Dhruv Verma
2017046

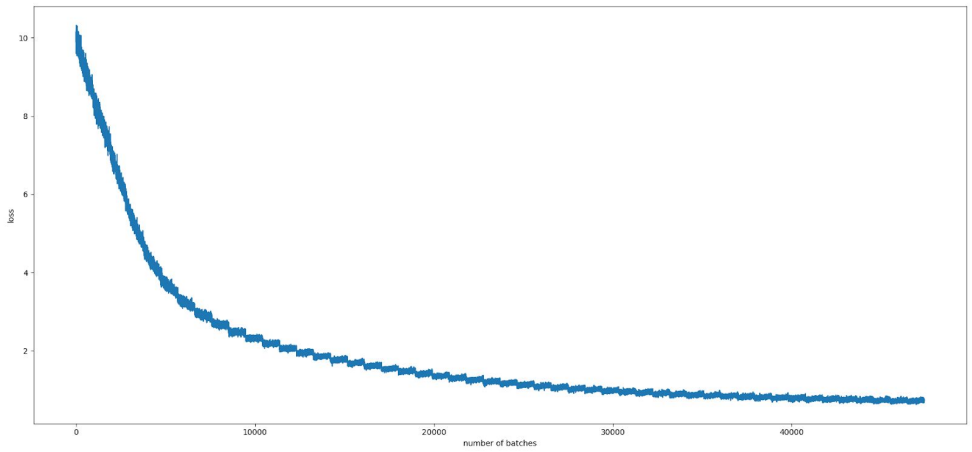
Homework 3: **Report**

Question 1: **Word2Vec**

Brief Description (Skip-Gram Model)

The objective of the skip-gram model is to predict context words given the target word. The corpus is preprocessed (removed stop-words and punctuations, **subsampling of highly frequent words**), then tokenized and indexed according to the vocabulary. In order to make the model more robust, we also incorporate some **negative samples (**chosen randomly from the specified distribution) apart from the positive ones. The model contains 2 parallel *embedding layers* (embedding size = 128) one for input and other for output. The model outputs the vector representation of the words passed as input. The **Negative Log Sigmoid** loss function is decided such that it maximizes the cosine similarity between the target and positive context samples and minimizes the cosine similarity between the target and negative context samples. The model was trained over 50 epochs with mini-batches (batch size = 1024). **Adam optimizer** was used with a learning rate of 0.001.

Loss vs Epochs/Batches



TSNE Visualization

Refer *TSNE_visualization.html*

After each epoch, with more and more training. The TSNE representation although looks similarly scattered but the clusters formed start making some linguistic sense. Words which are related appear closer to each other, whereas ones which are more unrelated are sparsely arranged.

Question 2: **Relevance Feedback in Information Retrieval**

Baseline
MAP: 0.49176786896815833

Relevance Feedback Only (alpha = **0.7,** beta = **0.3)**
After
Iteration 1: MAP: 0.5796658578711493
Iteration 2: MAP: 0.5857657707409156
Iteration 3: MAP 0.5890246386481792

Relevance Feedback with Query Expansion (alpha = **0.7,** beta = **0.3)**
After
Iteration 1: MAP: 0.5844218610282412
Iteration 2: MAP: 0.5938749256135473
Iteration 3: MAP: 0.5965043449056425

Analysis
The results show that adding relevance feedback could certainly improve the model accuracy ~10% in our case. We can also see that the increase in model performance saturates over iterations very quickly. Also, by seeing the difference in performances of with and without query expansion, we can say that Query expansion helps to a limited extent since it may add words/terms which might have high importance in relevant documents but may not be relevant to the query itself.