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. The gradient on the entire training set can be calculated as the average of the gradients on each training instance.

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

$$\begin{aligned}\frac{\partial \ell}{\partial s} &= -y\beta, \\ \frac{\partial \ell}{\partial \boldsymbol{W}} &= \frac{\partial \ell}{\partial s}\frac{\partial s}{\partial \boldsymbol{W}} \\ &= -y\beta\boldsymbol{h}, \\ \frac{\partial \ell}{\partial \boldsymbol{h}} &= \frac{\partial \ell}{\partial s}\frac{\partial s}{\partial \boldsymbol{h}} \\ &= -y\beta\boldsymbol{W},\end{aligned} \tag{1}$$

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:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \frac{\partial \sum_j \alpha_j \boldsymbol{h}_j}{\partial \boldsymbol{h}_i} \\ &= \sum_j \frac{\partial \alpha_j}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I},\end{aligned} \tag{2}$$

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. The partial derivative of α_j with respect to $\boldsymbol{h}_i$ can be represented as:

$$\frac{\partial \alpha_j}{\partial \boldsymbol{h}_i} = \frac{\partial \alpha_j}{\partial a_i}\frac{\partial a_i}{\partial \boldsymbol{h}_i} = \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}, \tag{3}$$

where δ_{ji} is the Kronecker delta function:

$$\delta_{ji} = \begin{cases} 1 & j = i \\ 0 & else. \end{cases} \tag{4}$$

Referring to Equation 3 and 4 , we can obtain the partial derivative of $\boldsymbol{h}$ with respect to $\boldsymbol{h}_i$ as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \sum_j \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I} \\ &= \alpha_i \left[\frac{\partial a_i}{\partial \boldsymbol{h}_i}(\boldsymbol{h}_i - \boldsymbol{h})^\top + \mathbf{I}\right].\end{aligned} \tag{5}$$

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

$$\begin{aligned}a_i &= \frac{\boldsymbol{h}_i^\top \boldsymbol{V}}{\lambda}, \\ \frac{\partial a_i}{\partial \boldsymbol{h}_i} &= \frac{\boldsymbol{V}}{\lambda},\end{aligned} \tag{6}$$

where $\boldsymbol{V} \in \mathbb{R}^d$ is the context vector. With Equation 5 and 6, we can have:

$$\begin{aligned}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i} &= \sum_j \alpha_j(\delta_{ji} - \alpha_i)\frac{\partial a_i}{\partial \boldsymbol{h}_i}\boldsymbol{h}_j^\top + \alpha_i \mathbf{I} \\ &= \alpha_i \left[\frac{\boldsymbol{V}(\boldsymbol{h}_i - \boldsymbol{h})^\top}{\lambda} + \mathbf{I}\right].\end{aligned} \tag{7}$$

With Equation 1 and 7, we can obtain the partial derivative of the loss with respect to $\boldsymbol{h}_i$:

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{h}_i} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{h}_i}\frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= -y\beta\alpha_i \left[\frac{\boldsymbol{V}(\boldsymbol{h}_i - \boldsymbol{h})^\top}{\lambda} + \mathbf{I}\right]\boldsymbol{W}.\end{aligned} \tag{8}$$

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

$$\frac{\partial \ell}{\partial \boldsymbol{e}} = -\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\left[\frac{\boldsymbol{V}(\boldsymbol{e} - \boldsymbol{h}^{(t)})^\top}{\lambda} + \mathbf{I}\right]\boldsymbol{W}, \tag{9}$$

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:

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{V}} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{V}}\frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= \sum_j \frac{\partial \alpha_j \boldsymbol{h}_j}{\partial \boldsymbol{V}}\frac{\partial \ell}{\partial \boldsymbol{h}} \\ &= \sum_j \frac{\partial \alpha_j}{\partial \boldsymbol{V}}\boldsymbol{h}_j^\top\frac{\partial \ell}{\partial \boldsymbol{h}},\end{aligned} \tag{10}$$

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:

$$\begin{aligned}\frac{\partial \alpha_j}{\partial \boldsymbol{V}} &= \sum_i \frac{\partial \alpha_j}{\partial a_i}\frac{\partial a_i}{\partial \boldsymbol{V}} \\ &= \sum_i \alpha_j(\delta_{ji} - \alpha_i)\frac{\boldsymbol{h}_i}{\lambda} \\ &= \frac{\alpha_j}{\lambda}(\boldsymbol{h}_j - \boldsymbol{h}).\end{aligned} \tag{11}$$

Substitute Equation 11 into Equation 10, and we will obtain the following:

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{V}} &= -\frac{y\beta}{\lambda}\sum_j \alpha_j(\boldsymbol{h}_j - \boldsymbol{h})\boldsymbol{h}_j^\top \boldsymbol{W} \\ &= -\frac{y\beta}{\lambda}[\sum_j \alpha_j \boldsymbol{h}_j \boldsymbol{h}_j^\top - \boldsymbol{h}\boldsymbol{h}^\top]\boldsymbol{W} \\ &= -\frac{y\beta}{\lambda}\sum_j \alpha_j \boldsymbol{h}_j(\boldsymbol{h}_j - \boldsymbol{h})^\top \boldsymbol{W}.\end{aligned} \tag{12}$$

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:

$$\begin{aligned}&\frac{\partial \ell}{\partial \boldsymbol{V}} = \\ &-\frac{1}{m\lambda}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\sum_j \alpha_j^{(t)}\boldsymbol{e}_j^{(t)}\left(\boldsymbol{e}_j^{(t)} - \boldsymbol{h}^{(t)}\right)^\top \boldsymbol{W} \\ &= -\frac{1}{m\lambda}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\sum_j \alpha_j^{(t)}\boldsymbol{e}_j^{(t)}\left(s_j^{(t)} - s^{(t)}\right),\end{aligned} \tag{13}$$

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.

B 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 described as:

$$\boldsymbol{h}_i = \mathbf{W}^e \boldsymbol{e}_i, \tag{14}$$

$$a_i = \frac{\boldsymbol{h}_i^\top \boldsymbol{V}}{\lambda}, \tag{15}$$

$$s_i = \boldsymbol{h}_i^\top \boldsymbol{W}, \tag{16}$$

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

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{e}} =& -\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_e - s^{(t)})\frac{(\mathbf{W}^e)^\top \boldsymbol{V}}{\lambda} \\ &- \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}(\mathbf{W}^e)^\top \boldsymbol{W}, \\ \frac{\partial \ell}{\partial \mathbf{W}^e} =& -\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)} - s^{(t)})\frac{\boldsymbol{V}(\boldsymbol{e}_j^{(t)})^\top}{\lambda} \\ &- \frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{W}(\boldsymbol{e}_j^{(t)})^\top, \\ \frac{\partial \ell}{\partial \boldsymbol{V}} =& -\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)} - s^{(t)})\frac{\boldsymbol{h}_j^{(t)}}{\lambda}.\end{aligned} \tag{17}$$

The update of the polarity score can be represented as:

$$\begin{aligned}\frac{ds_e}{d\tau} &= (\frac{d\boldsymbol{h}_e}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau} \\ &= (\frac{d\mathbf{W}^e \boldsymbol{e}}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau} \\ &= (\frac{d\boldsymbol{e}}{d\tau})^\top (\mathbf{W}^e)^\top \boldsymbol{W} + \boldsymbol{e}^\top (\frac{d\mathbf{W}^e}{d\tau})^\top \boldsymbol{W} + \boldsymbol{h}_e^\top \frac{d\boldsymbol{W}}{d\tau}.\end{aligned} \tag{18}$$

The update of s_e can be described as below:

$$\begin{aligned}\frac{ds_e}{d\tau} =& \frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}||\boldsymbol{W}^\top \mathbf{W}^e||_2^2 \\ &+ \frac{1}{m}\sum_{t=1}^{m} y^{(t)}\beta^{(t)}\boldsymbol{h}_e^\top \boldsymbol{h}^{(t)} \\ &+ \frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)}||\boldsymbol{W}||_2^2 \\ &+ \frac{1}{m\lambda}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_e - s^{(t)})\boldsymbol{W'}^\top \boldsymbol{V'} \\ &+ \frac{1}{m\lambda}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)} - s^{(t)})\boldsymbol{e}^\top \boldsymbol{e}_j^{(t)} Q.\end{aligned} \tag{19}$$

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

The update of the attention score can be written as:

$$\begin{aligned}\frac{da_e}{d\tau} &= \frac{1}{\lambda}(\frac{dh_e}{d\tau})^\top \boldsymbol{V} + \frac{1}{\lambda}h_e^\top \frac{d\boldsymbol{V}}{d\tau} \\ &= (\frac{d\boldsymbol{e}}{d\tau})^\top (\mathbf{W}^e)^\top \frac{\boldsymbol{V}}{\lambda} + \boldsymbol{e}^\top (\frac{d\mathbf{W}^e}{d\tau})^\top \frac{\boldsymbol{V}}{\lambda} \\ &+ \frac{1}{\lambda}\boldsymbol{h}_e^\top \frac{d\boldsymbol{V}}{d\tau}.\end{aligned} \tag{20}$$

Let us expand the terms as:

$$\begin{aligned}\frac{da_e}{d\tau}=&\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{||\boldsymbol{V}^\top\mathbf{W}^e||_2^2}{\lambda^2}(s_e-s)\\&+\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{\boldsymbol{W'}^\top\boldsymbol{V'}}{\lambda}\\&+\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\frac{\boldsymbol{W}^\top\boldsymbol{V}}{\lambda}\\&+\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)}-s^{(t)})\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\frac{||\boldsymbol{V}||_2^2}{\lambda^2}\\&+\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)}-s^{(t)})\frac{\boldsymbol{h}_e^\top\boldsymbol{h}_j^{(t)}}{\lambda^2},\end{aligned} \tag{21}$$

where $\boldsymbol{W'}=(\mathbf{W}^e)^\top\boldsymbol{W}$, $\boldsymbol{V'}=(\mathbf{W}^e)^\top\boldsymbol{V}$.

C Model with Additive Attention

Additive attention is commonly used for text classifications. 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:

$$a_i=\tanh\Big(\boldsymbol{e}_i^\top\mathbf{W}^a+\boldsymbol{b}^a\Big)\boldsymbol{V}/\lambda, \tag{22}$$

where $\mathbf{W}^a\in R^{d\times d}$ is the weight matrix, $\boldsymbol{b}^a\in R^d$ is the bias for the attention layer. For simplicity, we do not consider the bias here, and the attention score will be:

$$a_i=\frac{\tanh\big(\boldsymbol{e}_i^\top\mathbf{W}^a\big)\boldsymbol{V}}{\lambda}. \tag{23}$$

Let us use $\boldsymbol{x}_i$ to denote $\boldsymbol{e}_i^\top\mathbf{W}^a$, and:

$$a_i=\frac{\tanh(\boldsymbol{x}_i)\boldsymbol{V}}{\lambda}. \tag{24}$$

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

$$\frac{\partial\ell}{\partial\boldsymbol{e}_i}=-y\beta\alpha_i\left[\frac{\mathbf{W}^a\mathbf{D}_i\boldsymbol{V}(\boldsymbol{h}_i-\boldsymbol{h})^\top}{\lambda}+\mathbf{I}\right]\boldsymbol{W}, \tag{25}$$

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:

$$\frac{\partial\ell}{\partial\boldsymbol{V}}=-\frac{y\beta}{\lambda}\sum_j\alpha_j s_j(\tanh\boldsymbol{x}_j^\top-\sum_i\alpha_i\tanh\boldsymbol{x}_i^\top). \tag{26}$$

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

$$\frac{\partial\ell}{\partial\mathbf{W}^a}=-\frac{y\beta}{\lambda}\sum_j\alpha_j s_j(\boldsymbol{e}_j\boldsymbol{V}^\top\mathbf{D}_j-\sum_i\alpha_i\boldsymbol{e}_i\boldsymbol{V}^\top\mathbf{D}_i). \tag{27}$$

The update of the polarity score can be written as:

$$\frac{ds_e}{d\tau}=(\frac{d\boldsymbol{e}}{d\tau})^\top\boldsymbol{W}+\boldsymbol{e}^\top\frac{d\boldsymbol{W}}{d\tau}. \tag{28}$$

When the scaling factor λ is large enough, the derivative can be approximated in a way similar to the model with scaled dot-product attention. And we will make similar predictions on the trends of the polarity tokens as well as the neutral tokens.

However, the update of the attention score will be different from the scaled dot-product one as below:

$$\frac{da_e}{d\tau}=\frac{d\tanh\boldsymbol{x}_e^\top}{d\tau}\frac{\boldsymbol{V}}{\lambda}+\frac{\tanh\boldsymbol{x}_e^\top}{\lambda}\frac{d\boldsymbol{V}}{d\tau} \tag{29}$$

It is complex to handle the function $\tanh$ directly.

Alternatively, we can 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:

$$\begin{aligned}\frac{da_e}{d\tau}&\approx(\frac{d\boldsymbol{x}_e}{d\tau})^\top\boldsymbol{V}+\boldsymbol{x}_e^\top\frac{d\boldsymbol{V}}{d\tau}\\&=(\frac{d\boldsymbol{e}}{d\tau})^\top\mathbf{W}^a\frac{\boldsymbol{V}}{\lambda}+\boldsymbol{e}^\top\frac{d\mathbf{W}^a}{d\tau}\frac{\boldsymbol{V}}{\lambda}\\&+\frac{1}{\lambda}\boldsymbol{e}^\top\mathbf{W}^a\frac{d\boldsymbol{V}}{d\tau}\\&\approx\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{||\mathbf{W}^a\boldsymbol{V}||_2^2}{\lambda^2}(s_e-s^{(t)})\\&+\frac{1}{m}\sum_{(t,j):e_j^{(t)}\equiv e} y^{(t)}\beta^{(t)}\alpha_j^{(t)}\frac{\boldsymbol{W}^\top\mathbf{W}^a\boldsymbol{V}}{\lambda}\\&+\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)}-s^{(t)})\boldsymbol{e}^\top\boldsymbol{e}_j^{(t)}\frac{||\boldsymbol{V}||_2^2}{\lambda^2}\\&+\frac{1}{m}\sum_{t=1}^{m}\sum_j y^{(t)}\beta^{(t)}\alpha_j^{(t)}(s_j^{(t)}-s^{(t)})\frac{\boldsymbol{x}\boldsymbol{x}_j^\top}{\lambda^2},\end{aligned} \tag{30}$$

where $\boldsymbol{x}=\boldsymbol{e}^\top\mathbf{W}^a$.

D Multi-class Classification

The architecture for multi-class classification is similar to the one discussed in our paper except that: the last linear layer weight will be a matrix instead of a vector, and the sigmoid activation will be replaced by a softmax function. The token representation is the embedding.

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

$$\boldsymbol{s} = (\mathbf{W}^h)^\top \boldsymbol{h}, \tag{31}$$

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 element will be regarded as the polarity score with respect to its corresponding label. A desirable scenario is such: if the instance is with the label k, then the k-th element in $\boldsymbol{s}$ will be as large as possible whereas the other elements will be as small as possible .

We can get the probability distribution for all the labels:

$$\boldsymbol{p} = \mathrm{Softmax}(\boldsymbol{s}), \tag{32}$$

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:

$$\ell = -\log p_k, \tag{33}$$

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:

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{h}} &= \mathbf{W}^h \hat{\boldsymbol{p}}, \\ \hat{\boldsymbol{p}} &= \boldsymbol{p} - \boldsymbol{I}_k, \end{aligned} \tag{34}$$

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:

$$\frac{\partial \ell}{\partial \boldsymbol{s}} = \hat{\boldsymbol{p}}, \tag{35}$$

$$\frac{\partial \ell}{\partial \mathbf{W}^h} = \boldsymbol{h}\hat{\boldsymbol{p}}^\top. \tag{36}$$

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

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

$$\frac{\partial \ell}{\partial \mathbf{W}^h} = \frac{1}{m} \sum_{t=1}^{m} \boldsymbol{h}^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top. \tag{38}$$

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

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{V}} &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(\boldsymbol{e}_j^{(t)} - \boldsymbol{h}^{(t)} \right)^\top \mathbf{W}^h \hat{\boldsymbol{p}}^{(t)} \\ &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}_j^{(t)} \left(\boldsymbol{s}_j^{(t)} - \boldsymbol{s}^{(t)} \right)^\top \hat{\boldsymbol{p}}^{(t)}. \end{aligned} \tag{39}$$

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

$$\boldsymbol{s}_e = (\mathbf{W}^h)^\top \boldsymbol{e}. \tag{40}$$

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:

$$\begin{aligned} \frac{d\boldsymbol{s}_e}{d\tau} &= \frac{d(\mathbf{W}^h)^\top}{d\tau} \boldsymbol{e} + (\mathbf{W}^h)^\top \frac{d\boldsymbol{e}}{d\tau} \\ &= -\frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} (\mathbf{W}^h)^\top \boldsymbol{V} (\boldsymbol{s}_e - \boldsymbol{s}^{(t)})^\top \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m} \sum_{(t,j):e_j^{(t)} \equiv e} (\mathbf{W}^h)^\top \mathbf{W}^h \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m} \sum_{t=1}^{m} \hat{\boldsymbol{p}}^{(t)} (\boldsymbol{h}^{(t)})^\top \boldsymbol{e}. \end{aligned} \tag{41}$$

And the update of a_e wil be written as:

$$\begin{aligned} \frac{da_e}{d\tau} &= \frac{1}{\lambda} \boldsymbol{e}^\top \frac{d\boldsymbol{V}}{d\tau} + \frac{1}{\lambda} (\frac{d\boldsymbol{e}}{d\tau})^\top \boldsymbol{V} \\ &= \frac{1}{m\lambda} \sum_{t=1}^{m} \sum_{j} \alpha_j^{(t)} \boldsymbol{e}^\top \boldsymbol{e}_j^{(t)} \left(\boldsymbol{s}_j^{(t)} - \boldsymbol{s}^{(t)} \right)^\top \hat{\boldsymbol{p}}^{(t)} \\ &\quad - \frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} \alpha_j^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top \frac{(\boldsymbol{s}_e - \boldsymbol{s}^{(t)}) ||\boldsymbol{V}||_2^2}{\lambda} \\ &\quad - \frac{1}{m\lambda} \sum_{(t,j):e_j^{(t)} \equiv e} \alpha_j^{(t)} (\hat{\boldsymbol{p}}^{(t)})^\top (\mathbf{W}^h)^\top \boldsymbol{V}. \end{aligned} \tag{42}$$

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