Assignment 1

Text as Data

2023-09-18

Introduction

This first assignment will test your grasp of some of the concepts and methods we covered in sessions 1-2. Give an answer to each question, even if you are not sure. In an ideal world, you should present your work as a document with integrated code, execution and text, such as an Rmd file, or an .ipynb notebook, but if this doesn't work out a script and a pdf will do fine.

1

Consider the two following texts.

```
texts <- c(
  "I don't like cricket",
  "You like cricket"
)
```

1.1 Write two text processing pipelines that create document feature matrices that **do** and **do not** preserve the differences in the texts.

Here he is asking to write a bigram vs. a unigram feature

1.2. Describe how the preprocessing choices you make affect the representation of those texts.

1.3. Give one example each of **tasks** where your text preprocessing choices that do not preserve the differences in the texts **would** and **would not** limit our ability to perform the task.

2

would limit: victim vs. preditor detection, say in femicide analysis of newspapers. The problem is we don't know anymore who did what to whom and thereby who the preditor and who the victim is.
would not limit: topic classification - in most cases it is enough to detect keywords in order to classify documents into topics.

Now consider the following three texts.

```
texts <- c(
  "Climatic change is causing adverse impacts",
  "Changes in the climate have caused impacts to human systems",
  "Chelsea have a goal difference of zero in the premier league this season"
)
```

2.1. Turn these texts into a document feature matrix without any additional pre-processing steps

2.2. Calculate a simplistic measure of how similar each text is to each other, by reporting the number of columns where both texts contain a non-zero value (you can do this with code or by hand).

2.3. Create a text processing pipeline that results in a matrix that preserves the similarity we can see intuitively between the texts.

use TFIDF and take out stop words and lemmatise/stemming also works here

2.4. Comment on why the additional pre-processing steps created a more useful representation of the texts *in this case*.

There were stop word matches that were overly impacting the similarity measures (taken care of by excluding stop wrods and using TFIDF)
There were words which very unique for the third sentence which was very indicative for a big difference in terms of contextual similarity of the two texts such as Chelsea, goal and premier league - this could be identified using TFIDF
There were similar words between the first two documents which however did not flag as similar given they were different tenses or conjugation. By lemmatising these words got flagged as the "same" which increased the similarity between the first and second document.