

Assignment 1

Deadline : 11-09-2020, 23:55 Hrs

*Instructor: Dr. Manish Shrivastava**TA: Mounika Marreddy, Prashant Kodali*

1 General Instructions

1. The assignment can be implemented in Python.
2. Ensure that the submitted assignment is your original work. Please do not copy any part from any source including your friends, seniors, and/or the internet. If any such attempt is caught then serious actions including an F grade in the course is possible.
3. A single .zip file needs to be uploaded to the Moodle Course Portal.
4. Your grade will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.

2 Problem Statement

Modern distributional semantic algorithms (also known as word embedding algorithms) can be found inside many intelligent systems dealing with natural language. They became especially popular after the introduction of prediction-based models based on artificial neural networks, like Continuous Bag-of-Words and Continuous Skip-gram algorithms, first implemented in word2vec tool. The ultimate aim is to learn meaningful vectors (embeddings) forwards in natural language, such that semantically similar words have mathematically similar vectors.

The current assignment aims to make you familiar with the above algorithms and different loss functions such as negative sampling, and hierarchical softmax. In this assignment, you need to implement the following tasks :

1. Implement the Word2Vec model and train your own word vectors using CBOW model with full softmax. [10 marks]
2. Implement the Word2vec model and train your own word vectors using Skip Gram model with negative sampling and full softmax. [10 marks]

For each of the models, report the following for the model:

- Display the top-10 word vectors for 5 different words (a combination of nouns, verbs, adjectives etc) using the above pre-trained models (1,2) using t-SNE (or such methods). [5 marks]

- What are the top 10 closest words for the word ‘camera’ in the embeddings generated by your program. Compare them against the pre-trained word2vec embeddings that you can download off the shelf (can use gensim). (1,2). [5 marks]

3 Training corpus

Please train your model on the following corpus:

http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Electronics_5.json.gz

4 Submission Format

Zip the following into one file and submit in the Moodle course portal:

1. Source Code
2. Pre-trained model and Embeddings generated
3. Report answering all the questions raised above.
4. Readme File :on how to execute the code, how to restore the pre-trained model. Any other information.

If the pre-trained models and embeddings cross the file size limits: upload them to a OneDrive/GDrive and share the links in the Readme File.

5 Grading

1. Evaluation will be individual and will be based on your viva, report, submitted code review.
2. In the slot you are expected to walk us through your code, explain your experiments, and report.