Universität Stuttgart

Institut für Visualisierung
und Interaktive Systeme

Assignment 10

Information Visualization & Visual Analytics (WS 2019/20)

Due: Monday, 20.01.2020, 11:59 AM **Discussion:** Wednesday, 22.01.2020

Point of contact: Johannes Knittel `<johannes.knittel@vis.uni-stuttgart.de>`

Please solve the assignment in **groups of up to three (3) students**. Choose one student who uploads your solution on the assignments page in ILIAS as PDF (for theoretical submissions) or ZIP (for practical submissions $\boxed{\text{Impl}}$). The submitted files should follow the naming scheme `yourlastname1_yourlastname2_yourlastname3` with respective file-ending, of course. Make sure that you create your team before uploading the solution.

Please post general questions regarding the exercise, but *no solutions,* to the forum. In case of more specific problems, especially such that cannot be posed outside of the context of your own solution, please send an email to, or make an appointment with, the tutor responsible for the exercise.

Task 1 Document Similarity *[Points: 10]*

In visual text analytics it is often useful to estimate the similarity between text documents, e.g., for clustering document collections. One common approach is to transform each document to a vector-based representation and calculate the cosine similarity[1] between pairs of vectors. In this task you should calculate the pairwise similarities of the following sentences:

1. `impeachment - pelosi prepares to send articles to senate`
2. `head of hr watch denied entry to hong kong`
3. `senate awaits articles of impeachment`

(a) *(3 points)* Convert each sentence into the so-called bag-of-words[2] representation where each item of the vector represents exactly one word of the vocabulary and contains the number of occurrences of that particular word. *Hint:* You can leave out zero entries using the `index:value` notation.

(b) *(3 points)* Calculate the cosine similarity between each pair of vectors. If you compare the resulting similarities of the first to the other two sentences, what do you observe?

(c) *(4 points)* Imagine you want to calculate the similarity between millions of tweets. Describe two shortcomings of the bag-of-words model.

Task 2 $\boxed{\text{Impl}}$ Keyword Extraction *[Points: 10]*

Extracting representative keywords from documents is a popular way to quickly get an overview of large document collections. TF-IDF[3] is a basic but commonly used method to do this. The idea is to extract terms that appear often in the given document (high *term frequency*), but rarely in the complete corpus (low *document frequency*), which is expressed in the following formula:

$$\operatorname{tfidf}(t) = f_t \log\left(\frac{N}{n_t}\right) \tag{1}$$

The tfidf score of a term t is its frequency f_t in the given document multiplied with the logarithm of the inverse document frequency which is the number of documents N divided by the number of

[1] https://en.wikipedia.org/wiki/Cosine_similarity

[2] https://en.wikipedia.org/wiki/Bag-of-words_model

[3] https://en.wikipedia.org/wiki/Tf-idf

documents the term t appears in (n_t). Given a news article[4], your task is to implement this weighting scheme to extract the top 20 keywords with the highest tfidf score. You will find the project skeleton on ILIAS.

The data structure of this assignment is as follows:

DataProvider

getWords(): ArrayList<String> – returns the article as a list of its words.
getNumberOfDocuments(): int – returns number of documents N.
getDocumentFrequency(word): int – returns document frequency n_t of provided word.

Please note that you don't have to crawl the article from the site, the `getWords()` method already returns the article as a list of its words.

(a) *(4 points)* Calculate the **term frequency** for each distinct word of the article and print them to the Console. Look at the top words with the highest frequency. What do you observe?

(b) *(6 points)* Now calculate the **tfidf score** for each distinct word of the article. Sort the words in descending order of the score and draw the top 20 terms onto the canvas. Judge the quality of the extracted terms. Do you think you get better results compared to (a)?

(c) *(up to +4 points)* **Bonus (optional)**
Draw each word so that its font size is proportional to the square root of its tfidf score. Try to optimize space usage, for instance, using columns and rows.

[4] https://www.theguardian.com/uk-news/2020/jan/08/prince-harry-and-meghan-say-they-are-stepping-back-from-royal-family