

Lab 4 TDT4310

Vebjørn Ohr

April 09 2021

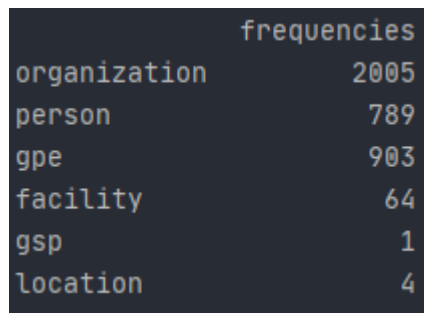
Exercise 1

a)

The (named) entities were found using the NLTK `ne_chunk_sents` function after tokenizing and part-of-speech tagging the sentences also using NLTK methods (didn't realise i could use spaCy).

b)

The frequencies were recorded by using the `ConditionalFreqDist` class of NLTK while iterating through the chunks (tree-structure) from task a. The frequency for each entity class can be seen in Figure 1.



	frequencies
organization	2005
person	789
gpe	903
facility	64
gsp	1
location	4

Figure 1: Task 1 named entities

c)

The top 10 persons was plotted for readability using Matplotlib. The names were split into single words so that for example 'Han' and 'Han Solo' both would be recorded under 'Han', as there is no named entity linking . The bar chart is shown in Figure 2.

d)

I think the answer to if the graph represents the importance of each character depends on how you define importance in the context of the movie. In terms of screen time the graph is not representative. The graph shows that Han has a higher frequency than Luke even though Luke has more screen time than Han. Still they are the two character with the most screen time and

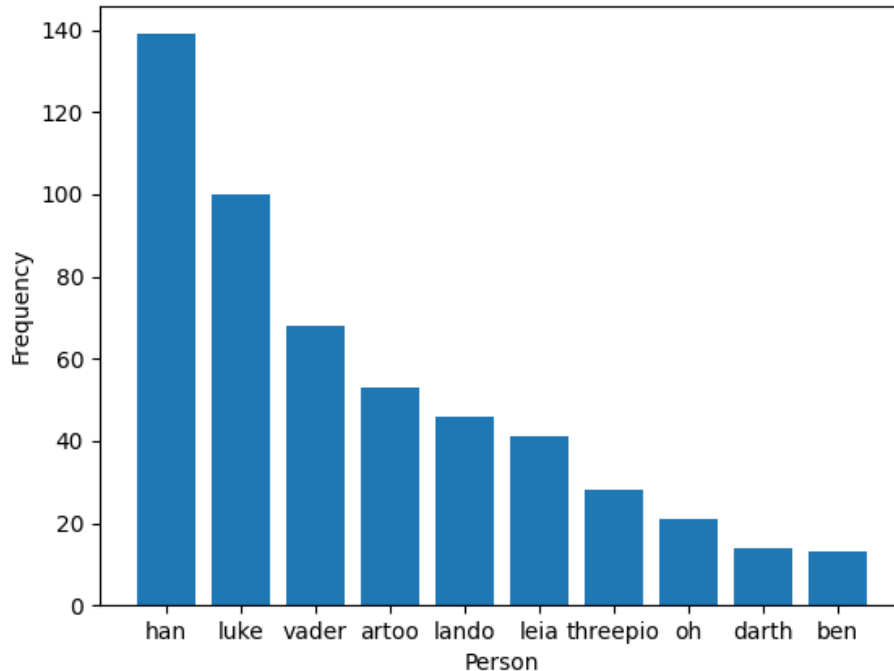


Figure 2: Task 1 person frequencies

the highest frequency¹. You could also look at the importance of characters in regards to the importance of the story. For example Vader gets the third highest frequency even though he doesn't have much screen time. Still you could argue that he has an important role in the Star Wars story. Still Leia seems to be underrepresented, and Chewbacca is not in the list (wrong entity class). A better way would perhaps be to look at how many scenes the characters appear in.

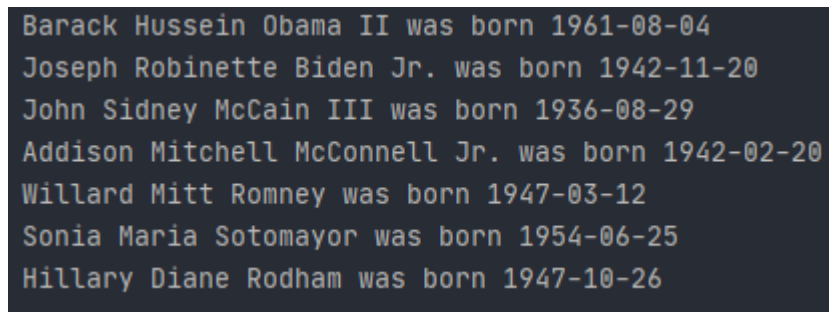
Exercise 2

The Python wikipedia package was used to get the data about Barack Obama (summary). The requests library of Python was used to send requests to the Spotlight API as well as for getting DBPedia resources². Using the type pa-

¹<https://screenrant.com/star-wars-episode-v-empire-strikes-back-characters-screen-time/>

²Using the API available at <http://vmdbpedia.informatik.uni-leipzig.de:8080/api/1.0.0/values>

parameter "DBpedia:Person" in the request to the Spotlight API returned all the persons found in the text. The end of URIs for these resources, e.g. Hillary_Clinton for the `http://dbpedia.org/resource/Hillary_Clinton` URI were used as the 'entities' parameter when getting more information from the DBPedia API. The properties chosen were "dbo:birthDate", and "dbo:birthName", returning the birth date and birth name for all the persons. The results can be seen in Figure 3.



```
Barack Hussein Obama II was born 1961-08-04
Joseph Robinette Biden Jr. was born 1942-11-20
John Sidney McCain III was born 1936-08-29
Addison Mitchell McConnell Jr. was born 1942-02-20
Willard Mitt Romney was born 1947-03-12
Sonia Maria Sotomayor was born 1954-06-25
Hillary Diane Rodham was born 1947-10-26
```

Figure 3: Task 2 b) Birth name and birth date extracted for the persons mentioned in the Barack Obama Wikipedia article (summary).

Exercise 3

a)

TextBlob³ was used for sentiment analysis. One thing to note is that TextBlob is not a 100% accurate, meaning that the accuracy of the final model might be higher or lower than the actual calculated accuracy value.

b)

The Sequential Model of Keras was used with 3 layers. The first layer is the word embedding layer used to represent the words as vectors. The second layer is the LSTM layer with dropout to reduce overfitting. The 3rd is the output layer using softmax as its activation function. Early stopping was used on the accuracy measure to avoid overfitting as well, and the most accurate model was saved for each epoch. Generally the model used around 11 epochs before stopping.

³<https://textblob.readthedocs.io/en/dev/>

c)

The final model got around 92% accuracy on the validation data, and 78% on Trump's tweets (with TextBlob sentiment analysis). This may be because the mode is trained on a variety of more general tweets, and Trump's tweet language may not fit this model that well.

d)

To make the data readable i decided to group the tweets into a set of tweets for each day in the two month period using *pandas*. Then the number of positive and the number of negative tweets were counted for each date. The results can be seen in Figure 4. The green line show the number of positive tweets for each chunk, and the blue line is the same for negative tweets. The horizontal lines are the average for each over the period. The graph does seem to indicate a sinking trend of positive tweets towards the end of the 60 day period. The spikes of positive tweets also reduces drastically after the first 30 day period, which would be the election day. The negative tweets seems stay more consistent with slight reduction. After the election day both the number of positive and negative tweets are more below their average lines. Calculating the average of positive tweets and for negative tweets for both the first and the last 30 days reveals a decrease in the average of 7.9 for positive and 3.1 for negative, indicating that the number of tweets went down overall, but mostly the positive labeled ones. Neutral labeled tweets are of less interest and are not included because many of them stems from tweets being empty after preprocessing (e.g a tweet containing only a hyperlink) and was not accounted for in this task.

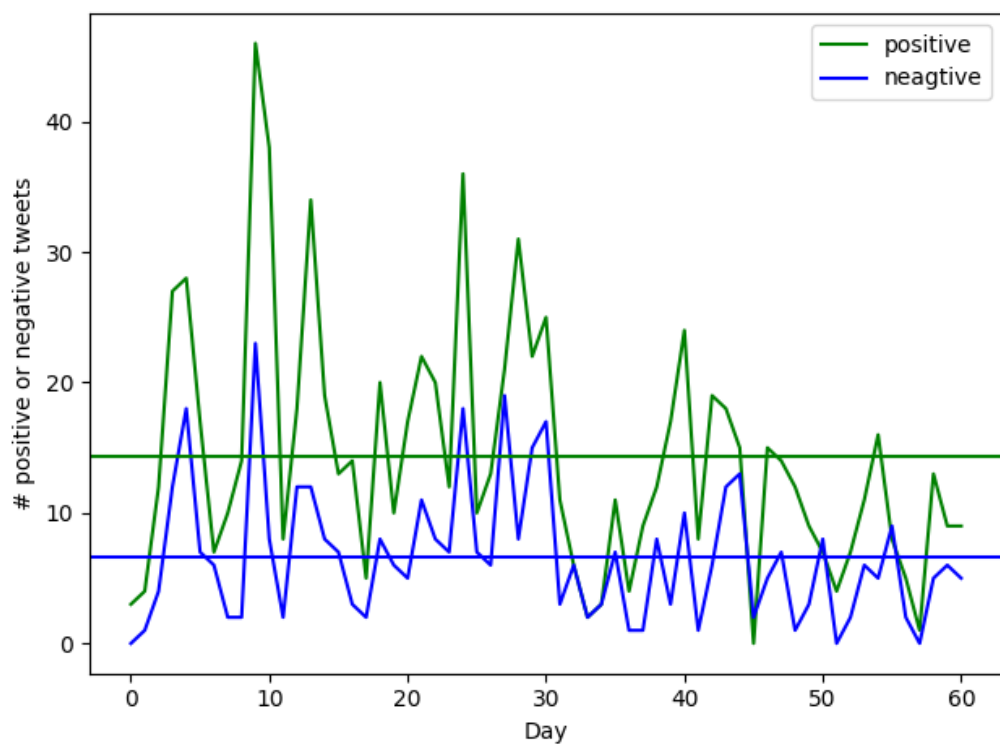


Figure 4: Task 3 d): The number of positive labeled tweets in green, and negative in blue over a two month period around the 2020 election.