Knowledge Based Systems (KBS)
Foundations of Probabilistic IR, WS 2018
Prof. Dr. Avishek Anand
Assignment 1, due: **06 Nov 2018**

You must return your assignment sheet and have a correct solution in order to present in the exercise groups. Please write legibly! Do not forget to put your name and matriculation number on your solution!

Problem 1. Let $\boldsymbol{A}$ be $m \times n$ and $\boldsymbol{B}$ be $n \times p$.
Prove:

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top$$

Problem 2.

(a) Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be $n \times n$ and invertible matrices.
Prove: $(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$

(b) Let $\boldsymbol{VDV}^{-1}$ be an eigendecomposition of a matrix $\boldsymbol{A}$, i.e. $\boldsymbol{V} = [\boldsymbol{v}^{(1)}, ..., \boldsymbol{v}^{(n)}]$ and $\boldsymbol{D} = \text{diag}([\lambda_1, ..., \lambda_n]^\top)$, where $\boldsymbol{v}^{(i)}$ are eigenvectors and λ_i are eigenvalues of $\boldsymbol{A}$.
Prove: $\boldsymbol{A}^m = \boldsymbol{VD}^m\boldsymbol{V}^{-1}$

Problem 3. Let $\boldsymbol{y} = \boldsymbol{Ax}$ where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{A}$ is $m \times n$, and $\boldsymbol{A}$ does not depend on $\boldsymbol{x}$. Suppose that $\boldsymbol{x}$ is a function of the vector $\boldsymbol{z}$, while $\boldsymbol{A}$ is independent of $\boldsymbol{z}$.
Prove:

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \boldsymbol{A}^\top$$

Problem 4. Let the scalar α be defined as $\alpha = \boldsymbol{y}^\top \boldsymbol{Ax}$ where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{A}$ is $m \times n$, and $\boldsymbol{A}$ is independent of $\boldsymbol{x}$ and $\boldsymbol{y}$.
Prove:

$$\frac{\partial \alpha}{\partial \boldsymbol{y}} = \boldsymbol{Ax}$$

Problem 5. Sentences can be represented as n-dimensional vectors by encoding the frequency of words from a vocabulary they contain, where n is the number of words in the vocabulary. For example, if we use as vocabulary the words `[the, quick, dog, horse, monkey, jumps]`, we can encode the sentence `the quick brown fox jumps over the lazy dog` using the vector

$$[\underset{\uparrow}{2}, 1, \underset{\downarrow}{1}, 0, \underset{\uparrow}{0}, 1]^\top.$$

`the` `dog` `monkey`

Write a program using **Python** and **Numpy** that

1. Reads the sentences from the provided dataset `shakespeare_sentences.txt` (every line contains one sentence)
2. Determines the k overall most frequent words
3. Encodes sentences as vectors using the frequent words as the vocabulary
4. Finds the l sentences that are most similar to a query sentence using *cosine similarity*

The cosine similarity of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as

$$sim(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2}.$$

Your program should accept the following arguments:

- A query sentence
- A value for k
- A value for l

Punctuation marks should be treated the same way as words. You can use the provided template (`cosine_sim_template.py`).