1 Question 1

Computing the partial derivatives of the loss w.r.t one positive example w_{c^+} and one negative example w_{c^-} gives the following results:

$$\frac{\partial L}{\partial w_{c^+}} = \frac{-w_t e^{-w_t . w_{c^+}}}{1 + e^{-w_t . w_{c^+}}} = \frac{-w_t}{1 + e^{w_t . w_{c^+}}} \tag{1}$$

And :

$$\frac{\partial L}{\partial w_{c^-}} = \frac{w_t e^{w_t . w_{c^-}}}{1 + e^{w_t . w_{c^-}}} = \frac{w_t}{1 + e^{-w_t . w_{c^-}}} \tag{2}$$

2 Question 2

Computing the partial derivatives of the loss w.r.t the target word w_t gives the following result :

$$\frac{\partial L}{\partial w_t} = \sum_{c \in C_t^+} \frac{-w_c e^{-w_t . w_c}}{1 + e^{-w_t . w_c}} + \sum_{c \in C_t^-} \frac{w_c e^{w_t . w_c}}{1 + e^{w_t . w_c}} \tag{3}$$

3 Question 3

3.1 Training :

I trained my model over 15 epochs using 900k windows instead of 1M for computing time reasons.

Parameter	Value
Epochs	15
time /epoch	20min
n windows	900k
Final loss	1.93

Table 1: Relevant values for the model training

3.2 Similarity values :

I calculated the cosine similarity between various instances of words linked together semantically by different maners : singular/plural, female/male, topic ...

Others :

- similarity(”banana”,”sky”) $= 0.0492$

The plot obtained after training our model is shown in the figure below :

The embedding space is made of several small clusters, with each cluster containing words that are close to eachother and thus are supposed to be close semantically too.
Some example such as : ”father”, ”mother”, ”wife” affirm this feature of the embedding space, others seems to be less correct.

”scene”,”scenes”	0.9873
”movie”,”movies”	0.9913
”woman”,”women”	0.98
”man”,”men”	0.9794
”camera”,”cameras”	0.3719
”car”,”cars”	0.7497

Table 2: Cosine similarity for singulars and plurals

”boy”,”girl”	0.9799
”male”,”female”	0.9021
”king”,”queen”	0.6608

Table 3: Cosine similarity for gender

”movie”,”film”	0.9395
”happy”,”excited”	0.9958
”king”,”man”	0.6503

Table 4: Cosine similarity for synonyms

”good”,”bad”	0.9903
”happy”,”sad”	0.9714
”alive”,”dead”	0.8915
”funny”,”dramatic”	0.9314

Table 5: Cosine similarity for synonyms

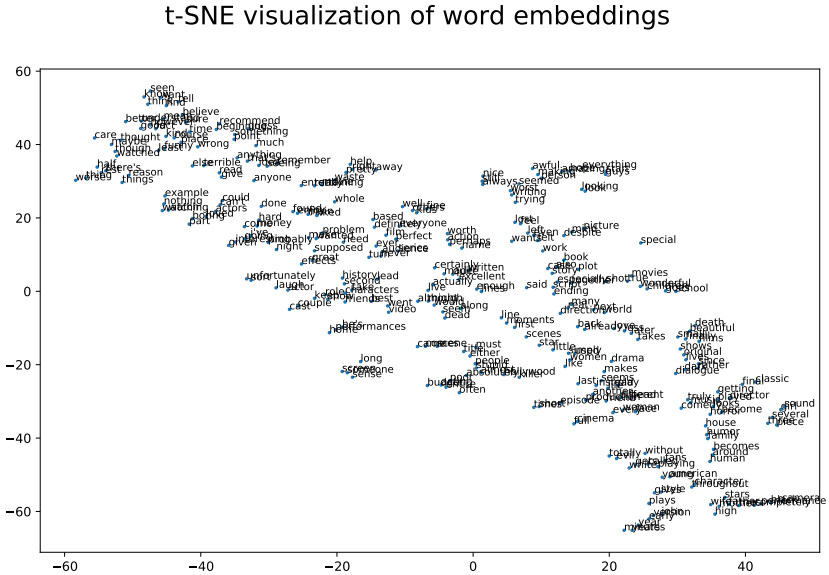

Figure 1: Word embeddings

The values of the cosine similarity show that the model is successful at determining most of the proximities between different words. Some exceptions such as ”camera”,”cameras” perform less.

4 Question 4

In word2vec using skip-gram, the context words are learnt from one target, resulting in word vectors which capture the semantical proximity between two words.

If we want to learn document vectors jointly with word vectors, we would like to have a document vector representing the probability of finding certain words in a certain document depending on the context of the document.

Similarly to the embedding space representation we obtained in Question 3, documents related to the same topic will be represented by document vectors which are close in cosine distance.

Creating a method which gives us words and documents vectors at the same time will require these following changes in the pipeline :

- **Preproccessing** : Labelling each document with a tag and creating a matrix W_d representing the docuemnt vector for each tagged document.
- **Loss function**: includes the dependance to the document vectors
- **Training** :
  - Include the document vector in the input of skip-gram for the learning of the context words, which gives :
    - Input : document vector + target
    - Action : concatenation or else
    - Output : Context wordss
  - Update the matrices W_t, W_c and W_t using stochastic gradient descent.