NPFL097 Assignment 3
Word Clustering and Component Analysis

David Mareček

December 2024

In this assignment, you will get a vocabulary of English words with corresponding 512-dimensional vector representations. These word vectors were used to predict the next English word in a generated English sentence and were obtained from the Czech-English Neural Machine Translation model.

Your task is to analyze this highly-dimensional vector space and find the main features by which the words are distributed. The preferred programming language is Python.

In this assignment, you can search for the available functions implementing the methods, for example, the Scikit-Learn library is very useful.

Send me your final source code and commented graphs and results by email: `marecek@ufal.mff.cuni.cz`.

Data

Download the file `nmt-en-dec-512.txt` containing word vectors of the 5,000 most frequent words from English fiction collection. There is one word on each line followed by its vector representation (512 tab-separated real numbers).

Tasks

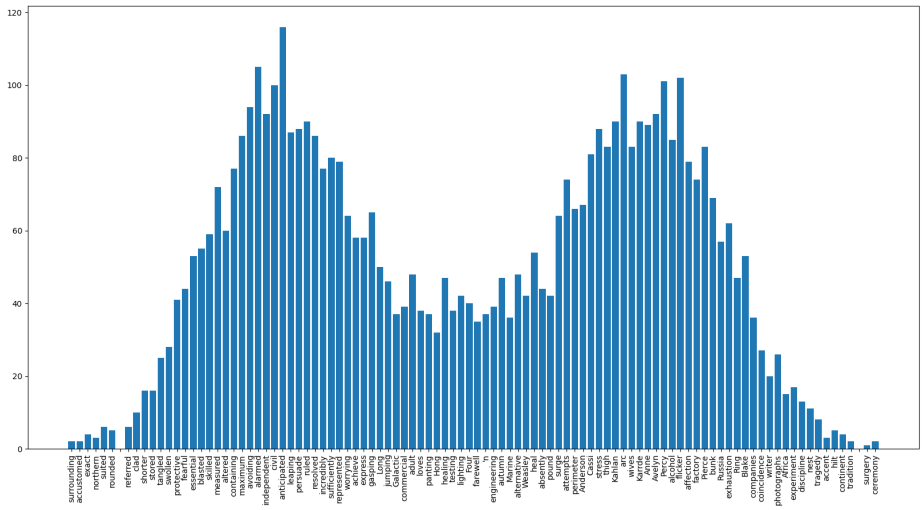
1. Visualize the word vectors in 2D using the T-SNE method. Learn how T-SNE works and what parameters it has. Draw a scatterplot with a reasonable number of labels, so that it is readable and that you can identify different clusters of words. *(1 pt)*

2. Cluster the word vectors in the original 512-dimensional vector space using the methods of K-means, GaussianMixture, and Agglomerative clustering. Try different numbers of clusters (3,5,10) and different linkage strategies (ward, single, complete). Visualize the clusters using different colors in

T-SNE. Compute the Silhouette coefficients of the individual clusterings. What method is the best? What methods failed? *(2 pts)*

3. Perform the Principal Component Analysis on the data. Visualize the data transformed into the first two principal components. Try to guess what features of words are represented by the first two components. Run the K-Means and GaussianMixture clustering on the data transformed into the first two PCA components. Compute the Silhouette coefficients and find the best number of clusters. Why the Silhouette coefficients are so different from those in Task 2? *(3 pts)*

4. Visualise the first 5 PCA components. Divide each component direction into 100 intervals (bins) and plot 5 bar charts showing the distribution of words along each component. (See the following picture.) Label the bars in your bar chart with words sampled from each bin. Try to explain what features of words are represented by individual PCA components. *(2 pts)*



5. Compute the Independent Component Analysis and obtain 50 independent components. Use '`sklearn.decomposition.FastICA`" with the parameter `n_components=50` to perform transformation into the first 50 PCA components as a pre-processing step. Visualize 5 chosen components in the same way as in Task 4. Are they non-Gaussian? What features of words do they represent? Hint: Look at the long tails of the distributions. *(2 pts)*

Extra Task

Implement the sparse auto-encoder on the given word vectors with the hidden layer of a reasonable size. Find hyperparameters for which good and inter-

pretable features are generated. Generate a file showing the list of activated words for each of the 2048 features. You can use ChatGPT to get the code you need, but be careful, there might be mistakes. *(extra 5 points, however, it is not guaranteed that a good solution exists on these data)*