Assignment 1

Ahmed Lahlou Mimi
Alaeddine Bouaouaja
Mohamed Dadoune
Hassen Zarouk

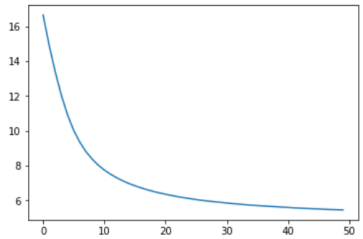
February 2019

Word2Vec is one of the most common methods in Natural Language Processing. It was initially designed to represent words in a low-dimentional space. Word2Vec method depend on several hyperparameters, some of witch are already tuned by the algorithm's designers in order to perform well in some NLP tasks such as word similarities. In this report, we will be focusing on the Skip-Gram model with negative sampling model and studying the marginal importance of each hyper-parameter used in this model.

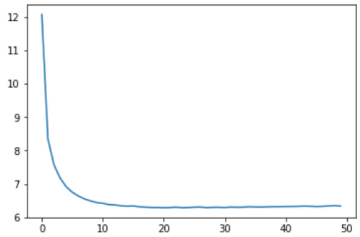
1 Hyper-parameters Tuning

We have conducted several experiments in order to test the effect of each parameter used in the Skip-Gram model on the model performance as well as on the quality of the results. We only tested different values of the hyper-parameters on a small corpus (approximately 5k articles) due to small processing power. To compare the different parameters, we look at the evolution of the loss and some qualitative results presented in the next section.

- **Embedding Size:** This is the dimension of the space where the words are represented. we can see below the difference between training a model with 100 elements and one with 300:



(a) Embedding size : 300



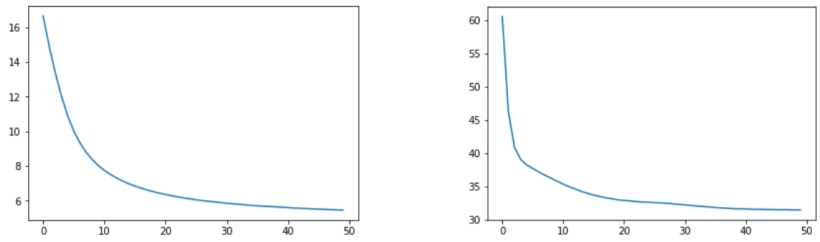
(b) Embedding size : 100

Figure 1: Embedding Size effect

in Figure 1, we can see that a larger embedding leads to a better result (more degrees of liberty) but also takes more time to converge.

- **Window Size :** We observed that if the Window Size is too large, the context information may become too blurry and lead to poor performance. In the other hand, if the Window Size is too small, it may not be sufficient for the model to learn a correct mapping between word and context.

Figure 2 show an example with a training performed on a window size of 5 and a window size of 20.



(a) Window Size : 5 (b) Window Size : 20

Figure 2: Window Size effect

- **Learning rate :** Our experiments showed that a smaller learning rate usually lead to better performance (better convergence), but is usually slower than a model with greater value. Therefore, a good trade-off had to be found between the desired speed of the algorithm and the quality of the results.

- **Negative rate :** Increasing this parameter improve the performance of the algorithm. Setting this value high allows the algorithm to have more examples to discriminate between in-context words and out-of-context words. But increasing this value also leads to longer training since the training examples increase. We observed those results by qualitatively assessing our model.

2 Qualitative Assessment of the results

In order to qualitatively evaluate our results, we used several methods. First, we evaluated the model by displaying the words most similar to a chosen word using the embeddings. To have a more global vision of the results, we have reduced the size of the embeddings (PCA) and visualized the word in a 3D space (Figure). The tests have been realized with a model trained on 100K articles.

(a) Spatial representation

(b) Most similar words to media

Figure 3

Also, in order to test the performance of the model in real situation, we combined it with several machine learning models to predict the topic of several newspaper articles and compared it with other models. Here is an example of the performances obtained:

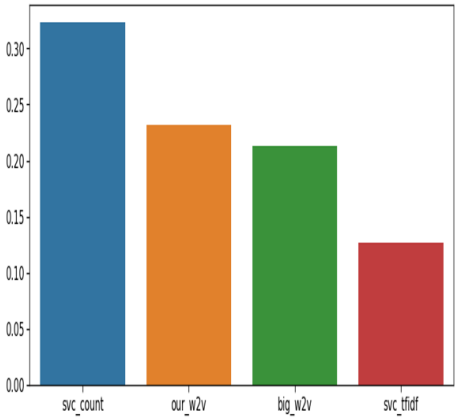


Figure 4: Performance of the model