Semantic Computing

Tutorial 6

Summer Semester 2018

Today we need some new python packages. Please *pip install* gensim. Then download Tutorial 6 from `https://github.com/dgromann/SemanticComputing`.

Exercise 1

Play with pre-trained embeddings

a) Take a look at the vector for "good" in the pre-trained embedding library *model* (see code template). Which datatype is this vector?

b) How many dimensions does this word2vec vector in this pre-trained library have?

c) Look at the 10 most similar words to "good" in the pretrained embeddings and describe their relationship to good. Is it all just synonymy?

d) What does the number in the tuple with the word mean in the result?

e) Rewrite the analogy task by getting the target vector of queen first, that is, use addition and subtraction of the vectors and then get the word for the resulting vector with the method $model.similar\_by\_vector(vector)$. Is it still queen? Is queen in the top 3 results? Why do you think there is a difference?

Exercise 2

Train your own embeddings

a) Process the already loaded "20_newsgroup" dataset in a way that it results in an array of sentences of the form [["*first*", "*sentence*], ["*second*", "*sentence*"]], where each sentence has to be split into its words which are stored in an array. The overall array then represents an array of those split sentences and that is the input to the word2vec training method.

b) Input the sentences to the method to create word2vec embeddings (see code template). What do the parameters in brackets for training with the method $Word2Vec$ mean?

c) Print the whole vocabulary you created to make sure it represents individual words. What is the length of your vocabulary?

d) Store your newly trained embeddings using the provided method. Outcomment the code for training and load the newly stored embeddings.

Exercise 3

Evaluate both embedding libraries on the analogy task
Work with the analogy.txt file provided (this is a subset of the questions-words.txt provided in the original word2vec library `https://code.google.com/archive/p/word2vec/source/default/source`).

a) Which types of relations are contained in the text file (headers are indicated by ": ")?

b) Load the analogy.txt file and use the first three words of each line in the analogy task to predict the fourth line. Check whether the correct result (word number four) is in the top 3 predictions of each embedding library. What is the accuracy on the 20_*newsgroup* embeddings? What is the accuracy with the *model* embeddings? What do you think makes the difference?

c) Evaluate the errors in predictions that are made. Can you observe a difference in the two libraries in terms of wrong predictions that are made?