NDAK15005U Information Retrieval (IR) 2019 - Assignment 1

Deadline: 13 May, 23h55. Submissions must be anonymous (no name, no KUid)

May 5, 2019

This assignment has the following three learning objectives:

1. Learn to evaluate existing distributional semantics models (known as word embeddings models) (25% of the whole assignment grade) – see Section 1.
2. Use pre-trained word embeddings models to build a text classifier (25% of the whole assignment grade) – see Section 2.
3. Extend word embeddings models to the state of the art (50% of the whole assignment grade) – see Section 3.

1 Evaluate Existing Word Embeddings Models

You are going to evaluate pretrained word embedding models. Before you begin, you have to do the following warm-up exercise.

1.1 Warm-up

1. Find an existing pretrained word embedding and explain how it was trained. A popular word embedding choice is `https://github.com/mmihaltz/word2vec-GoogleNews-vectors`, but you can decide on another embedding if you so choose.
2. When you have found a pretrained word embedding, we ask you to find the 5 nearest neighbours to the following words `King`, `London`, `Good` and `Apple`. You should find the nearest neighbours based on both Euclidean distance and cosine similarity.
3. Reduce the dimensionality of the word embeddings to 2 dimensions using Singular Value Decomposition (SVD), and choose a subset of 500 words to

plot in this low dimensional space. Describe how the words are clustered in the plot.

The python library Gensim (`https://radimrehurek.com/gensim/`) is recommended for the 3 tasks above.

1.2 Main task

You now have to evaluate the word embeddings model on analogical inferences, e.g., guessing the last element of the sequence `Paris`, `France`, `Oslo`, `X`. On `https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art)`) you will find a link to a list of 19544 word sequences. The list contains both semantic and syntactic sequences, and your task is to employ the vector offset method using the word embedding of the first 3 words in the sequence, to find the last word.

Report the accuracy for the semantic and syntactic sequences separately, as well as for the two types of sequences combined. Note that this exercise requires no training, and you should therefore evaluate on the full list. '

2 Word Embeddings for Text Classification

In this task you have to predict if a given movie review has a positive or negative sentiment using the pre-trained word embeddings from Section 1. For this you are given the IMDB sentiment dataset `https://ai.stanford.edu/~amaas/data/sentiment/`, which consists of 25000 training reviews and 25000 test reviews. Your have to train a classifier that uses the pre-trained word embeddings, but you are otherwise free to choose your classification model.

You have to describe the model and how it is trained, as well as the experimental evaluation used for testing it. When describing the model, experimental evaluation and results, address all the point that are relevant from the following guidelines: `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`.

3 Extend Word Embeddings Models to the State of the Art

You have to implement the method described in the paper "*Contextually Propagated Term Weights for Document Representation*" which can be found on Absalon under the name *CPTW.pdf*.

You have to implement the same experimental evaluation as described in the paper, but it is sufficient to only implement TF-IDF as the baseline. It is sufficient to only use the Reuters dataset (see *reuters.zip* on Absalon) for the evaluation. Describe your implementation and show how the predictive performance is affected when the threshold parameter is varied. Please present your results in the format of Table 2 and Figure 2 from the CPTW.pdf.