Introduction to Web Science

Assignment 6

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: December 6, 2016, 10:00 a.m.
Tutorial on: December 9, 2016, 12:00 p.m.

Please look at the lessons 1) **Simple descriptive text models** & 2) **Advanced descriptive text models**

For all the assignment questions that require you to write code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Team Oscar: Abdullah Elkindy, Stela Nebiaj, Fiorela Ciroku

1 Digging deeper into Norms (10 points)

You have been introduced to the concept of a norm and have seen that the uniform norm $||\cdot||_\infty$ fullfills all three axioms of a norm which are:

1. Positive definite
2. Homogeneous
3. Triangle inequality

Recall that for a function $f : M \longrightarrow \mathbb{R}$ with M being a finite set[1] we have defined the L_1-norm of f as:

$$||f||_1 := \sum_{x \in M} |f(x)| \tag{1}$$

In this exercise you should

1. calculate $||f-g||_1$ and $||f-g||_\infty$ for the functions f and g that are defined as
   - $f(0) = 2, f(1) = -4, f(2) = 8, f(3) = -4$ and
   - $g(0) = 5, f(1) = 1, g(2) = 7, g(3) = -3$
2. proof that all three axioms for norms hold for the L_1-norm.

Calculations : L-1 Norm

$$||f-g||_1 = \sum_{x \in M} |f(x) - g(x)| \tag{2}$$

$$= |f(0) - g(0)| + |f(1) - g(1)| + |f(2) - g(2)| + |f(3) - g(3)| \tag{3}$$

$$= |2-5| + |-4-1| + |8-7| + |-4-(-3)| = 10 \tag{4}$$

Uniform Norm

$$||f-g||_\infty = sup_{x \in M}||f(x) - g(x)|| \tag{5}$$

$$= sup(|f(0) - g(0)|, |f(1) - g(1)|, |f(2) - g(2)|, |f(3) - g(3)|) \tag{6}$$

$$= sup(|2-5|, |-4-1|, |8-7|, |-4-(-3)|) \tag{7}$$

$$= sup(|-3|, |-5|, |1|, |-1|) := 5 \tag{8}$$

- Positive definite

$$||f||_1 = 0 \longrightarrow \ f = 0 \tag{9}$$

Proof :

$$||f||_1 = 0 \longrightarrow \sum_{x \in M} |f(x)| = 0 \tag{10}$$

[1] You could for example think of the function measuring the frequency of a word depening on its rank.

$$\longrightarrow \; ||f(x)|| = 0 \forall x \tag{11}$$

$$\longrightarrow f(x) = 0 \forall x \longrightarrow f = 0 \tag{12}$$

- Homogeneous:

$$||\alpha f||_1 = \alpha ||f||_1, \alpha \in R \tag{13}$$

Proof:

$$||\alpha f||_1 = \sum_{x \in M} |\alpha f(x)| = \sum_{x \in M} |\alpha||f(x)| = \alpha \sum_{x \in M} |f(x)| = \alpha * ||f||_1 \tag{14}$$

- Triangle inequality

$$||f + g||_1 \leq ||f||_1 + ||g||_1 \tag{15}$$

Proof:

$$||f + g||_1 = \sum_{x \in M} |f(x) + g(x)| \tag{16}$$

$$\leq \sum_{x \in M} |f(x)| + |g(x)| \tag{17}$$

$$\leq \sum_{x \in M} |f(x)| + \sum_{x \in M} |g(x)| = ||f||_1 + ||g||_1 \tag{18}$$

2 Coming up with a research hypothesis (12 points)

You can find all the text of the articles from Simple English Wikipedia at `http://141.26.208.82/simple-20160801-1-article-per-line.zip` each line contains one single article.

In this task we want you to be creative and do some research on this data set. The ultimate goal for this exercise is to practice the way of coming up with a research hypothesis and testable predictions.

In order to do this please **shortly**[2] answer the following questions:

1. What are some observations about the data set that you can make? State at least three observations.

   - A lot of words in the data set are small words.
   - A lot of articles contain places (countries, cities).
   - There are more short articles than long ones.
   - The longer an article is, the more likely to contain smaller words.

2. Which of these observations make you curious and awaken your interest? Ask a question about why this pattern could occur.

   We are more interested in the forth observation: ”The longer an article is, the more likely to contain smaller words.”. We think that this occurs because the longer the article is, the more functional words it contains. Most of the functional words are small words, up to 3 or 4 characters. Therefor this decreases the average of the length of all the words in the article.

3. Formulate up to three potential research hypothesis.

   - More than 80 percent of the frequent words in SimpleEnWiki are small words. Where in definition, a small word is any word that consists of 4 or less characters.
   - 60 percent of the articles in SimpleEnWiki contain places.
   - Less than 25 percent of the articles are long articles. Where, in definition, a long article is an article that contains more than x words.
   - With the increase of the number of words in an article, the article's Word-Length-Average decreases. Where in definition, an article's Word-Length-Average is the average of length of its words.

4. Take the most promising hypothesis and develop testable predictions.

[2] Depending on the question shortly could mean one or two sentences or up to a thousand characters. We don't want to give a harsh limit because we trust in you to be reasonable.

In order to test our hypothesis we are going to count the number of words per article and also the total number of letters per articles. After this we divide the total number of letters per article with the number of words per article in order to find the average number of letters per word in an article. We will store the number of words per article and the average of letters per word in an article in two different arrays and plot them in a graph so that we have a more clear picture of the outcome.

5. Explain how you would like to use the data set to test the prediction by means of descriptive statistics. Also explain how you would expect your outcome.

   After storing the number of words per article and the average of letters per word in an article, we can plot them. In the x-axis of the diagram is the number of words per article and in the y-axis is the average of letters per word in a article. After plotting we expect to see a decreasing tendency of the average number of letters per word with the increase of the number of words in an article.

   (If you realize that the last two steps would not lead anywhere repeat with one of your other research hypothesis.)

2.1 Hints:

- The first question could already include some diagrams (from the lecture or ones that you did yourselves).
- In step 3 explain how each of your hypothesis is falsifiable.
- In the fifth step you could state something like: ”We expect to see two diagrams. The first one has ... on the x-axis and ... on the y-axis. The image should look like a ... The second diagram ...”. You could even draw a sketch of the diagram and explain how this would support or reject your testable hypothesis.

3 Statistical Validity (8 points)

In the above question, you were asked to formulate your hypothesis. In this one, you should follow your own defined roadmap from task 2 validate (or reject) your hypothesis.

3.1 Hints:

- In case feel uncomfortable to test one of the predictions from task 2 you can ”steal” one of the many hypothesis (and with them implicitly associated testable predictions) or diagrams depicted from the lecture and reproduce it. However in that case you cannot expect to get the total amount of points for task 3.

```
import re
import matplotlib.pyplot as plt
import statistics
words_per_article_high=list()
words_per_article_low=list()
average_high=list()
average_low=list()
with open('simple-20160801-1-article-per-line','rb') as file:
    content=file.readlines()
index=0
while index<len(content):
    line=content[index].decode()
    count =0
    wordlist= re.findall(r"\b([a-zA-Z]+-*[a-zA-Z]*)\b",line, re.I)
    av=0
    for w in wordlist:
        count = count +len(w)
    if(len(wordlist))==0 :
        av=0
    else:
        av=count/float(len(wordlist))
#value we precalculated to the median of the averages so that we seperated into a
    if(av>=4.68075117370892):
        average_high.append(av)
        words_per_article_high.append(len(wordlist))
    else:
        average_low.append(av)
        words_per_article_low.append(len(wordlist))
    index=index+1
file.close()
perc=len(words_per_article_high)/(len(words_per_article_high)+len(words_per_artic
perc=int(perc*100)
print("The hypothesis holds for around "+str(perc)+"% of the data")
#print(statistics.median(average))
plt.title("Exercise 3")
plt.xlabel('words per document')
plt.ylabel('average')
plt.scatter(words_per_article_high, average_high)
plt.scatter(words_per_article_low, average_low,c=[2000]*len(average_low))
plt.ylim(0,max(average_high))
plt.xlim(0,max(words_per_article_high))
plt.show()
```

Our hypothesis is: ”With the increase of the number of words in an article, the article’s Word-Length-Average decreases. Where in definition, an article’s Word-Length- Average is the average of length of its words.”

The above code gives out the information given in Figure 1, which denies our hypothesis. As it seems from the plot that the hypothesis applies to half of the data, the light blue ones. Where we can easily see a gradual decrease in the average number of words per doc. However, the other half of the data are the opposite. We thought through our observation and hypothesis and given the fact that the bigger an article is the more frequent words in has (small size words) thus this will lead to what we hypothesised but through statistical analysis it was denied.

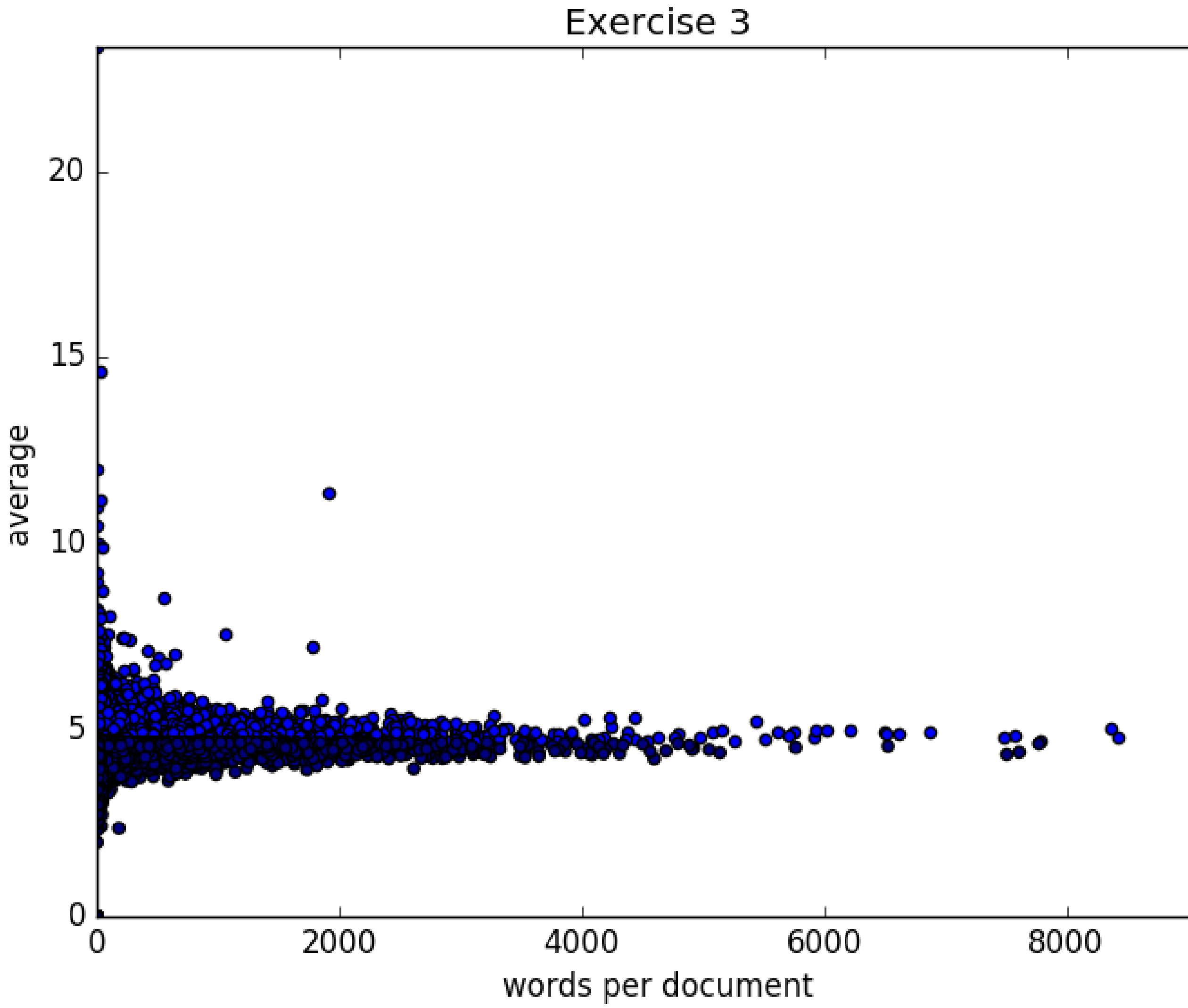
Exercise 3
average
words per document
0
5
10
15
20
0
2000
4000
6000
8000

Figure 1

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment6/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of ”Web Information Retrieval”.

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.