Massive Data Storage & Retrieval Assignment: Vector Representations

Gerard de Melo

In class, we have recently been studying deep learning and representation learning. A few weeks ago, we looked the word2vec method to create vector representations of words, i.e., every word is mapped to a vector. There is another method called GloVe that relies on a different algorithmic principle but also yields vector representations of words. Download the GloVe word vectors from `http://nlp.stanford.edu/data/glove.6B.zip`. Note that this is a 822 MB download, so use the Rutgers University network if necessary.

Within the `glove.6B.300d.txt` text file inside the archive, each line contains a word followed by a space and then a series of floating point numbers (also space-separated). The floating point numbers for a word (300 in total) constitute the word's vector representation in a 300-dimensional word vector space.

Please write code to achieve the following tasks and report the results. Do not use any libraries for the nearest neighbor computation, but instead write your own code for this (calling library functions for linear algebra or data import are fine, but do not use library functions for cosine similarity or nearest neighbours). You may use any programming language (it is easy to store and manipulate a 300-dimensional array of floating point numbers in almost any programming language). Attach all code with your submission (you may submit a PDF version of a Jupyter/Zeppelin/other notebook or include your source code directly as part of a submission in ZIP format).

1 Task 1

Determine the 5 nearest neighbours of your first name in terms of the cosine similarity measure, along with the respective cosine similarity scores. For

each neighbour, list the word/name, not the vector (please provide the actual output list, not just your source code).

You may need to lower-case your name to find it (e.g. "nicole" instead of "Nicole"). If (and only if) your first name is genuinely not covered by the word vector data, then report this fact and use the first name of a celebrity instead.

2 Task 2

Write code to create a vector representation for an entire sentence simply by taking the average of all word vectors for words in that sentence. This involves 1) tokenizing a sentence, i.e., splitting it into words, for which you may use a very naïve and imperfect method. Then 2) look up the word vectors for those tokens. Make sure to apply lower-casing if necessary. You may ignore tokens that are not covered by the vocabulary of the word vectors. Finally, 3) take the average, i.e. compute the component-wise sum of the word vectors, and then divide each component by the number of words in the sentence that were covered by the data.

Next, choose a random sentence S_0 and compute the vector representation of that sentence using the above method. List the nearest neighbour *words* to that sentence vector (i.e., determine which words in the data have a similar vector representation to the vector for the sentence).

Provide the chosen sentence and the list of words in your response.

2.1 Task 3

Choose two other sentences S_1 and S_2 such that S_1 is similar in meaning to S_0, and S_2 is dissimilar in meaning to S_0. Create the sentence vectors using the method from Task 2, and report the cosine similarities between the vectors for S_0 and S_1, and between the vectors for S_0 and S_2.

Explain whether the obtained cosine similarity scores are reasonable or not. If they are reasonable, attempt to give a brief explanation of why this may have happened. If they are unreasonable, as well attempt to give a brief explanation.