# 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 Name: Tango

1. Mariya Chkalova
   mchkalova@uni-koblenz.de
2. Arsenii Smyrnov
   smyrnov@uni-koblenz.de
3. Simon Schauß
   sschauss@uni-koblenz.de
4. Lukas Härtel
   lukashaertel@uni-koblenz.de

# 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. Positiv 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.

## 1.1 Hints:

1. The proofs work in a very similar fashion to those from the uniform norm that was depicted in the videos.
2. You can expect that the proofs for each property also will be "three-liners".
3. Both parts of this exercise are meant to practice proper and clean mathematical notation as this is very helpfull when reading and understanding research papers. Discuss in your study group not only the logics of the calculation and the proof (before submission) but try to emphasize on the question whether your submission is able to communicate exactly what you are doing.

1.

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

$$\begin{aligned}||f-g||_1 :&= \sum_{x\in 1,2,3,4} |f(x)-g(x)| \\ &= |f(0)-g(0)|+|f(1)-g(1)|+|f(2)-g(2)|+|f(3)-g(3)| \\ &= |2-5|+|-4-1|+|8-7|+|-4+3| \\ &= 3+5+1+1 \\ &= 10\end{aligned}$$

$$\begin{aligned}||f-g||_\infty :&= sup\,|f(x)-g(x)| \\ &= |f(1)-g(1)| \\ &= |-4-1| \\ &= 5\end{aligned}$$

2. The proof:

- positive definite

If $||f-g||_1 = 0$, then $\sum_{x\in M}|f(x)-g(x)|$=0,
|f(x) - g(x)| = 0 for any x.
That means that f(x) - g(x) = 0.
So f-g = 0.

- homogeneous (a is a real number)

$$\begin{aligned}||a(f-g)||_1 :&= \sum_{x\in M} |a(f(x)-g(x)| \\ &= \sum_{x\in M} |a||f(x)-g(x)| \\ &= |a|\sum_{x\in M} |f(x)-g(x)| \\ &= |a|||(f-g||_1\end{aligned}$$

- triangle inequality

$$\begin{aligned}||(f-g)+h||_1 &:= \sum_{x\in M} |(f(x)-g(x))+h(x)| \\ &<= \sum_{x\in M} |(f(x)-g(x)|+|h(x)| \\ &<= \sum_{x\in M} |(f(x)-g(x)|+\sum_{x\in M}|h(x)| \\ &<= ||f-g||+||h||\end{aligned}$$

## 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 obervations about the data set that you can make? State at least three obervations.
2. Which of these observations make you curious and awaken your interest? Ask a question about why this pattern could occur.
3. Formulate up to three potentiel research hypothesis.
4. Take the most promesing hypothesis and develop testable predictions.
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.

   (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.

1. Observations:
   a) Some articles are blank - 1854 empty lines were found on the dataset of articles.
   b) Articles have different length.
   c) Articles normally start with a sentence containing ”is”, ”are”, ”was”, or ”were”.

[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.

d) There are fewer articles that start in the past (”was”, ”were”) than in present (”is”, ”are”)

e) Articles that start in the past (”was”, ”were”) are generally longer than those that start in present (”is”, ”are”).

2. Interesting observations: observation e) from the above makes us to ask the question what could be the reason of such phenomena? Could it be connected with the assumption that articles, started with ’was’, ’were’ deal with hystorical materials and this could influence on the length? What is to calculate the exact length of articles and compare?

3. Hypothesis:

a) The length of the article increases the complexity of understanding the article. (This hypothesis is falsifiable as it deals with a compute exercise and the result of counting the length of the articles and ARI indexes could reject this hypothesis).

b) Guidelines for article writing may require a straightforward explanation of what the article is about.

c) Articles about concepts, notions and ideas outnumber historic articles.

d) The concluded nature of historic articles allows for greater and more comprehensive content, thus, longer articles. (Hypothesis is falsifiable as it is possible to compute the exact length and approve or reject this assumption).

4. Testable predictions:

- Hypothesis a: The complexity of understanding the article could be measured by the ARI (Automated Readability Index). The length of the article will be defined by the number of characters in the text. Characters are the number of letters and numbers. Blank articles that were observed in the data set will not be analyzed. In order to test the prediction it is necessary to calculate the number of characters and the ARI for each article. Then we should observe that longer articles should have a higher ARI - the linear dependency of the article length and ARI.

- Hypothesis d: If the first sentence in an article contains ”was” or ”were”, the article is assumed to be longer than the general dataset population.

5. Dataset usage

- Hypothesis a: Let assume that after step 4 the dataframe is prepared with the columns NumCharacters, ARI. Then the plot could be build to visualize the dependency. X axis will have the number of characters in an article (on log scale), Y axis - calculated ARI. We expect to the see that the number of characters and the ARI are proportionally increasing.

- Hypothesis d: All articles with ”was” or ”were” are put into the ”past” bucket. Mean values are calculated all articles and for those marked as ”past”. The mean value of ”past”, ”present” and ”all” articles are calculated and compared.

# 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.

- Hypothesis a: The data is calculated (length of characters in the article, ARI index) and plotted. However as it could be considered ARI index does not depend on the length, so the hypothesis fails.

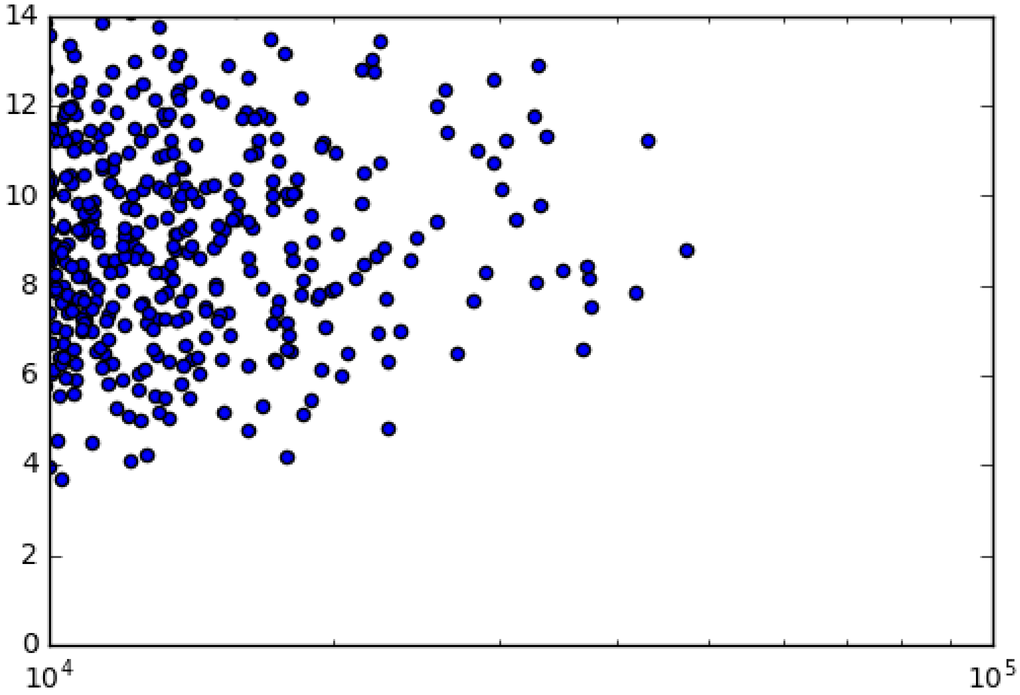

**Figure 1:** Depending of the ARI index vs article length

- Hypothesis d: The data is processed and plotted as a histogram, accompanied by visible locations of mean values. Fig. 1 shows the result. The mean value of ”past” articles is significantly greater than those of ”present” and ”all” articles.

## 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 imlicitly 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.

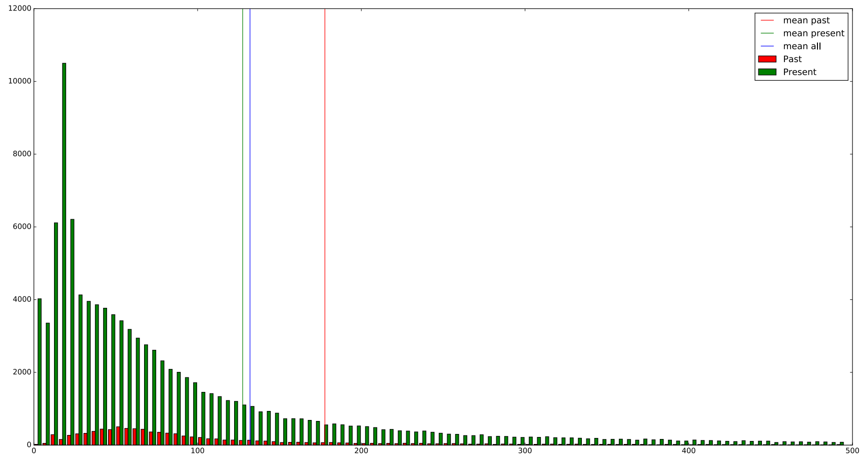

**Figure 2:** Count on the Y-axis, length of articles (in number of sentences) on the X-axis. Values, means and standard deviations for ”all” in blue, ”present” in green, and ”past” in red. X-axis is cut off after 500.

**Listing 1** Ari index, Article length calculation

```
# coding: utf-8

# In[29]:

file=r"simple-20160801-1-article-per-line"

def readWordsFromWiki (filename):
    f=open(filename,'r',encoding='utf-8')
    allWords=[]
    for line in f:
        line=line[:-1]
        words=line.split()
        allWords.extend(words)
    f.close()
    return len(allWords)

words=readWordsFromWiki(file)


# In[31]:

import re

# Get rid of empty lines
def get_lines_notEmpty (filename):
    f = open(filename,'r',encoding='utf-8')
    contents = f.readlines()
    f.close()

    file_content = []
    for line in contents:
        # Strip whitespace, should leave nothing if empty line was just "\n"
        if re.match(r'^\s*$', line):
            continue
        # We got something, save it
        else:
            file_content.append(line)
    return file_content

def get_lines (filename):
    f = open(filename,'r',encoding='utf-8')
    contents = f.readlines()
    f.close()
    return contents

Lines = get_lines(file)
Lines_notEmpty = get_lines_notEmpty(file)

count_lines_notEmpty = len(Lines_notEmpty)
count_lines = len(Lines)


# In[32]:

print("Empty lines: ", count_lines - count_lines_notEmpty)
```

**Listing 2** Program for hystogram

```
# coding: utf-8

# In[ ]:

from collections import Counter
import numpy as np
import matplotlib.pyplot as plt


def fst(item):
    """Function to get the first element."""
    return item[0]


def snd(item):
    """Function to get the second element."""
    return item[1]


def prepare_hist(counter):
    """Prepares the histogram by unrolling a counter into domain and the mapped value.""
    ordered = sorted(counter.items(), key=fst)
    return np.fromiter(map(fst, ordered), float), np.fromiter(map(snd, ordered), float)


def flatten_counter(counter):
    """Flattens the counter, repeats in a list the elements as often as mentioned in the
    Counter(xs) == Counter(flatten_counter(Counter(xs))) should hold.
    """
    return [k for k, v in counter.items() for _ in range(0, v)]


if __name__ == '__main__':
    print("The file 'simple-20160801-1-article-per-line' should exist in the working dire

    # Set up counters for sentences in the appropriate categories
    past = Counter()
    present = Counter()

    # Count in present and past
    with open("simple-20160801-1-article-per-line", "r") as file:
        for line in file:
            article = line.split('.', maxsplit=1)[0].lower()
            x = max(article.find("was"), article.find("were"))
            y = max(article.find("is"), article.find("are"))

            if x >= 0 and not y >= 0:
                past[line.count(" ")] += 1
            else:
                present[line.count(" ")] += 1

    # Flatten for numpy calculations
    flat_past = flatten_counter(past)
    flat_present = flatten_counter(present)
    flat_all = flat_past + flat_present
```

# 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`.