p}}^\top. \tag{77}$$

On the entire dataset, the partial derivatives will be calculated as:

$$\frac{\partial \ell}{\partial \boldsymbol{e}} = \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} \alpha_j^{(t)} \left[\frac{\boldsymbol{V}(\boldsymbol{e} - \boldsymbol{h}^{(t)})^\top}{\lambda} + \mathbf{I} \right] \mathbf{W}^h \hat{\boldsymbol{p}}^{(t)}, \tag{78}$$

$$\frac{\partial \ell}{\partial \mathbf{W}^h} = \frac{1}{m} \sum_{t=1}^{m} \boldsymbol{h}^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top. \tag{79}$$

On the entire training set, the gradient of $\boldsymbol{V}$ can be calculated as:

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{V}} &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(\boldsymbol{e}_j^{(t)} - \boldsymbol{h}^{(t)} \right)^\top \mathbf{W}^h \hat{\boldsymbol{p}}^{(t)} \\ &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(\boldsymbol{s}_j^{(t)} - \boldsymbol{s}^{(t)} \right)^\top \hat{\boldsymbol{p}}^{(t)}. \end{aligned} \tag{80}$$

Accordingly, given a token e, the token-level polarity scores will be calculated as:

$$\boldsymbol{s}_e = (\mathbf{W}^h)^\top \boldsymbol{e}. \tag{81}$$

Note that $\boldsymbol{s}_e$ is a vector that has K elements, each corresponds the polarity score with respect to the label.

The update of $\boldsymbol{s}_e$ will be written as:

$$\begin{aligned} \frac{d\boldsymbol{s}_e}{d\tau} &= \frac{d(\mathbf{W}^h)^\top}{d\tau} \boldsymbol{e} + (\mathbf{W}^h)^\top \frac{d\boldsymbol{e}}{d\tau} \\ &= -\frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} (\mathbf{W}^h)^\top \boldsymbol{V} \left(\boldsymbol{s}_e - \boldsymbol{s}^{(t)} \right)^\top \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} (\mathbf{W}^h)^\top \mathbf{W}^h \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m} \sum_{t=1}^{m} \hat{\boldsymbol{p}}^{(t)} (\boldsymbol{h}^{(t)})^\top \boldsymbol{e}. \end{aligned} \tag{82}$$

Similar to the analysis in our paper, when λ is properly set, the first term above can be reasonably small and negligible. For the second term above, $(\mathbf{W}^h)^\top \mathbf{W}^h$ is a symmetric matrix with the diagonal elements as the corresponding squared L_2-norms of column weight vectors. We can assume that the row weight vectors in $\mathbf{W}^h$ are different from each other. So the dot-products between two different row weight vectors are relatively small (even negative) compared to the squared L_2-norms of row weight vectors. Hence, for the second term, the dimension that corresponds to the ground truth label will likely gain incremental values during training. It is also the case for the third term. Therefore, if e is strongly associated with the label k, its polarity score corresponding to the label k will be likely increasing during training and end up with a large positive value.

And the update of a_e wil be written as:

$$\begin{aligned} \frac{da_e}{d\tau} &= \frac{1}{\lambda} \boldsymbol{e}^\top \frac{d\boldsymbol{V}}{d\tau} + \frac{1}{\lambda} (\frac{d\boldsymbol{e}}{d\tau})^\top \boldsymbol{V} \\ &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}^\top \boldsymbol{e}_j^{(t)} \left(\boldsymbol{s}_j^{(t)} - \boldsymbol{s}^{(t)} \right)^\top \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} \alpha_j^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top \frac{\left(\boldsymbol{s}_e - \boldsymbol{s}^{(t)} \right) ||\boldsymbol{V}||_2^2}{\lambda} \\ &\quad - \frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} \alpha_j^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top (\mathbf{W}^h)^\top \boldsymbol{V}. \end{aligned} \tag{83}$$

Still similar to the analysis for the model discussed in the paper. The first term can be assumed to be small and negligible, the second term will likely give incremental values to the attention scores for the polarity tokens during training, and the last term has opposite impacts on the positive and the negative tokens respectively.

If the second term is dominant for strong polarity tokens, the attention score a_e will likely be increasing and end up with relatively large positive values. Figures 11, 12 supports our analysis.

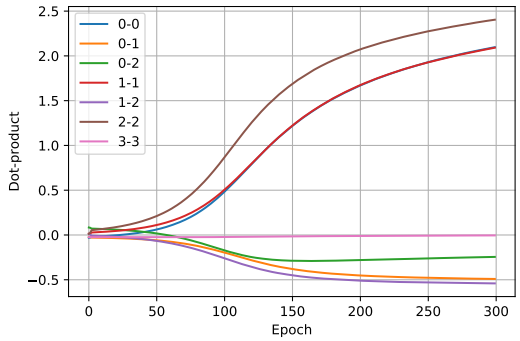


Figure 11: The dot-products between polarity tokens from the same group are larger than those between tokens from different groups. A SGD optimizer is adopted and the parameters are randomly initialized, the learning rate is not decayed. '0', '1', '2' and '3' refer to the token associated with the Label '0', '1', '2' and '3' respectively. Particularly, Label '3' refers to the neutral tokens that appear evenly across different types of instances.

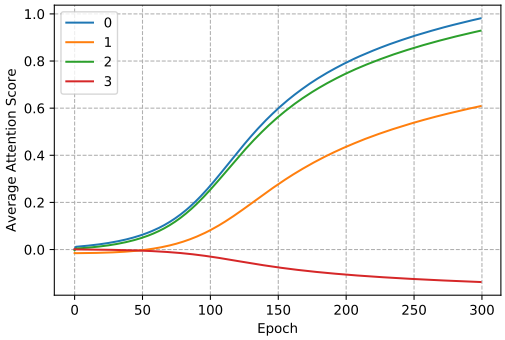


Figure 12: The average attention scores of the polarity tokens are increasing during training. A SGD optimizer is adopted and the parameters are randomly initialized, the learning rate is not decayed. '0', '1', '2' and '3' refer to the token associated with the Label '0', '1', '2' and '3' respectively.

Dataset	Token Type	Num	Sample
SST	Positive	412	delicious, ingenious, heartbreaking, ahead, slightly, world, hilarious, remarkably, superb, joyous
	Negative	265	mediocre, bad, failure, thinks, poor, preachy, sustain, bears, sink, lazy, boring, worst, plain, sappy
	Neutral	1384	supply, figured, goodwill, induces, formal, experimental, shots, unrelenting, lost, course, ensues
IMDB	Positive	2163	packs, ullman, undeniable, beauty, beery, lightly, mat, bizet, busts, sampson, adulthood, korda
	Negative	1900	appleby, jenkins, hacks, objectively, titillation, cannibal, violated, drilling, payroll, hallucination, brinke
	Neutral	5968	quiroz, gauzy, mysticism, hoards, tail, chopping, milk, reminiscing, hungrily, leftists, trial, disturbed
20News I	Positive	957	hockey, bay, powerplay, oilers,pocklington, l friend, vincent, loke, skrudland, andy, zubov, carried
	Negative	557	uniform, writes, cuyler, friday, leagues, batted, ability, judgement, steph, umpire, hernandez, outs, edge
	Neutral	1011	means, having, code, mind, barrier, virtually, file, stir, ohio, shelled, rubber, toward, pushing, offensively
20News II	Motor	49	riders, bmw, rider, ground, exhaust, plastic, car, road, driving, traffic, riding, speed, lock, jacket, bikes
	Med	268	blood, calcium, page, approach, sinus, food, magnesium, photography, robert, gordon, typing, injuries
	Guns	192	homicides, incident, laws, control, enforcement, fired, passed, cult, majority, defend, themselves
	Neutral	164	planning, considering, yet, price, than, boy, really, stopping, face, race, glad, cut, doing, saw, correctly

Table 1: Example selected tokens for each dataset. The tokens are chosen based on their association with specific labels on the training set, and can be affected by the distribution of the training data. They may bot be consistent with our human's perspective.

G More Results from Experiments on Real Datasets

We conducted more experiments on real datasets. One set of results were reported for each model. Adagrad optimizers were used here. [2] The tokens were selected based on their association with instance labels as shown in Table 1.

G.1 LSTM encoder

We defined two types of token-level polarity and attention scores for models with LSTM encoders:

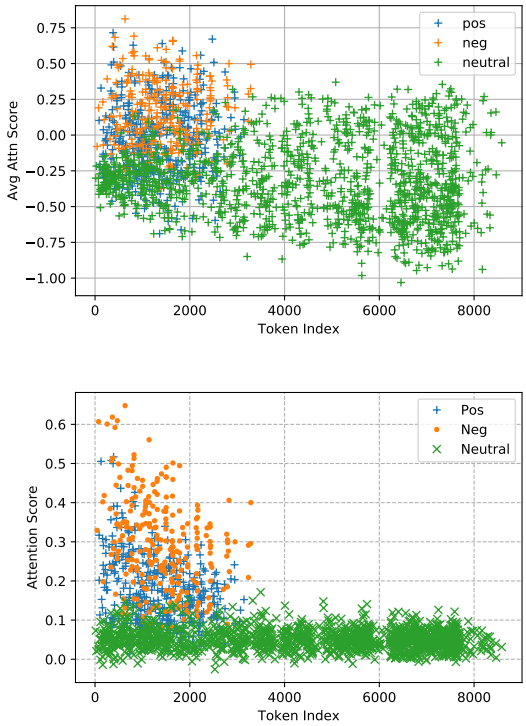


Figure 13: Top: the average local attention score for each selected token. Bottom: the global attention score for each selected token. A bidirectional LSTM encoder is adopted in the model with scaled dot-product attention, the scaling factor is 10. The experiment is conducted on SST with an Adagrad optimizer.

local polarity/attention score and *global polarity/attention score*.

We calculated the local polarity scores and local attention scores for the tokens in each instance based on the LSTM hidden states. We averaged the attention scores for each token in the vocabulary across all the training instances. We fed each token in the vocabulary to the model directly, with the single output representation we could obtain the corresponding global polarity score and global attention score. From Figure 13, it can be seen that the polarity tokens still generally have larger average local attention scores than the neutral tokens. However, the pattern seems more apparent for the global attention scores.

G.2 Other Results

On top of the scaled dot-product attention discussed in the paper, we investigated whether similar patterns could be found on more models under different settings. Note that DP is the basic model, DP-L is the model with a LSTM encoder[3], DP-A

[2] We also conducted experiments using Adam optimizers, the patterns were still similar.

[3] L_2-regularizations were applied to the affine layer and the LSTM layer respectively.

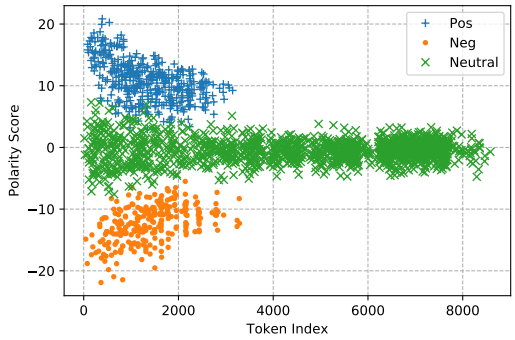


Figure 14: SST (DP-A, λ=20)

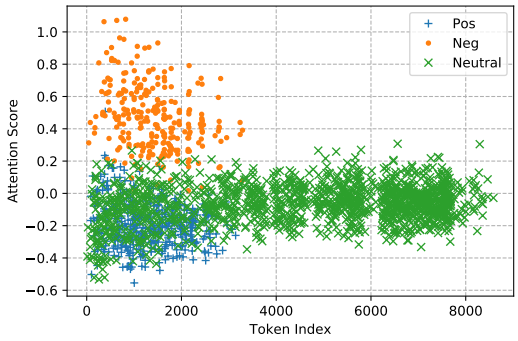


Figure 15: SST (DP-A, λ=20)

is the model with an affine encoder. AD refers to the model with additive attention. Similar patterns could be found on different models and different scaling factors from Figure 14 to 28:

References

Xiaobing Sun and Wei Lu. 2020. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3418–3428.

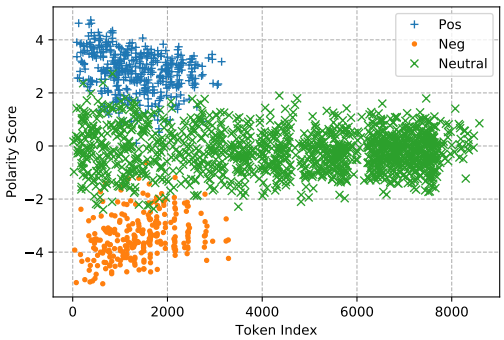


Figure 16: SST (DP-L, λ=10)

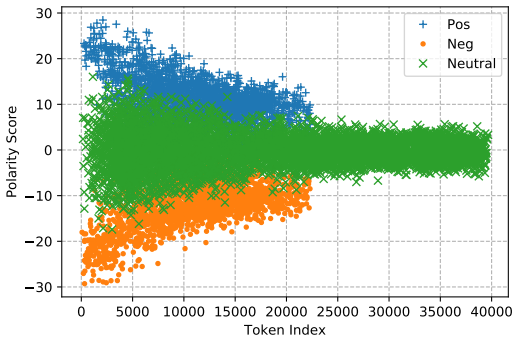


Figure 17: IMDB (DP-A, λ=10)

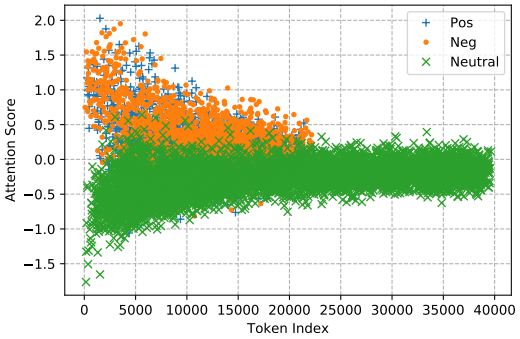


Figure 18: IMDB (DP-A, λ=10)

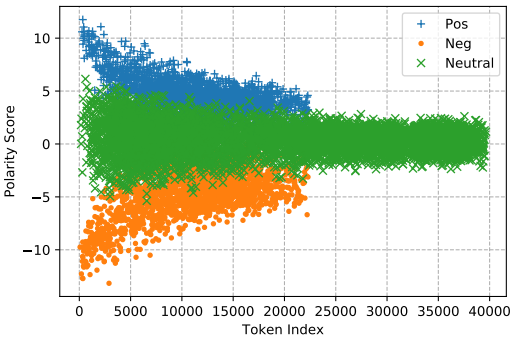


Figure 19: IMDB (AD, λ=10)

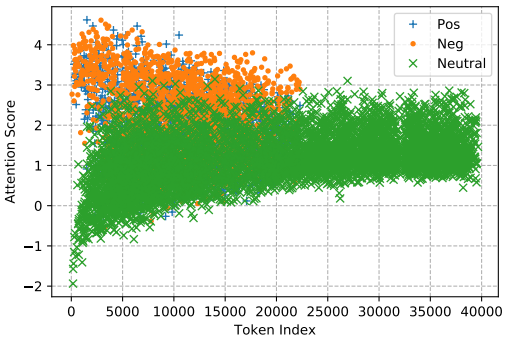


Figure 20: IMDB (AD, λ=10)

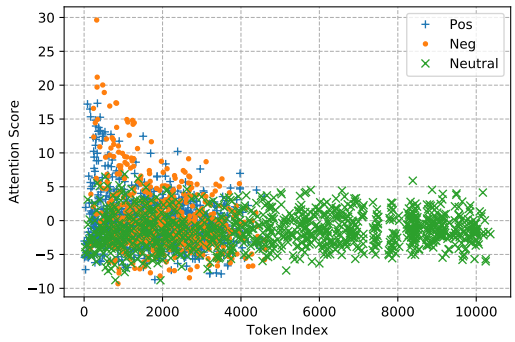


Figure 21: 20News I (DP-A, λ=1)

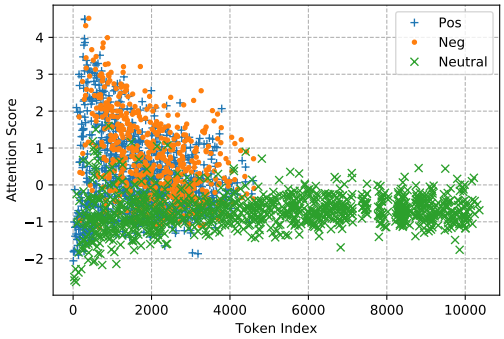


Figure 22: 20News I (DP-A, λ=10)

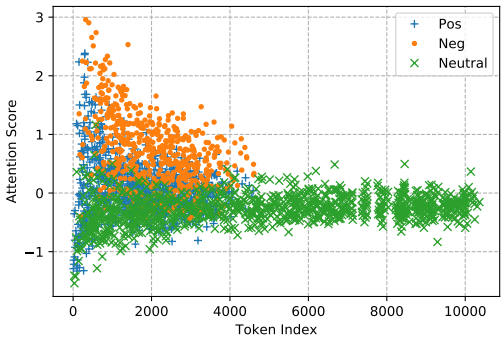


Figure 23: 20News I (DP-A, λ=20)

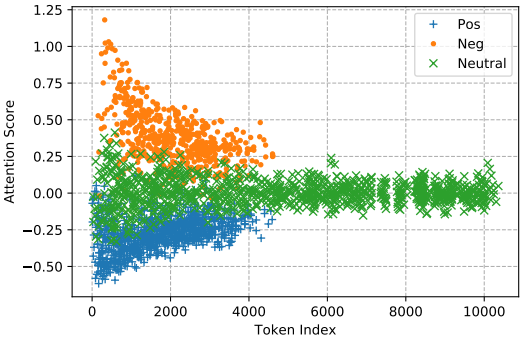


Figure 24: 20News I (DP-A, λ=50)

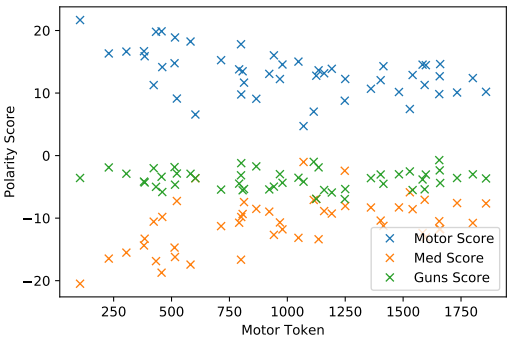


Figure 25: 20News II (DP-A, λ=10)

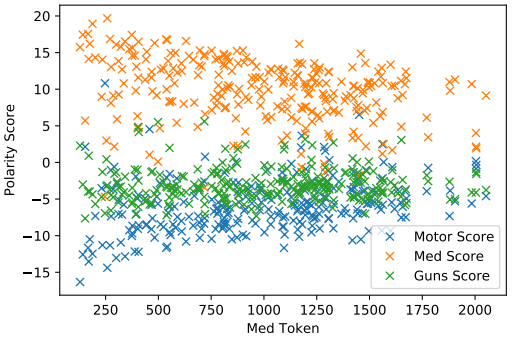


Figure 26: 20News II (DP-A, λ=10)

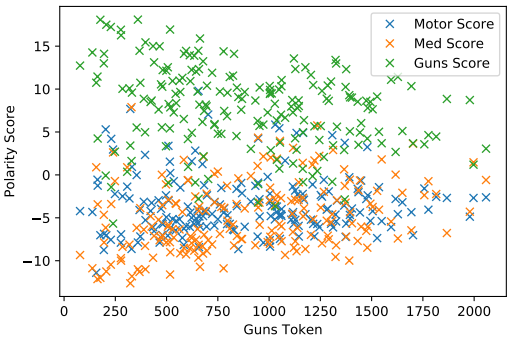


Figure 27: 20News II (DP-A, λ=10)

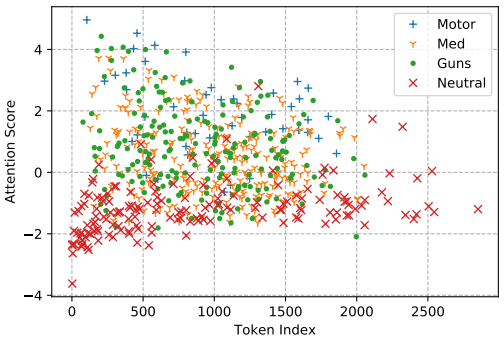


Figure 28: 20News II (DP-A, λ=10)