1 Question 1

We consider the following notations:

$$u_4 = w_{c^+} \odot w_t \tag{1}$$

$$u_3 = sum(-w_{c^+} \odot w_t) = sum(-u_4) \tag{2}$$

$$u_2 = e^{-w_{c^+} \cdot w_t} = e^{u_3} \tag{3}$$

$$u_1 = 1 + e^{-w_{c^+} \cdot w_t} = 1 + u_2 \tag{4}$$

$$y_1 = \log(1 + e^{-w_{c^+} \cdot w_t}) = \log(u_1) \tag{5}$$

The operation $\odot$ is is the Hadamard product, so it computes the element-wise multiplication between the vectors w_{c^+} and w_t. After summing all the elements from the resulting vector, the result will be a scalar which is equal to the dot product between w_{c^+} and w_t. We split the dot product into two operations $w_{c^+} \cdot w_t = sum(w_{c^+} \odot w_t)$ in order to make the computation of the derivatives easier to follow.
The partial derivative of the loss w.r.t one positive example can be computed as it follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}(t, C_t^+, C_t^-)}{\partial w_{c^+}} =& \frac{\partial}{\partial w_{c^+}} [\sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t})] = \\
=& \frac{\partial}{\partial w_{c^+}} \sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \frac{\partial}{\partial w_{c^+}} \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t}) = \\
=& \sum_{c \in C_t^+} \frac{\partial}{\partial w_{c^+}} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \frac{\partial}{\partial w_{c^+}} \log(1 + e^{w_c \cdot w_t}) = \\
=& \frac{\partial}{\partial w_{c^+}} \log(1 + e^{-w_{c^+} \cdot w_t}) = \frac{\partial y_1}{\partial w_{c^+}} = \\
=& \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial u_2} \cdot \frac{\partial u_2}{\partial u_3} \cdot \frac{\partial u_3}{\partial u_4} \cdot \frac{\partial u_4}{\partial w_{c^+}} = \\
=& \frac{1}{u_1} \cdot 1 \cdot e^{u_3} \cdot (\vec{-1})^T \cdot diag(w_t) = \\
=& \frac{1}{1 + e^{-w_{c^+} \cdot w_t}} \cdot e^{-w_{c^+} \cdot w_t} \cdot (-w_t^T) = \\
=& - w_t^T \cdot \frac{1}{1 + e^{w_{c^+} \cdot w_t}}
\end{aligned}$$

In order to compute the partial derivative of the loss w.r.t one negative example, we consider the following notations:

$$v_4 = w_{c^-} \odot w_t \tag{6}$$

$$v_3 = sum(w_{c^-} \odot w_t) = sum(v_4) \tag{7}$$

$$v_2 = e^{w_{c^-} \cdot w_t} = e^{v_3} \tag{8}$$

$$v_1 = 1 + e^{w_{c^-} \cdot w_t} = 1 + v_2 \tag{9}$$

$$y_2 = \log(1 + e^{w_{c^-} \cdot w_t}) = \log(v_1) \tag{10}$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}(t, C_t^+, C_t^-)}{\partial w_{c^-}} =& \frac{\partial}{\partial w_{c^-}} [\sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t})] = \\
=& \frac{\partial}{\partial w_{c^-}} \sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \frac{\partial}{\partial w_{c^-}} \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t}) = \\
=& \sum_{c \in C_t^+} \frac{\partial}{\partial w_{c^-}} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \frac{\partial}{\partial w_{c^-}} \log(1 + e^{w_c \cdot w_t}) = \\
=& \frac{\partial}{\partial w_{c^-}} \log(1 + e^{w_{c^-} \cdot w_t}) = \frac{\partial y_2}{\partial w_{c^-}} = \\
=& \frac{\partial y_2}{\partial v_1} \cdot \frac{\partial v_1}{\partial v_2} \cdot \frac{\partial v_2}{\partial v_3} \cdot \frac{\partial v_3}{\partial v_4} \cdot \frac{\partial v_4}{\partial w_{c^-}} = \\
=& \frac{1}{u_1} \cdot 1 \cdot e^{v_3} \cdot \vec{1}^T \cdot diag(w_t) = \\
=& \frac{1}{1 + e^{w_{c^-} \cdot w_t}} \cdot e^{w_{c^-} \cdot w_t} \cdot w_t^T = \\
=& w_t^T \cdot \frac{1}{1 + e^{-w_{c^-} \cdot w_t}}
\end{aligned}
$$

2 Question 2

We consider the notations from the Eq. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

$$
\begin{aligned}
\frac{\partial \mathcal{L}(t, C_t^+, C_t^-)}{\partial w_t} =& \frac{\partial}{\partial w_t} [\sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t})] = \\
=& \frac{\partial}{\partial w_t} \sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t}) + \frac{\partial}{\partial w_t} \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t}) = \\
=& \sum_{c \in C_t^+} \frac{\partial}{\partial w_t} \log(1 + e^{-w_c \cdot w_t}) + \sum_{c \in C_t^-} \frac{\partial}{\partial w_t} \log(1 + e^{w_c \cdot w_t}) = \\
=& \sum_{c \in C_t^+} \frac{\partial y_1}{\partial w_t} + \sum_{c \in C_t^-} \frac{\partial y_2}{\partial w_t} = \\
=& \sum_{c \in C_t^+} \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial u_2} \cdot \frac{\partial u_2}{\partial u_3} \cdot \frac{\partial u_3}{\partial u_4} \cdot \frac{\partial u_4}{\partial w_t} + \sum_{c \in C_t^-} \frac{\partial y_2}{\partial v_1} \cdot \frac{\partial v_1}{\partial v_2} \cdot \frac{\partial v_2}{\partial v_3} \cdot \frac{\partial v_3}{\partial v_4} \cdot \frac{\partial v_4}{\partial w_t} = \\
=& \sum_{c \in C_t^+} \frac{1}{u_1} \cdot 1 \cdot e^{u_3} \cdot (\vec{-1})^T \cdot diag(w_c) + \sum_{c \in C_t^-} \frac{1}{v_1} \cdot 1 \cdot e^{v_3} \cdot \vec{1}^T \cdot diag(w_c) = \\
=& \sum_{c \in C_t^+} -w_c^T \cdot \frac{1}{1 + e^{w_c \cdot w_t}} + \sum_{c \in C_t^-} w_c^T \cdot \frac{1}{1 + e^{-w_c \cdot w_t}}
\end{aligned}
$$

3 Question 3

The t-SNE visualization of word embeddings is presented in Fig. 1. In the figure, some embeddings which are very close are synonyms, such as 'films' and 'movies', which seems very reasonable (the cosine similarity between the embeddings of these two words - 'films', 'movies' - is equal to 0.9934, so almost equal to 1, which is the maximum value returned by the cosine similarity function). At other times, words which might seem very different are very close to one another, probably due to belonging to a same category (e.g. 'horror' and 'comedy', which are both movie genres). The cosine similarity between 'horror' and 'comedy' is equal to 0.9812, which is a high value, as expected from looking at the plot. Other embeddings which are very close together seem more difficult to interpret and less intuitive (e.g. 'love' and 'mind', cosine similarity equal to 0.976). At least some unrelated words, like 'movie' and 'banana' have a low similarity score (0.11) as would be expected. Overall, the t-SNE visualization provides interesting results, but it probably shouldn't be overinterpreted, especially since it reduces a 30D space to a 2D one. Computing word similarities might be more reliable

(though maybe not as intuitive). A possible explanation for some of the counter-intuitive examples might be the fact that the embedding space is not sufficiently high-dimensional, so it may not be sufficiently expressive. For example, in [1], the word embedding dimensionality is 400, while the plotted embeddings come from a 30D space.

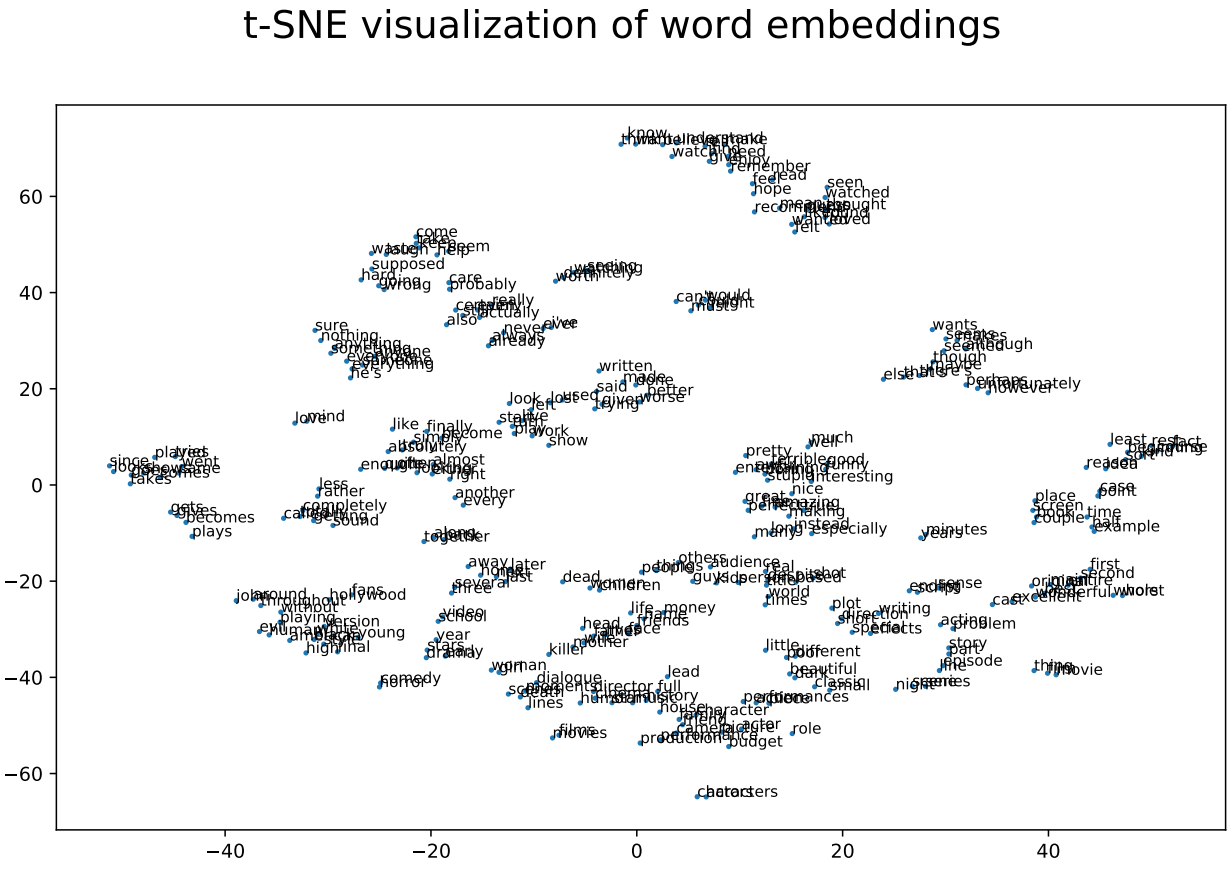

Figure 1: t-SNE visualization of word embeddings.

4 Question 4

The answer is based on the architecture described in [1] as Distributed Memory Model of Paragraph Vectors (PV-DM). We will depict the steps that can be used in order to modify the current pipeline following the guidelines from [1].

First, in order to learn separate embeddings for each review (document) we need to modify the preprocessing pipeline so that every window that will be selected will contain different words from the same review and we will also memorize the review ID associated with each window (the review from which the window is extracted).

We will need another matrix that will contain the embedding for each review, which will be initialized randomly. The simplest scenario would be to choose the same dimension for the review embeddings as for the word embeddings.

We can preserve the initial architecture with the exception that we add the review embedding as an input. Thus, the loss will change so that the dot products between the context word embeddings and the review embedding will also be taken into account. We denote the index of the review as r and the embedding of the review as d_r. The modified loss formula is presented in Eq. 11:

$$\mathcal{L}(p, t, C_t^+, C_t^-) = \sum_{c \in C_t^+} \log(1 + e^{-w_c \cdot w_t - w_c \cdot d_r}) + \sum_{c \in C_t^-} \log(1 + e^{w_c \cdot w_t + w_c \cdot d_r}) \tag{11}$$

Proceeding this way, during the training step, the review embeddings will be learned jointly with the word embeddings using Stochastic Gradient Descent (the gradient of the loss function with respect to the review

embedding will be computed using backpropagation and the review embedding will be updated using this gradient).
At test time, the word embeddings will be fixed (they will not be updated during testing). Because we encounter new reviews during testing, we need to compute their embeddings. We will use exactly the same pipeline from training, with the only exception that we will update just the review embeddings (which will be initialized randomly), while the word embeddings have fixed values (we will use the values computed during training). After computing the embeddings for all the reviews that we encounter at test time, we can use those vector representations in order to accomplish a specific task. For example, we may use the review embeddings as inputs to a logistic classifier in order to predict if the dominant sentiment of the review is negative or positive (the sentiment analysis task). We can proceed this way because these embeddings can capture the semantics of the reviews, thus they can be interpreted as features and used as inputs to different types of classifiers from machine learning.
This proposal is the simplest implementation with the smallest number of changes with regard to the current pipeline. Obviously, more of the components from [1] could be integrated, but at the cost of more changes to the code (e.g. hierarchical softmax, PV-DBOW, different dimensionality for word and review embeddings, etc.).

References

[1] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 2014.