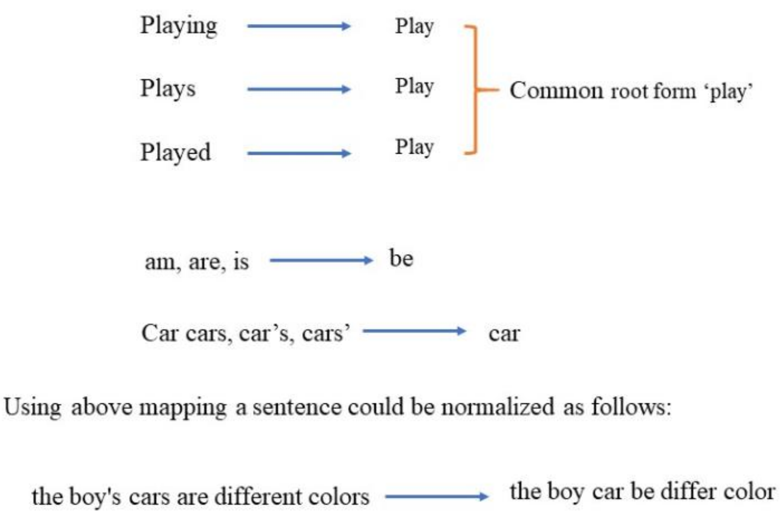
Alisher Siddikov
MSDS 453 - NLP
Assignment 1
4/29/2020

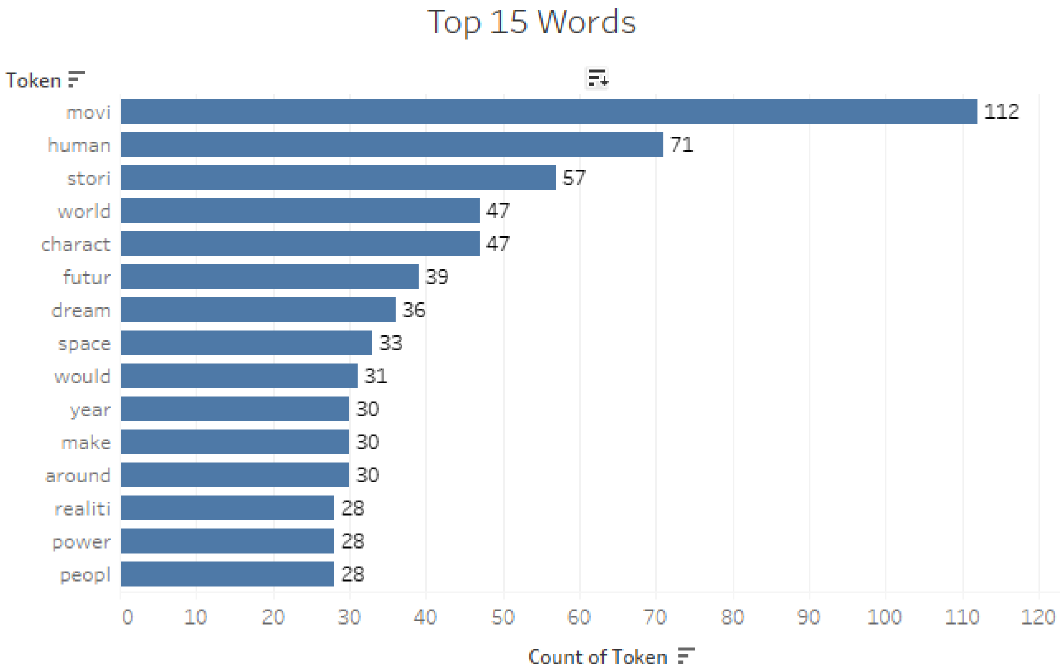
I started this text analytics assignment by splitting the text of a collection of movie reviews (or corpus) into the tokens (single words) and then cleaning them: removed punctuations, non-alphabetical (special characters and numbers), and short words (less than 4). I also removed stop words, the words that are not useful for our analysis, typically extremely common words such as "the", "of", "to", and so forth in English. I also converted each token to a lower case and then normalized them by stemming. Stemming is the process of reducing inflection in words to their common root forms, such as mapping a group of words to the same stem even if the stem itself is not a valid word in English. Here is a DataCamp example of how the stemming works.



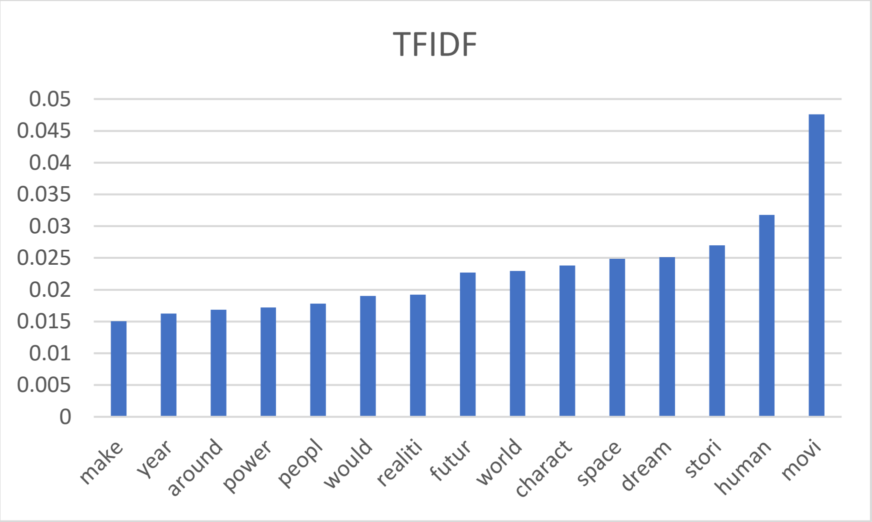
Using above mapping a sentence could be normalized as follows:

Using term frequency and inverse document frequency (TF-IDF) allowed me to find words that were characteristic for one movie review within a collection of movie reviews. TF means that the number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency. IDF means that the log of the number of documents divided by the number of documents that contain the word. Inverse data frequency determines the weight of rare words across all documents in the corpus. (Maklin, n.d.)

Here are the top fifteen words after we parse and clean the corpus. We can eliminate some of those



We can explore TFIDF score of the top fifteen words. We can see that words are less meaningful when the TFIDF average score is lower. As it gets higher, the word gets more meaningful.



Below are the k-means clustering results for TFIDF about the Wall-E movie. After comparing my three Wall-E movie reviews to the rest of the corpus, I was able to identify a single class that I believe characterizes my three documents, and that would be "robots." I can see that all three Wall-E reviews were clustered in the same cluster group along with Ex-Machina and Matrix movies, which contain similar themes and elements. Those movies were clustered more on robots, artificial intelligence, machines, and humanoid robots, and we can see those terms below. I believe that the El-Duderino movie does not fit this cluster.

```
 walle
 robot
 intellig
 caleb
 human
 nathan
 machina
 artifici
 machin
 suppli
Cluster 7 titles: PD_Doc1_Ex-Machina_Fembot_Prob.docx,
PD_Doc2_AI_Gods_Egos_Ex-Machina.docx, PD_Doc3_Ex-
Machina_AI_moviewithbrains.docx, AS_Doc1_Walle2.docx, AS_Doc1_Walle3.docx,
AS_Doc1_Walle1.docx, AB_Doc3_AI__What_Is_The_Matrix_.docx, ECC_Doc1_El-
Duderino.docx
```

Some of the terms that could fall within the "robots" class are "artificial intelligence," "machine," and "humanoid." Caleb Nathan is a person who meets the female robot, Ex-Machina. Therefore, Caleb, Nathan, and Machina are proper nouns. We can exclude them in the second round of our assignment. I believe the "robot" class meets the criteria for identifying classes because it is at an appropriate level of abstraction, and there is an adequate number of terms that can support the class.

However, Dec2Vec classification tells a different story than the TF-IDF classification. When I tried the Doc2Vec, I can see that cluster is not that meaningful. My three review documents are placed in two different clusters. The first cluster is more about robots, dreams, and sci-fi movies like Matrix, Ex-Machina, Avengers, Back to the Future, etc. The sixth cluster is about sci-fi movies, but it also includes wars and crimes such as Star Wars and Irishman. I noticed that when I restarted the Python kernel, I got different clustering results for Doc2Vec even though k-means had a seed number set to 89.

```
1: ['SA_DOC3_BladeRunnerReview.docx',
  'CMB_Doc1_Avengers_InfinityWar.docx',
  'AB_Doc1_The_Matrix_20.docx',
  'SL_Doc1_Off-to-the-Stars.docx',
  'SMN_Doc1_Back-Future-THR.docx',
  'SA_DOC1_WhenBladeRunner.docx',
  'ECC_Doc3_Film-Critics-Blind.docx',
  'ECC_Doc1_El-Duderino.docx',
  'PD_Doc3_Ex-Machina_AI_moviewithbrains.docx',
  'CT_DOC2_CheckCashesIn.docx',
  'AS_Doc1_Walle2.docx',
  'AS_Doc1_Walle3.docx',
  'SA_DOC2_WhyBladeRunner.docx',
  'SJB_Doc3_This-Time-the-Dream_s.docx',
  'AB_Doc3_AI__What_Is_The_Matrix_.docx',
  'RRM_Doc1_Minority-Report-Predictions.docx',
  'NA_Doc1_Toy-Story-Trilogy-Epilogue.docx',
  'MAS_Doc3_TheLordoftheRings_TheFellowshipoftheRing.docx',
  'SMN_Doc3_From-Archives-LATimes.docx'],
6: ['AB_Doc2_The_Matrix_Predicted.docx',
  'NA_Doc3_Toy-Story-4-Existential-Terror.docx',
  'MAS_Doc1_TheLordoftheRings_TheFellowshipoftheRing.docx',
  'AS_Doc1_Walle1.docx',
  'CT_DOC3_BlankCheck.docx',
  'MRR_Doc2_Cats-Review-Will.docx',
  'MCL_Doc3_StarWarsThe.docx',
  'In_The_Irishman_BG.docx'],
```

I believe that Doc2Vec did not cluster well because we need a larger and higher quality dataset than the corpus we currently have. However, the clusters for TFIDF were adequate for my Wall-E review. When we compared the TFIDF and Doc2Vec cluster results, we can see that TFIDF clustered movies better. During our discussion exercise, the following suggestion emerged: TFI-IDF might be useful for weighing how important terms are to a document, while Doc2Vec is useful in diving deeper into the document and measuring similarities between sentences.

Bibliography

Maklin, C. (n.d.). *TF IDF | TFIDF Python Example*. Retrieved from Towards Data Science: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76