CS119 QUIZ 5 - Darcy Corson

PART 1 - ABOUT TF-IDF

1. The TF value for a word in a document is a measure of how frequently that word occurs in a given document. The IDF is a measure of how important a word in a document is to a collection of documents, or corpus. That is, it is a measure of how common or rare a word is in an entire document set. Thus, while TF can be calculated for a word in a single document, the calculation of IDF can only be performed for a set of documents, because the IDF value means to express something about a set of documents.

2. Scikit-Learn's IDF calculation methodology is different from the 'standard' IDF methodology in two ways. First, and unlike for the 'standard' IDF equation, some constant 1 is added to both the numerator and denominator of the Scikit-Learn's IDF equation as though an extra document was seen that contains every term in the collection exactly once. This has the effect of eliminating the possibility of the denominator being 0, wholly preventing division by 0. Division by 0 is undesirable because it necessarily results in an IDF value that is undefined. Second, and also unlike the 'standard' IDF equation, the Scikit-Learn's IDF equation requires the addition of 1 to the final result, as an act of 'smoothing.' The result of adding one ensures that no word ends up with a 0 IDF score just because it is common. When looking for information or sorting documents into categories, it is important that every word has the opportunity to help in that process, because common words can play a role in defining the content and context of the documents. By ensuring that no word has a zero weight, the Scikit-Learn's IDF equation may make better use of all the words, potentially leading to improved performance in tasks like search and document classification.

PART 2 - PRESIDENTIAL SPEECHES

3. For each speech, the mapper receives raw text from one of the 10 inaugural speeches as input. The input is the entire speech document, and it is cleaned before being analyzed; cleaning includes the provided actions and also the removal of single letter words. The mapper will tokenize the speech into words. For each word in a given inaugural speech, the mapper will emit a key-value pair where the key is a tuple of (word, document_id) and the value is the count of 1. The output will be key-value pairs of the format: ((word, document_id), 1). Before the reducer, Hadoop will sort and group these keys. Note that the mapper calculates the raw frequency (which is then normalized in the reducer to calculate TF). I assume that this is what the assignment instructions mean when they say that the mapper should be used for TF; the mapper should not calculate the actual TF part of the TF-IDF because it does not have information about the total number of words in the document to normalize the term frequency. Thus, it makes the most sense for the mapper's role to be to process records independently and prepare them for the reducer, which then aggregates these intermediate results.

   The reducer receives (word, document_id), [1, 1, 1, ...]. It sums the counts for each word to find the total term frequency (TF) for that word in the document. Then, it calculates the document frequency (DF) for each word, which is the number of documents in the corpus that contain that word. The IDF is calculated using the standard approach: IDF = log((Total number of documents) / (DF). The TF-IDF score is found by multiplying the TF by the IDF. The reducer maintains a local priority queue that

keeps track of the top 20 highest TF-IDF scores for the words that it processes. The reducer's final output will be the top-20 words with the highest TF-IDF scores for each document.

4. To begin the analysis, the preprocessing script below is designed and executed to preliminarily access and normalize the text data of the last 10 inaugural speeches.

```
#!/usr/bin/env python
import nltk
import nltk.corpus
from nltk.corpus import inaugural
nltk.download('inaugural')

REPLACEMENTS = [
    (". ",   ".\n"),
    ("Mr.\n", "Mr. "),
    ("Ms.\n", "Ms. "),
    ("Mrs.\n", "Mrs. "),
]

speech_names = nltk.corpus.inaugural.fileids()[-10:]

for name in speech_names:
    text = inaugural.raw(name)
    text = ''.join([i if ord(i) < 128 else ' ' for i in text])
    text = ' '.join(text.split())

    for old, new in REPLACEMENTS:
        text = text.replace(old, new)

    # Write the cleaned text to a new file
    with open('inaug_' + name, 'w') as writer:
        writer.write(text + '\n')
```

The script begins by importing the necessary modules from the Natural Language Toolkit (NLTK). Then, it downloads the inaugural corpus from NLTK, which contains a collection of inaugural speeches from various presidents of the United States. A list of REPLACEMENTS is defined for standardizing certain patterns in the text. For example, after a period, a newline character is added to make the text more readable. Corrections for abbreviations like "Mr.", "Ms.", and "Mrs." are also employed to avoid breaking lines incorrectly after these terms. Though not necessary for this analysis per se, such standardization of text is good practice and was thus performed. Specifically, the script fetches the file IDs of the 10 inaugural speeches from the inaugural corpus and, for each of these speeches, performs the following preprocessing steps: it retrieves the raw text of the speech, it removes any non-ASCII characters and replaces them with a space to avoid encoding issues that could complicate text analysis, it reduces any sequences of multiple spaces to a single space thus leading to a more consistent and clean text, and it applies the previously defined replacements to handle periods and certain abbreviations appropriately. Finally, the script writes the processed text to a new file. Each file is named by appending 'inaug_' to the original file name, and the content is written with an additional newline character at the end for better formatting. The speech files are now ready to pass to the mapper.

The mapper script plays an important initial role in analyzing the inaugural speeches to identify the top-20 TF-IDF values for each of the last 10 presidential speeches.

```
!/usr/bin/env python3
import sys
import os
import re
import string

def clean_text(text):
    text = text.lower()
    text = re.sub(r'\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), ' ', text)
    text = re.sub('\w*\d\w*', '', text)
    text = re.sub('\s+', ' ', text)
    return text.strip()

file_path = os.environ['map_input_file']
document_id = os.path.basename(file_path)

for content in sys.stdin:
    words = clean_text(content).split()

    # Emit each word with the document_id and a count of 1, filtering out single-character words
    for word in words:
        if len(word) > 1:  # Only emit words longer than 1 character
            print(f'{word}\t{document_id}\t1')
```

Specifically, it processes the text of each speech to prepare it for TF-IDF analysis. The script begins by defining a clean_text function, which standardizes the raw text by converting it to lowercase, stripping annotations, punctuation, numbers, and excessive whitespace, thereby ensuring that the analysis is focused on meaningful content. After obtaining the name of the input file, which serves as the document identifier, the script reads the content from standard input. Then, it cleans and tokenizes the text, excluding single-character tokens which are generally not significant for TF-IDF analysis.

For each meaningful word, the mapper emits a key-value pair where the key is the word itself concatenated with its document identifier, and the value is the integer one. These pairs are ready for use by the reducer, where the actual calculation of TF and subsequent IDF is performed – please see the response to prompt 3) for a short discussion of why the decision to handle the calculation of TF-IDF this way was made. The output of the mapper is designed to ensure accurate TF calculation for the reducer, setting the stage for the final TF-IDF computation.

The reducer computes the TF-IDF values for words across the 10 presidential inaugural speeches.

```
#!/usr/bin/env python
import sys
import math
from collections import defaultdict

# Number of documents in the corpus
total_documents = 10

# Data structures for calculations
word_counts = defaultdict(lambda: defaultdict(int))  # word -> doc -> count
doc_word_counts = defaultdict(int)  # doc -> total words
doc_count = defaultdict(int)  # word -> number of documents containing word

for line in sys.stdin:
    line = line.strip()
    word, document_id, count = line.split('\t', 2)
    count = int(count)
    word_counts[word][document_id] += count
    doc_word_counts[document_id] += count

# Compute DF for each word
for word in word_counts:
    doc_count[word] = len(word_counts[word])

# Calculate TF-IDF for each word in each document and keep DF information
tf_idf_scores = defaultdict(list)  # doc -> [(tf_idf, word, df), ...]

for word, docs in word_counts.items():
    idf = math.log(total_documents / doc_count[word])
    for document_id, count in docs.items():
        tf = count / float(doc_word_counts[document_id])
        tf_idf = tf * idf
        df = doc_count[word]
        tf_idf_scores[document_id].append((tf_idf, word, df))

# Output the top 20 words by TF-IDF for each document along with their DF for error checking
for document_id, scores in tf_idf_scores.items():
    top_20 = sorted(scores, key=lambda x: x[0], reverse=True)[:20]
    for tf_idf, word, df in top_20:
        print(f"{document_id}\t{word}\t{tf_idf}\t{df}")
```

After the mapper processes and emits key-value pairs representing words and their occurrences in each document, the reducer takes over to aggregate these counts and calculate the TF-IDF scores. The reducer's script begins by initializing data structures to store word counts within each document, total word counts per document, and document frequency counts for each word across all documents. It processes input from the mapper line by line, parsing the word, document identifier, and count, and updating its data structures accordingly. With this data, the reducer calculates the DF for each word, which is the number of documents the word appears in. This DF is used to compute the IDF score. The script calculates TF by dividing the count of a word in a document by the total word count of that document, then multiplies TF by IDF to get the TF-IDF score for each word in each document. Finally, for each document, the reducer sorts the words by their TF-IDF scores in descending order and outputs the top 20 words along with their TF-IDF scores and document frequency for manual checking. This output highlights the

words that are uniquely significant in individual speeches relative to the entire corpus of the last 10 presidential inauguration speeches.

The Hadoop Streaming command below is a critical step that orchestrates the entire MapReduce process described above.

```
.hadoop jar /usr/lib/hadoop/hadoop-streaming.jar \
    -files inaug_mapper.py,inaug_reducer.py \
    -mapper "python3 inaug_mapper.py" \
    -reducer "python3 inaug_reducer.py" \
    -input /user/hadoop/inaug_speeches/* \
    -output /user/hadoop/output/inaug_results \
    -numReduceTasks 1
```

The command explicitly defines the input directory '/user/hadoop/inaug_speeches/*', where all speeches are stored, and the output directory '/user/hadoop/output/inaug_results', where the results, including the top-20 TF-IDF scores per speech, are written. Setting '-numReduceTasks 1' ensures that a single reducer is used for this job, which is suitable for this assignment's scale and helps to maintain the order of results as required by the assignment spec. Executing this Hadoop Streaming command initiates the MapReduce job.

As discussed, the output of the MapReduce job are the top-20 TF-IDF words for each of the 10 inaugural speeches. These are listed below.

```
inaug_2005-Bush.txt    xand    0.00564912927623662    1
inaug_2005-Bush.txt    seen    0.00354457155346203    3
inaug_2005-Bush.txt    excuse  0.003389477565741971   1
inaug_2005-Bush.txt    questions       0.003389477565741971   1
inaug_2005-Bush.txt    fire    0.0031588575317646715  2
inaug_2005-Bush.txt    freedom 0.0029562688348791282  8
inaug_2005-Bush.txt    tyranny 0.002953809627885025   3
inaug_2005-Bush.txt    human   0.002697617463810074   4
inaug_2005-Bush.txt    defended        0.0023691431488235036  2
inaug_2005-Bush.txt    goal    0.0023691431488235036  2
inaug_2005-Bush.txt    institutions    0.0023691431488235036  2
inaug_2005-Bush.txt    permanent       0.0023691431488235036  2
inaug_2005-Bush.txt    abandon 0.0022596517104946476  1
inaug_2005-Bush.txt    accepted        0.0022596517104946476  1
inaug_2005-Bush.txt    events  0.0022596517104946476  1
inaug_2005-Bush.txt    fulfill 0.0022596517104946476  1
inaug_2005-Bush.txt    influence       0.0022596517104946476  1
inaug_2005-Bush.txt    meant   0.0022596517104946476  1
inaug_2005-Bush.txt    mercy   0.0022596517104946476  1
inaug_2005-Bush.txt    ownership       0.0022596517104946476  1

inaug_2009-Obama.txt   carried 0.0029394703314817605  1
inaug_2009-Obama.txt   father  0.0020546015903414043  2
inaug_2009-Obama.txt   calls   0.001959646887654507   1
inaug_2009-Obama.txt   charter 0.001959646887654507   1
inaug_2009-Obama.txt   conflict        0.001959646887654507   1
inaug_2009-Obama.txt   cooperation     0.001959646887654507   1
inaug_2009-Obama.txt   faced   0.001959646887654507   1
inaug_2009-Obama.txt   false   0.001959646887654507   1
inaug_2009-Obama.txt   icy     0.001959646887654507   1
inaug_2009-Obama.txt   short   0.001959646887654507   1
inaug_2009-Obama.txt   virtue  0.001959646887654507   1
inaug_2009-Obama.txt   waters  0.001959646887654507   1
```

inaug_2009-Obama.txt why 0.001959646887654507 1
inaug_2009-Obama.txt willingness 0.001959646887654507 1
inaug_2009-Obama.txt crisis 0.001559643798934732 4
inaug_2009-Obama.txt whether 0.001559643798934732 4
inaug_2009-Obama.txt off 0.001536986558713961 3
inaug_2009-Obama.txt rather 0.001536986558713961 3
inaug_2009-Obama.txt less 0.0015216082410050788 6
inaug_2009-Obama.txt ambitions 0.0013697343935609365 2

inaug_2001-Bush.txt civility 0.00603165708708329 1
inaug_2001-Bush.txt story 0.0060005941838517035 4
inaug_2001-Bush.txt affirm 0.0031619605352339883 2
inaug_2001-Bush.txt commitment 0.0031619605352339883 2
inaug_2001-Bush.txt compassion 0.0031619605352339883 2
inaug_2001-Bush.txt sometimes 0.0031538252896553664 3
inaug_2001-Bush.txt angel 0.003015828543541645 1
inaug_2001-Bush.txt directs 0.003015828543541645 1
inaug_2001-Bush.txt grand 0.003015828543541645 1
inaug_2001-Bush.txt laws 0.003015828543541645 1
inaug_2001-Bush.txt rides 0.003015828543541645 1
inaug_2001-Bush.txt stakes 0.003015828543541645 1
inaug_2001-Bush.txt whirlwind 0.003015828543541645 1
inaug_2001-Bush.txt everyone 0.002365368967241525 3
inaug_2001-Bush.txt principles 0.002365368967241525 3
inaug_2001-Bush.txt generous 0.0021079736901559926 2
inaug_2001-Bush.txt honored 0.0021079736901559926 2
inaug_2001-Bush.txt humanity 0.0021079736901559926 2
inaug_2001-Bush.txt share 0.0021079736901559926 2
inaug_2001-Bush.txt beyond 0.001815709706771304 5

inaug_1993-Clinton.txt season 0.005847835156810276 1
inaug_1993-Clinton.txt renewal 0.0040874613649120005 2
inaug_1993-Clinton.txt whom 0.0040874613649120005 2
inaug_1993-Clinton.txt posterity 0.0030655960236840004 2
inaug_1993-Clinton.txt raised 0.0030655960236840004 2
inaug_1993-Clinton.txt sake 0.0030655960236840004 2
inaug_1993-Clinton.txt spring 0.0030655960236840004 2
inaug_1993-Clinton.txt compete 0.002923917578405138 1
inaug_1993-Clinton.txt serving 0.002923917578405138 1
inaug_1993-Clinton.txt change 0.0029190035643770896 6
inaug_1993-Clinton.txt idea 0.002908859466267159 4
inaug_1993-Clinton.txt capitol 0.0022932815320494022 3
inaug_1993-Clinton.txt powerful 0.0017453156797602955 4
inaug_1993-Clinton.txt done 0.001621668646876161 6
inaug_1993-Clinton.txt ceremony 0.0015288543546996014 3
inaug_1993-Clinton.txt depression 0.0015288543546996014 3
inaug_1993-Clinton.txt forth 0.0015288543546996014 3
inaug_1993-Clinton.txt founders 0.0015288543546996014 3
inaug_1993-Clinton.txt mission 0.0015288543546996014 3
inaug_1993-Clinton.txt preserve 0.0015288543546996014 3

inaug_1997-Clinton.txt enough 0.0034596919664538392 3
inaug_1997-Clinton.txt century 0.0034164266660798117 7
inaug_1997-Clinton.txt awful 0.00220554127681422 1
inaug_1997-Clinton.txt coast 0.00220554127681422 1
inaug_1997-Clinton.txt doors 0.00220554127681422 1
inaug_1997-Clinton.txt information 0.00220554127681422 1
inaug_1997-Clinton.txt labors 0.00220554127681422 1
inaug_1997-Clinton.txt moved 0.00220554127681422 1
inaug_1997-Clinton.txt quest 0.00220554127681422 1
inaug_1997-Clinton.txt religious 0.00220554127681422 1
inaug_1997-Clinton.txt saved 0.00220554127681422 1

inaug_1997-Clinton.txt seemed 0.00220554127681422 1
inaug_1997-Clinton.txt solve 0.00220554127681422 1
inaug_1997-Clinton.txt sustain 0.00220554127681422 1
inaug_1997-Clinton.txt waste 0.00220554127681422 1
inaug_1997-Clinton.txt human 0.0021941827870549688 4
inaug_1997-Clinton.txt dreams 0.0017298459832269196 3
inaug_1997-Clinton.txt streets 0.0017298459832269196 3
inaug_1997-Clinton.txt bridge 0.0015416071958181037 2
inaug_1997-Clinton.txt bright 0.0015416071958181037 2

inaug_2021-Biden.txt re 0.00562291842000988 1
inaug_2021-Biden.txt me 0.004102237708838301 4
inaug_2021-Biden.txt virus 0.0037486122800065866 1
inaug_2021-Biden.txt days 0.003430121949646542 3
inaug_2021-Biden.txt story 0.0033563763072313375 4
inaug_2021-Biden.txt truth 0.0032752094270128208 2
inaug_2021-Biden.txt lies 0.00281145921000494 1
inaug_2021-Biden.txt objects 0.00281145921000494 1
inaug_2021-Biden.txt prevailed 0.00281145921000494 1
inaug_2021-Biden.txt uniting 0.00281145921000494 1
inaug_2021-Biden.txt cry 0.0019651256562076926 2
inaug_2021-Biden.txt get 0.0019651256562076926 2
inaug_2021-Biden.txt violence 0.0019651256562076926 2
inaug_2021-Biden.txt ve 0.0019600696855123093 3
inaug_2021-Biden.txt disagree 0.0018743061400032933 1
inaug_2021-Biden.txt disagreement 0.0018743061400032933 1
inaug_2021-Biden.txt doesn 0.0018743061400032933 1
inaug_2021-Biden.txt everything 0.0018743061400032933 1
inaug_2021-Biden.txt extremism 0.0018743061400032933 1
inaug_2021-Biden.txt folks 0.0018743061400032933 1

inaug_2017-Trump.txt protected 0.0055844479959545465 2
inaug_2017-Trump.txt countries 0.004793723302555265 1
inaug_2017-Trump.txt obama 0.004793723302555265 1
inaug_2017-Trump.txt dreams 0.004177560042768689 3
inaug_2017-Trump.txt capital 0.0033506687975727277 2
inaug_2017-Trump.txt everyone 0.003342048034214951 3
inaug_2017-Trump.txt breath 0.0031958155350368437 1
inaug_2017-Trump.txt exists 0.0031958155350368437 1
inaug_2017-Trump.txt glorious 0.0031958155350368437 1
inaug_2017-Trump.txt industry 0.0031958155350368437 1
inaug_2017-Trump.txt mountain 0.0031958155350368437 1
inaug_2017-Trump.txt politicians 0.0031958155350368437 1
inaug_2017-Trump.txt righteous 0.0031958155350368437 1
inaug_2017-Trump.txt shine 0.0031958155350368437 1
inaug_2017-Trump.txt stops 0.0031958155350368437 1
inaug_2017-Trump.txt transferring 0.0031958155350368437 1
inaug_2017-Trump.txt trillions 0.0031958155350368437 1
inaug_2017-Trump.txt winning 0.0031958155350368437 1
inaug_2017-Trump.txt again 0.003190444562036028 6
inaug_2017-Trump.txt while 0.002886109009965074 5

inaug_2013-Obama.txt complete 0.003891290890798115 2
inaug_2013-Obama.txt until 0.003493151269804457 3
inaug_2013-Obama.txt evident 0.0033403071948656373 1
inaug_2013-Obama.txt knowing 0.0033403071948656373 1
inaug_2013-Obama.txt she 0.003113032712638492 2
inaug_2013-Obama.txt requires 0.0029109593915037143 3
inaug_2013-Obama.txt happiness 0.0023287675132029713 3
inaug_2013-Obama.txt compel 0.002226871463243758 1
inaug_2013-Obama.txt harm 0.002226871463243758 1
inaug_2013-Obama.txt initiative 0.002226871463243758 1

CS119 QUIZ 5 - Darcy Corson

inaug_2013-Obama.txt lessons 0.002226871463243758 1
inaug_2013-Obama.txt thrives 0.002226871463243758 1
inaug_2013-Obama.txt train 0.002226871463243758 1
inaug_2013-Obama.txt enduring 0.0017465756349022284 3
inaug_2013-Obama.txt principles 0.0017465756349022284 3
inaug_2013-Obama.txt require 0.0017465756349022284 3
inaug_2013-Obama.txt self 0.0017465756349022284 3
inaug_2013-Obama.txt creed 0.0016758877673112799 5
inaug_2013-Obama.txt capacity 0.001556516356319246 2
inaug_2013-Obama.txt debates 0.001556516356319246 2

inaug_1989-Bush.txt breeze 0.0051930200563690705 1
inaug_1989-Bush.txt door 0.004154416045095257 1
inaug_1989-Bush.txt fact 0.003629765251317321 2
inaug_1989-Bush.txt word 0.003629765251317321 2
inaug_1989-Bush.txt don 0.003258383773547865 3
inaug_1989-Bush.txt mr 0.003258383773547865 3
inaug_1989-Bush.txt blowing 0.0031158120338214425 1
inaug_1989-Bush.txt expression 0.0031158120338214425 1
inaug_1989-Bush.txt loyal 0.0031158120338214425 1
inaug_1989-Bush.txt crucial 0.002177859150790393 2
inaug_1989-Bush.txt engagement 0.002177859150790393 2
inaug_1989-Bush.txt seems 0.002177859150790393 2
inaug_1989-Bush.txt mean 0.00217225584903191 3
inaug_1989-Bush.txt begins 0.0020772080225476284 1
inaug_1989-Bush.txt bow 0.0020772080225476284 1
inaug_1989-Bush.txt executive 0.0020772080225476284 1
inaug_1989-Bush.txt heads 0.0020772080225476284 1
inaug_1989-Bush.txt involved 0.0020772080225476284 1
inaug_1989-Bush.txt leadership 0.0020772080225476284 1
inaug_1989-Bush.txt majority 0.0020772080225476284 1

inaug_1985-Reagan.txt increase 0.0037019052942026463 1
inaug_1985-Reagan.txt human 0.0033145565059756413 4
inaug_1985-Reagan.txt federal 0.0032344009494254425 2
inaug_1985-Reagan.txt nuclear 0.002903471393068978 3
inaug_1985-Reagan.txt weapons 0.002903471393068978 3
inaug_1985-Reagan.txt number 0.0027764289706519846 1
inaug_1985-Reagan.txt senator 0.002587520759540354 2
inaug_1985-Reagan.txt tax 0.002587520759540354 2
inaug_1985-Reagan.txt reduce 0.002419559494224148 3
inaug_1985-Reagan.txt beginning 0.0019406405696552659 2
inaug_1985-Reagan.txt destroy 0.0019406405696552659 2
inaug_1985-Reagan.txt program 0.0019406405696552659 2
inaug_1985-Reagan.txt song 0.0019406405696552659 2
inaug_1985-Reagan.txt spend 0.0019406405696552659 2
inaug_1985-Reagan.txt economic 0.0019356475953793185 3
inaug_1985-Reagan.txt self 0.0019356475953793185 3
inaug_1985-Reagan.txt aimed 0.0018509526471013232 1
inaug_1985-Reagan.txt arm 0.0018509526471013232 1
inaug_1985-Reagan.txt arsenals 0.0018509526471013232 1
inaug_1985-Reagan.txt blow 0.0018509526471013232 1

The top-20 TF-IDF words from George W. Bush's inaugural speeches in 2001 and 2005 reveal significant insights into the main issues and concerns that occupied the United States during his presidency. In the 2001 speech, terms such as "civility," "compassion," "commitment," "principles," and "humanity" indicate a focus on unity, ethical governance, and social values, reflecting the context of a country seeking direction after a contentious election. Notably, "civility" and "compassion" suggest an appeal for national harmony and understanding, aligning with Bush's initial approach to domestic policy and social welfare. In contrast, the 2005 speech, occurring in the context of post-9/11 America, emphasizes terms like "freedom," "tyranny," "liberty," and "ownership." These words reflect the global challenges that the US faced at the time, particularly in terms of terrorism and the wars in Afghanistan and Iraq. Notably, "freedom" and the fight against "tyranny" highlight Bush's focus on spreading democracy and combating terrorism as central themes of his foreign policy. Additionally, "ownership" and "liberty" resonate with Bush's domestic agenda, promoting individual rights and economic freedom.

Similarly, the top-20 TF-IDF words from Barack Obama's inaugural speeches in 2009 and 2013 reflect the themes and concerns that were prevalent in the United States during his presidency. In the 2009 speech, words like "crisis," "ambitions," "conflict," and "cooperation" reflect Obama's focus on the significant economic challenges facing the country at the time as well as his appeal for unity and collective effort to overcome these obstacles. Notably, Obama's mention of "crisis" underscores the gravity of the economic downturn at the time, while "cooperation" and "ambitions" reflect Obama's call for national solidarity to rebuild the economy. Additionally, words like "father," "charter," and "virtue" suggest an emphasis on returning to foundational values and principles as a way to guide the nation through difficult times. In Obama's 2013 speech, terms such as "complete," "enduring," "principles," and "requires" point towards a narrative of continuity, resilience, and the ongoing work required to sustain democracy and progress. Notably, the words "complete" and "enduring" suggest a long-term perspective on the challenges and achievements of the nation, while "principles" and "requires" underscore the importance of adhering to fundamental democratic values and the effort needed to maintain them. Words like "happiness," "capacity," and "debates" highlight Obama's focus on the well-being of citizens and the importance of dialogue and discussion in a healthy democracy. Further, the presence of words related to personal responsibility and moral action, such as "compel," "harm," "initiative," and "self," in both speeches, reflects Obama's emphasis on the importance of individual responsibility in contributing to the collective good and addressing societal challenges. This narrative aligns with the broader issues the US was grappling with at the time, including economic recovery, healthcare reform, and climate change.