Department of Computer Engineering

Natural Language Processing

Final Project

Phase 1[1]

Alireza Moradi
Student No.: 96521479

Prof. Sauleh Eetemadi
Spring 2021

---

[1] https://github.com/Alireza1044/CS224n-project-dataset

Contents

List of Figures

List of Tables

1 Source

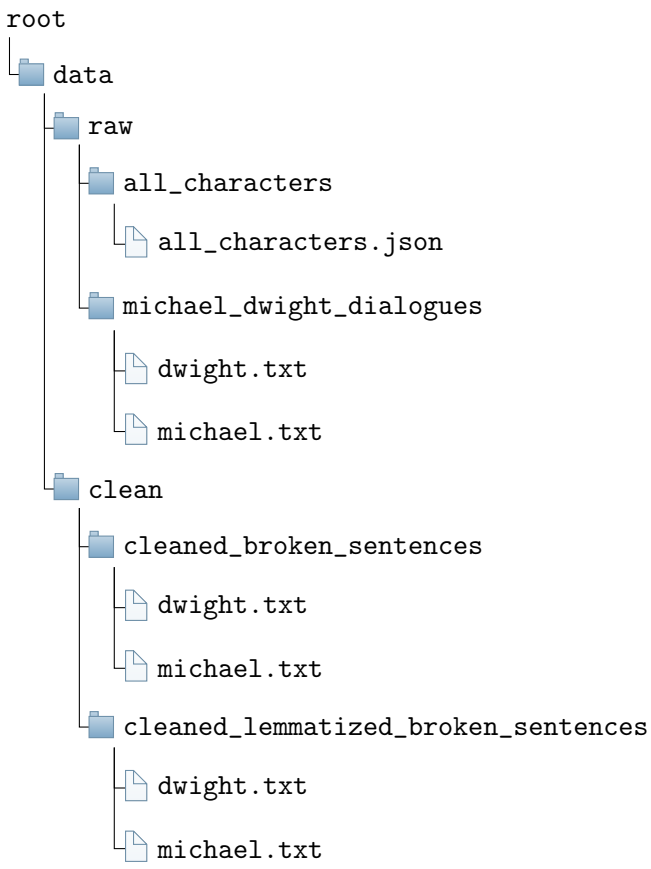
The dataset contains every dialogue ever said by Michael Scott and Dwight K. Schrute, two of the most popular characters from NBC's The Office. All of the data was obtained from OfficeQuotes.net, which assembled every line ever said on the show, along with the speaker's names.

2 Methods to Collect the Data

The data was crawled from the website using a python script and with the help of `Beautiful Soup` and `re` python packages. Each dialogue is in a `<div>` with `quote` in HTML source. Also, the speaker's name is in a `<b>` tag in the beginning of the sentence. There are some additional sentences in `[]` or `()` in the middle of some of the dialogues. I used `re` and its regex support to remove `[]` and `()` and everything inside them. There are some special characters in the source of the web pages, and they are replaced right after they are crawled. All of the dialogues will be saved in a python dictionary and after crawling is finished, the two chosen characters will be easily selected.

3 Data Format

The raw data, which contains every single dialogue of all characters in the show, is saved in a json format, and will be loaded to a python dictionary with the name of characters as keys and the dialogues, as a list of strings, as values (`root/data/raw/all_characters/all_characters.json`). Then then 2 classes (Dwight and Michael) are selected from this dictionary and saved to a text file (`root/data/raw/michael_dwight_dialogues`). After separating all sentences of each class (sentence tokenization) and applying the preprocessing steps, each class will be saved to a text file, each sentence in one line (`root/data/clean/cleaned_broken_sentences`). Lemmatized corpus is also saved in a separate folder with the same naming criteria (`root/data/clean/cleaned_lemmatized_broken_sentences`). Each text file's name is same as the name of that character, and that is how different classes are distinguishable.
File hierarchy is as follows:



4 Preprocesses

There are several cleaning and preprocessing steps:

- First of all, everything regarding a single class will be concatenated together, and will be broken to individual sentences using `nltk`'s sentence tokenizer.
- There are some words with accented characters, like café. These characters are replaced by a normal one using a package called `unidecode`.
- Contractions like `It's` will be expanded (`It is`) using another package, `contractions`.
- `nltk`'s word lemmatizer is applied to each word in the corpus (depending on the final results of the project, I might remove this step).
- Sentences with less than 5 words are discarded.

Stopwords and punctuations are **NOT** removed. Depending on the results this might be added in the future.
The dataset for this project is considered clean if each sentence has minimum modification (apart from those mentioned above) before it can be fed to a neural network.
The size of the data before and after cleaning and preprocessing is as follows:

	Sentence Count	Word Count
Before	13506	106269
After	8895	90068

Table 1: First Class (Dwight) Size Comparison

	Sentence Count	Word Count
Before	25384	210470
After	17069	181144

Table 2: First Class (Michael) Size Comparison

5 Tagging

Each sentence is considered one unit of data. A classifier should be able to assign each sentence to each of the two characters.
Each sentence has the name of the speaker in the beginning, which was used to tag the dataset.

6 Statistics

6.1 Sentence/Unit Count

There are a total of 25964 sentences in the corpus.

	Sentences/Units	Words	Distinct Words
Count	25964	271212	15445

Table 3: Sentence, Words and Distinct Words Count

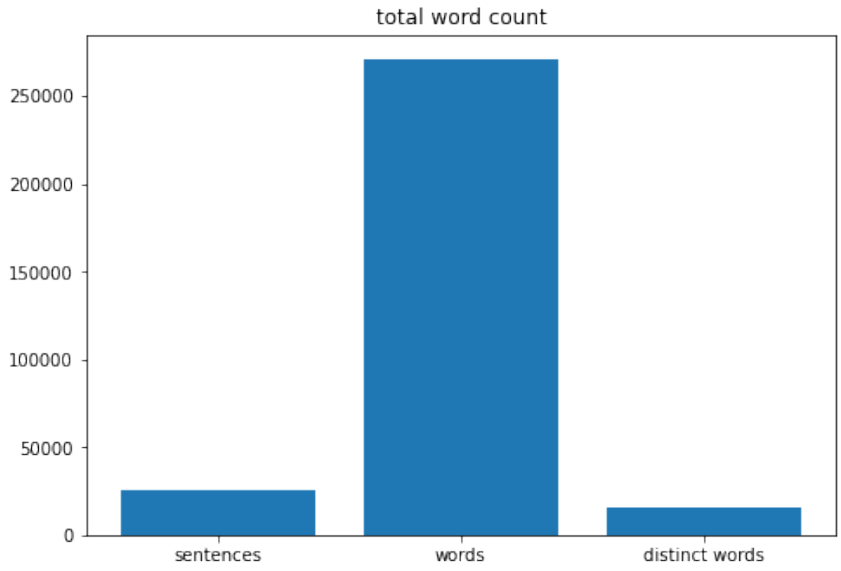


Figure 1: Sentence, Words and Distinct Words

	Common	First (Dwight)	Second (Michael)
Count	3952	4372	3169

Table 4: Distinct Words Count Based on Each Class

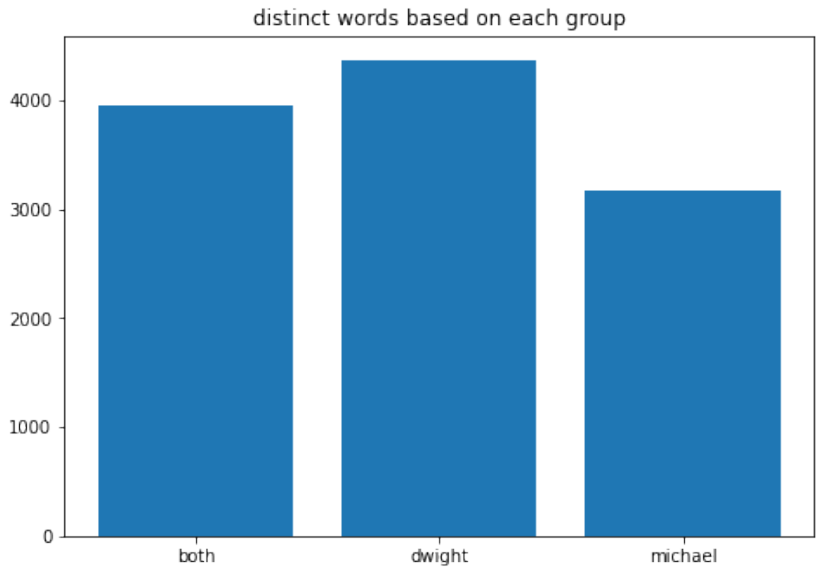


Figure 2: Distinct Words Based on Each Class

	sensei	deputy	j	belsnickel	farmer	pum	alliance	superior	grandfather	knot
Count	15	11	11	11	10	9	9	8	8	8

Table 5: First Class (Dwight) Most Repeated Words

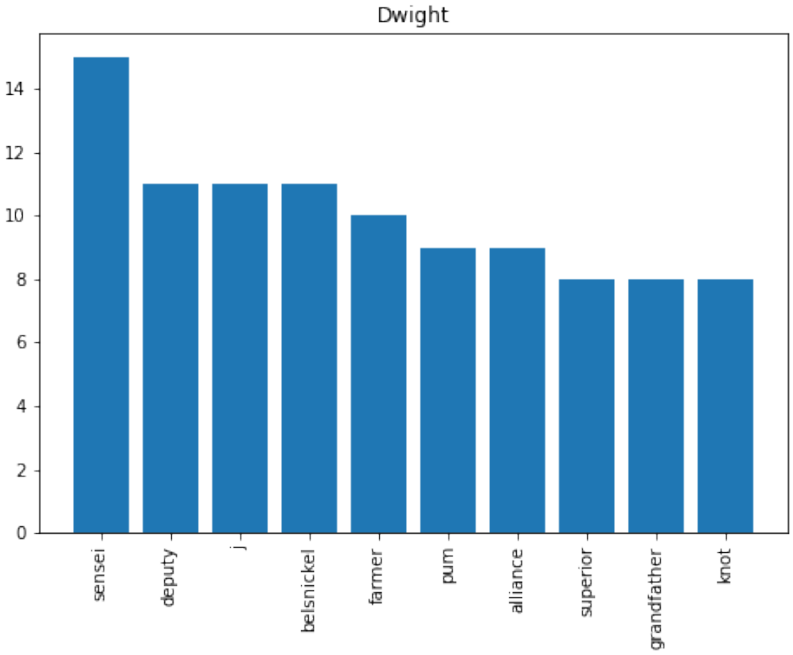


Figure 3: First Class (Dwight) Most Repeated Words

	na	beep	jamaica	daddy	googi	luke	comedy	lame	whether	martin
Count	32	24	20	17	16	15	14	14	14	13

Table 6: Second Class (Michael) Most Repeated Words

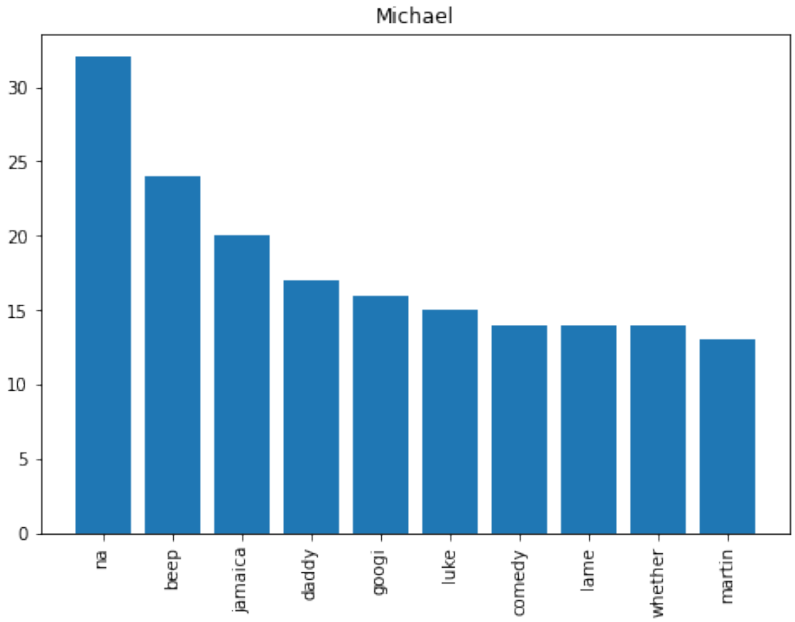


Figure 4: Second Class (Michael) Most Repeated Words

	mose	cousin	sheriff	nellie	enemy	goat	hay	wolf	beet	officer
Count	76.42	34.19	32.18	28.16	26.15	24.13	20.11	16.09	16.09	16.09

Table 7: First Class (Dwight) Most Repeated Words - Relative Normalized Frequency

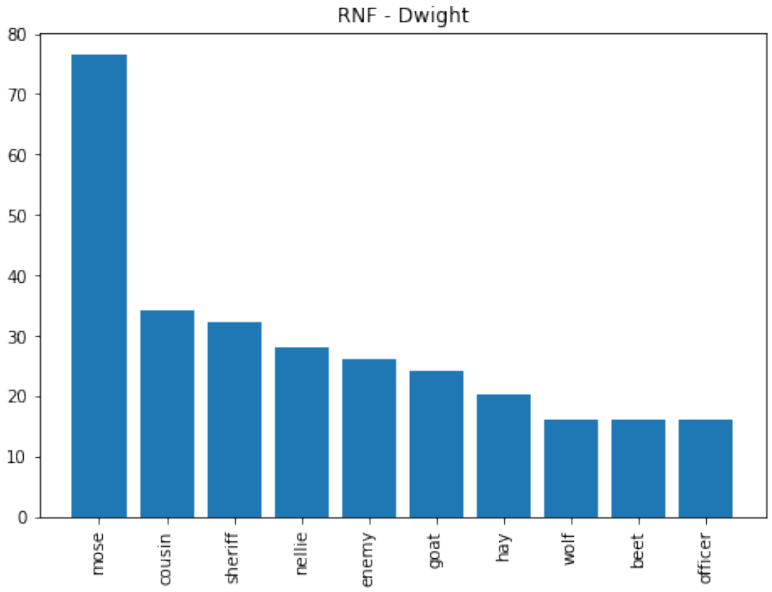


Figure 5: First Class (Dwight) - Relative Normalized Frequency

	umm	dating	anybody	everybody	jerk	somebody	cry	bob	lover	condo
Count	13.92	11.93	10.11	9.99	9.94	8.86	8.45	7.95	7.95	7.95

Table 8: Second Class (Michael) Most Repeated Words - Relative Normalized Frequency

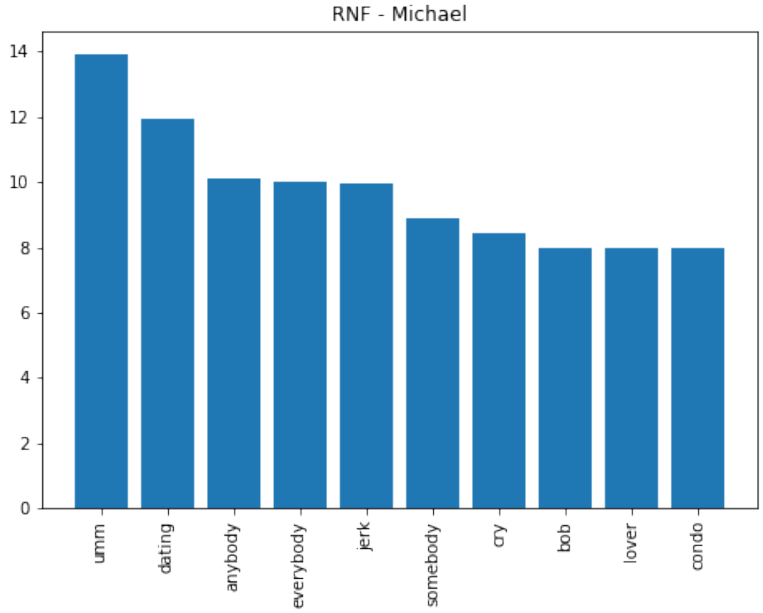


Figure 6: Second Class (Michael) - Relative Normalized Frequency

	sensei	j	deputy	belsnickel	farmer	alliance	pum	superior	esther	grandfather
$\text{Count}_{(e-05)}$	11.54	8.465	8.465	8.465	7.695	6.926	6.926	6.156	6.156	6.156

Table 9: First Class (Dwight) Most Repeated Words - TF-IDF

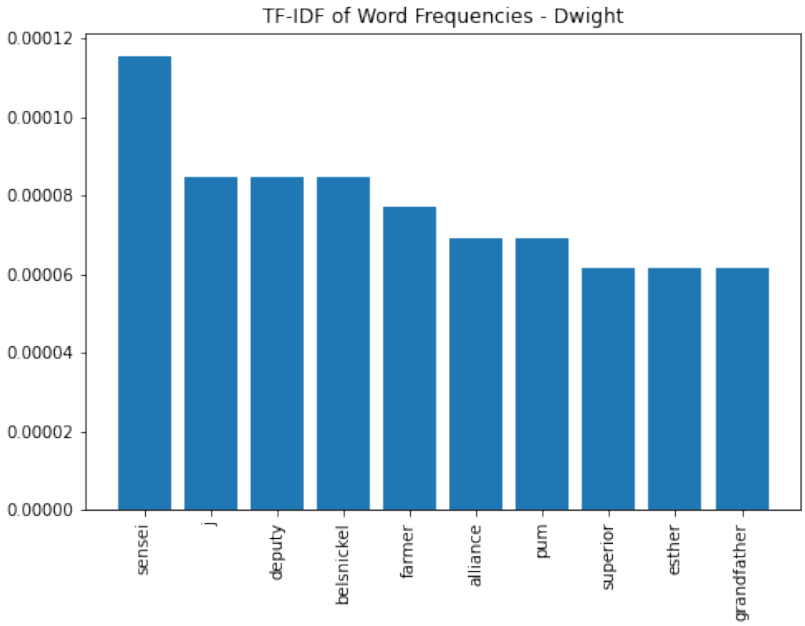


Figure 7: First Class (Dwight) - TF-IDF

	na	beep	jamaica	daddy	googi	luke	comedy	whether	lame	attitude
$\text{Count}_{(e-05)}$	12.24	9.183	7.652	6.505	6.122	5.739	5.357	5.357	5.357	4.974

Table 10: Second Class (Michael) Most Repeated Words - TF-IDF

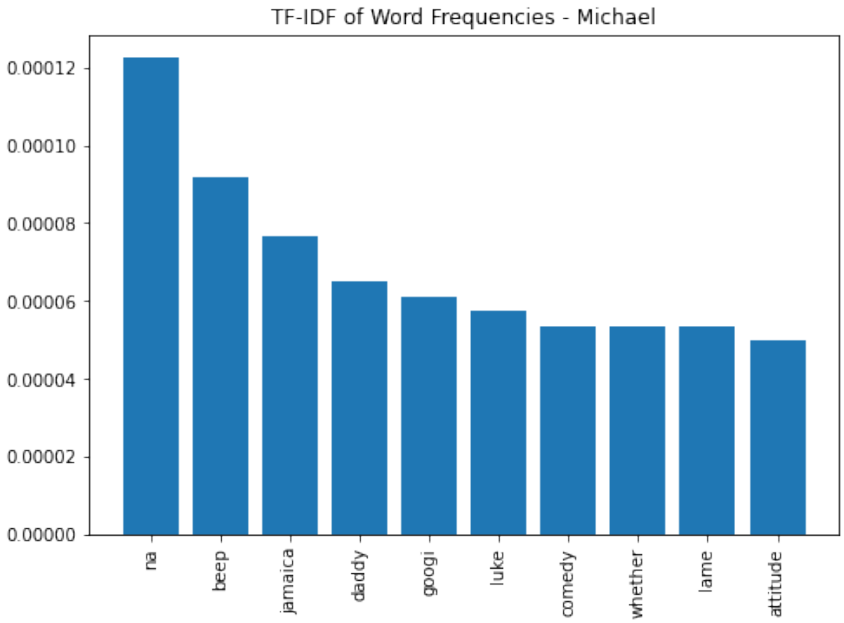


Figure 8: Second Class (Michael) - TF-IDF

	.	,	you	i	is	to	the	a	?	it
Count	20181	16280	8184	7952	6386	6244	6207	5843	4584	4305

Table 11: Highest Frequency Words

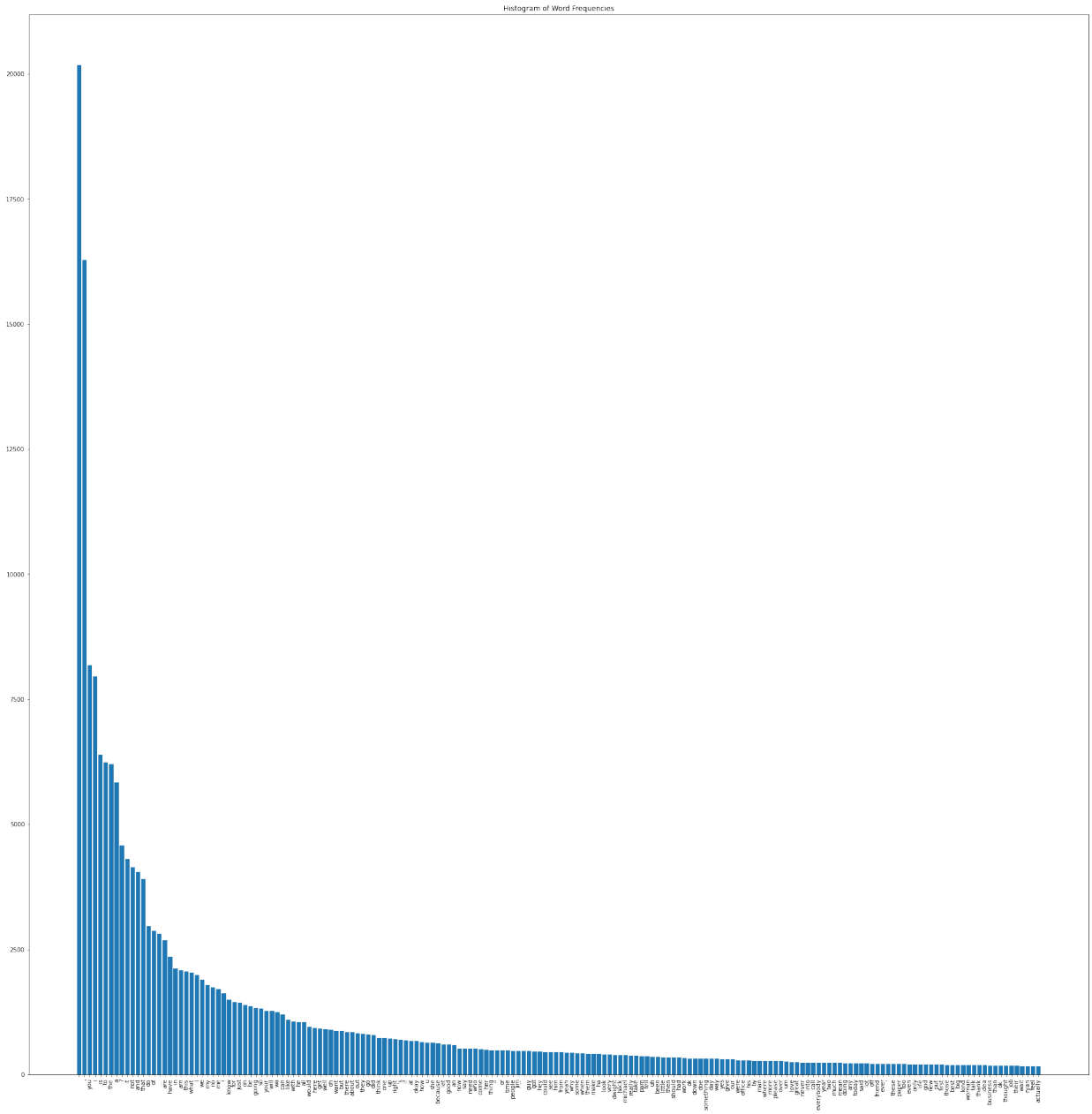


Figure 9: Highest Frequency Words