Description of Assignment 1

Maria Maistro
mm@di.ku.dk
May 6, 2019

Important Information

- Deadline: 13 May, 23h55;
- Submissions must be anonymous (no name, no email, no KUid);
- Submit both the code and a report, the report has to include a detailed description of the warm-up exercise and task 1, task 2, and task 3.

Learning Objectives

- OBJ1: Learn to evaluate existing word embeddings models (25%);

- OBJ2: Use pre-trained word embeddings models to build a text classifier (25%);

- OBJ3: Extend word embeddings models to the state of the art (50%).

Warm-up Exercise

- Find an existing pre-trained word embedding and explain how it was trained (https://github.com/mmihaltz/word2vec-GoogleNews-vectors);

- Find the 5 nearest neighbours (both with Euclidean distance and cosine similarity) to the following words: King, London, Good, Apple;

- 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;

- Recommendation: https://radimrehurek.com/gensim/.

OBJ1: Evaluate existing word embeddings models

- Download the Google analogy test set at this link: https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art);

- The dataset contains sequences of 4 words as for example: `Athens Greece Baghdad Iraq`;

- You need to use both semantic (e.g. `capital-common-countries`, `capital-world`, ...) and syntactic sequences (e.g. `gram1-adjective-to-adverb`, `gram2-opposite`, ...);

- With the pre-trained word embedding found in the previous exercise, employ the vector offset method using the word embedding of the first 3 words in the sequence, to predict the last word: e.g. use `Athens Greece Baghdad` to predict `Iraq`;

- Report the average accuracy of your method for: semantic sequences, syntactic sequences, semantic and syntactic sequences together.

OBJ2: Build a text classifier

- Download the IMDB sentiment dataset at this link: https://ai.stanford.edu/~amaas/data/sentiment/;
- The dataset contains reviews of movies;
- The task is sentiment classification: given a review your system has to predict whether it was positive or negative;
- Build a text classifier on top of the pre-trained word embedding found in the previous exercise;
- Train and test the classifier on the IMDB dataset;
- Describe the model implemented and the evaluation results;
- For the report you can follow the guidelines reported here: https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf.

The Machine Learning Reproducibility Checklist (Version 1.2, Mar.27 2019)

For all **models** and **algorithms** presented, check if you include:

- ❑ A clear description of the mathematical setting, algorithm, and/or model.
- ❑ An analysis of the complexity (time, space, sample size) of any algorithm.
- ❑ A link to a downloadable source code, with specification of all dependencies, including external libraries.

For any **theoretical claim**, check if you include:

- ❑ A statement of the result.
- ❑ A clear explanation of any assumptions.
- ❑ A complete proof of the claim.

For all **figures** and **tables** that present empirical results, check if you include:

- ❑ A complete description of the data collection process, including sample size.
- ❑ A link to a downloadable version of the dataset or simulation environment.
- ❑ An explanation of any data that were excluded, description of any pre-processing step.
- ❑ An explanation of how samples were allocated for training / validation / testing.
- ❑ The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.
- ❑ The exact number of evaluation runs.
- ❑ A description of how experiments were run.
- ❑ A clear definition of the specific measure or statistics used to report results.
- ❑ Clearly defined error bars.
- ❑ A description of results with central tendency (e.g. mean) & variation (e.g. stddev).
- ❑ A description of the computing infrastructure used.

OBJ3: Extend word embeddings models

- Download the Reuters dataset from Absalon;
- Implement the method described in the paper “Contextually Propagated Term Weights for Document Representation”;
- Implement TF-IDF as baseline;
- Perform the same experimental evaluation reported in the paper;
- Analyse how the predictive performance is affected when the threshold parameter is varied;
- Describe your implementation and the experimental evaluation.

Useful Links

- GloVe is an unsupervised learning algorithm for obtaining vector representations for words: https://nlp.stanford.edu/projects/glove/;

- Text Classification using CNN and RNN: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f;

- LSTM Tutorial with Keras: https://towardsdatascience.com/multi-class-text-classification-with-lstm-1590bee1bd17.

Questions?