Assignment 2

Course 55807

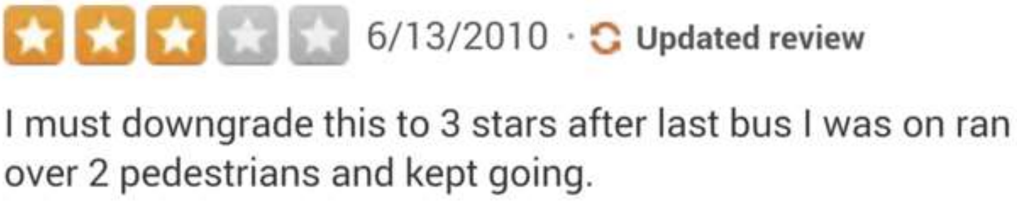
Due date: January 11, 2021. Assignment must be submitted electronically by 23:59 on the due date

Instructions

1. Assignments should be submitted electronically, as MS Word or PDF document containing text and embedded graphics. Label each plot, chart or table, add detailed title and reference it in your text (i.e. “Figure 1 shows … “).
2. The deadline is firm. Late submissions are not allowed.

Good luck

In this assignment you will use Doc2vec algorithm to perform sentiment analysis on Yelp reviews. The objective is to construct a model capable to predict review *star rating* from the text alone.



6/13/2010 · Updated review

I must downgrade this to 3 stars after last bus I was on ran over 2 pedestrians and kept going.

You can use Logistic Regression and Random Forest but are welcome to try other classifiers (e.g. Artificial Neural Network) as well.

Doc2vec is a method that embeds words in a latent space similar to Word2vec. In addition, Doc2vec also generates embeddings of the tags assigned to documents. Each document may be accompanied by a list of tags – document or paragraph id, author id or date. Some tags will appear only once in the corpus (e.g. document id) while others will be encountered multiple times (e.g. author id). Doc2vec constructs a space (docvects) and each tag is represented by a vector in that space. You will use doc2vec to generate vector representation of every comment. These vectors can be used later to train a classifier to predict the comment star rating.

Before proceeding, we recommend reading Gensim documentation for doc2vec. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#sphx-glr-auto-examples-tutorials-run-doc2vec-lee-py

Use pandas to load the Yelp reviews file and run some EDA (exploratory data analysis).

https://filesender.huji.ac.il/filesender/?s=download&token=a4c20672-7db7-4f61-a085-2207572b1712

Note that pandas can read CSV file directly from archive.

Pay attention to the stars rating distribution.

1. Normalize the text (no need to lemmatize).
2. Split the data into train & test.
3. Generate a list of gensim’s TaggedDocument objects, so that each document would contain a list of words and a list of tags. In our case, the document tag list contains a single element – the document id. Note that you need to split your data into train and test sets.
   Example of a single document:
   TaggedDocument(words=['just', 'called', 'ask', 'product', 'guy', 'phone', 'so', 'rude', 'acted', 'like', 'burden', 'i', 'phone', "there's", 'way', 'busy', '10:30', 'morning', 'never', 'going', 'location', 'learn', 'manners'], tags=[‘3’])
4. Train the Doc2vec model for at least 10 epochs with default hyperparameters (except for the vector_size which can be set to 100).
5. Now, we can use the trained model to generate vector representations for each document in the corpus. Note that the vectors for the trained set are already available at model.docvecs[‘doc_id’], or can be generated using the *model.infer_vector* function.
6. Train few classifiers using the inferred vectors and their star ratings.

7. Recall that we deal with multilabel classification, meaning you need to adjust your algorithm (e.g. multinomial logistic regression) and the hyperparameters for each algorithm (for example you should change the solver hyperparameter because it is an imbalanced dataset).
8. Which tests would you use to evaluate your models? How good are your predictions? Which model is better? For what kind of comments (star rating, number of words) does it tend to fail?

II. Improve the model:

Group comments by their score into two categories: bad (<=3 stars) and good (4 & 5 stars).

Retrain and evaluate the classifier.

III. Train and use word2vec for the same task. Compute mean of the vectors that represent the words in each document to obtain document vectors. Evaluate performance of the sentiment analysis classifier.

Submit your code and print screens of its performance evaluation in a single archive with word/pdf, python and output files (make sure to cite python/output files in the text).