University of Waterloo

ECE 657A:

Data and Knowledge Modeling and Analysis

Winter 2021

Assignment 3:

Embedding with dimensionality reduction methods and
embedding methods for natural language

Due: March 22, 2021 at 11:59pm EST

Overview

Collaboration: You may do your work individually or in pairs. You may collaborate with other students in the class on the right tools to use and setting up your programming environment but work on your own solution but be done by members of your group alone. Both members of the pair must sign up for a PairGroup in LEARN *and* in Crowdmark.

Hand in: One report per person or pair, via the CROWDMARK website in PDF, or image, format. You will need to divide the PDF up into one file for each `[CM#]` question. These files can be multiple pages. Some “questions” in this assignment have no output so no pdf if needed. It is best to start each of `[CM#]` solution on a new page and then drag and drop each onto the relevant question in Crowdmark. You should receive an invite to Crowdmark by email. Overlap and duplicated text between questions is alright, as long as the *entire answer* for each question fully contained within that question’s pdf file. Also submit the code/scripts needed to reproduce your work as a python jupyter notebook to the LEARN dropbox.

Specific objectives:

- Load datasets and perform some exploratory plots.
- Study how to apply some of the methods discussed in class and gain experience on the use of classification algorithms: PCA, LLE, t-SNE, Word2Vec
- Working on natural language embedding and processing.

Tools: You can use libraries available in python. You need to mention which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.

Dataset - HuggingFace

The HuggingFace repository hosts a large number of datasets and models for use in Natural Language Processing Research. A good example we’ve tested is

“ClimateFever”, this dataset consists of 1,535 real-world claims about climate-change collected from the internet called “ClimateFever”. It can be loaded into Python through the `datasets` library. NLTK is suggested for text preprocessing.

You can use this dataset or pick another of interest to you for the following questions. We advise picking something smal, in the 1000-3,000 words range so that you can train it effeciently
You can see that sublist here:
`https://huggingface.co/datasets?filter=size_categories:1K<Cn<3C10K`

Question 1: Representation Learning for Text Embedding

`[CM1]`:

1. Embed the text dataset with Word2Vec to convert every word of the corpus to embedding vectors. Split data into train-val-test sets with portions 80-10-10 percent. Set random state to zero for all functions of this assignment. To train the Word2Vec model, follow the instructions in the GENSIM documentation linked in the notes below. You can use the `gensim.models.word2vec.Word2Vec` function described there to train it.
2. Analyze the Word2Vec embedding space using the cosine similarity measure. Discuss if the words in similar context are actually similar and describe your analysis completely.
3. Further analyze the quality of the embeddings by trying to find 5 arithmetic computations on the embedding vectors (ie. Relationships such as “King”-“Man”+“Woman”=“Queen” that we discussed in class). The words you choose obviously will depend on the dataset you are using. Discuss the relationships you find and what they mean. Note that this dataset is small and training it from scratch may not arrive as the best possible arithmetic relationships.
4. Now load two pretrained models, one Word2Vec and and Glove model (`glove-wiki-gigaword-50` is a good one to try) on the same dataset and compare the results for arithmetic relationships on the same words to compare the scores.

Question 2: Representation Learning and dimensionality Reduction

`[CM2]`: Part 1: PCA

1. Apply PCA on the Word2Vec embeddings. Notice the train, val, and test sets in embedding with dimensionality reduction methods.

2. Use a scree plot to find the best dimensionality using the PCA subspace. In practice, people usually use either train or val sets for the scree plot but here, use the val set.

3. Visualize the first four dimensionalities of this subspace using a pairs plot. Discuss the PCA embedding applied on the Word2Vec embeddings.

4. Compare the PCA embeddings with Word2Vec embeddings. Use cosine similarity/dissimilarity for comparisons and discussions.

`[CM3]`: Part 2: LLE

1. Apply LLE on the Word2Vec embeddings. Notice the train, val, and test sets in embedding with dimensionality reduction methods. Set the embedding dimensionality to four.

2. Visualize the four dimensionalities of this subspace using a pairs plot. Discuss the LLE embedding applied on the Word2Vec embedding.

3. Compare these embeddings with Word2Vec embeddings. Use cosine similarity/dissimilarity for comparisons and discussions.

`[CM4]`: Part 3: t-SNE

1. Apply t-SNE on the Word2Vec embeddings. Notice the train, val, and test sets in embedding with dimensionality reduction methods. The correct dimensionality for this embedding should be clear from your understaning of t-SNE, explain and discuss the reason for your choice of dimensionality with your analysis.

2. Visualize the t-SNE embedding. Discuss the t-SNE embedding applied on the Word2Vec embedding.

3. Compare the embeddings with Word2Vec embeddings. Use cosine similarity/dissimilarity for comparisons and discussions.

Notes

You might find the following links are useful to solve this assignment:

- `https://radimrehurek.com/gensim/models/word2vec.html`

- `https://radimrehurek.com/gensim/auto_examples/howtos/run_downloader_api.html`

- `https://huggingface.co/datasets/climate_fever`

- `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

- https://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html
- https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html