Mind Masters (Sittichai Chaikamol, Andrew Kuruvilla, Oren Moreno, Auny Nazmul)

ITAI 2376

April 18, 2024

Report on Text Treasure Hunt: The Vectorization Adventure

**Introduction**

The purpose of the report is to document the activities performed while completing “The Vectorization Student Notebook” part of “The Vectorization Adventure” assignment. It will detail the processes, techniques, and knowledge that group members used to solve the problems presented in the puzzle.

**Discussion**

The collaborative methods that our team used to complete the puzzle included discussions and updates through a groupchat and the creation of a shared Google Colab notebooks for writing, modifying, and testing possible code solutions. These methods proved effective for enabling collaborative problem-solving and keeping each other updated on progress. The “Vectorization Adventure” notebook begins by installing and importing the libraries that are used throughout it which includes Pandas, Scikit-learn, NLTK, Gensim, and Re. The first task was to discover what newsgroup from the popular 20 newsgroups dataset our initial clues alluded to. After reading the riddles, we determined that the keywords were “radio” and “voltage.” These words are both important to the topic of electronics, so we concluded that the newsgroup to search within was the “sci.electronics” dataset. With this knowledge, we created a variable to call the loaded data with the following code:  
  
categories = ['sci.electronics']

newsgroups\_train = fetch\_20newsgroups(subset='train', categories=categories)

The next task in the adventure was to extract the TF-IDF scores for our keywords from the selected dataset. This was started by the given code that creates a TF-IDF vectorizer, provides the stopwords to be ignored by it, fits it to the selected dataset, and creates a dataframe of the TF-IDF scores for each token in the dataset. We added to this provided code to get the TF-IDF scores for “radio” and “voltage” specifically:

# Checks for "radio" and "voltage" keywords in sci.electronics dataset and return their TF-IDF scores

if 'radio' in df.columns:

radio\_score = df.loc[df['radio'] > 0, 'radio'].values[0] #accesses the TF-IDF score for 'radio' in the dataframe

else:

radio\_score = None #returns none if 'radio' not found within the dataset

if 'voltage' in df.columns:

voltage\_score = df.loc[df['voltage'] > 0, 'voltage'].values[0]

else:

voltage\_score = None

return radio\_score, voltage\_score

The if/else statements were utilized here to avoid an error in the case that the keywords could not be found within the dataset. A modification to the provided code was also made to print out the TF-IDF scores for “radio” and “voltage”:

my\_text = newsgroups\_train.data #need .data attribute to get the text data from dataset

radio\_score, voltage\_score = extract\_keywords(my\_text) #Runs keyword function to get radio and voltage TF-IDF scores

print(f"TF-IDF score for 'radio': {radio\_score}")

print(f"TF-IDF score for 'voltage': {voltage\_score}")

We were then tasked with using a pre-trained Word2Vec model to find words similar to our keywords and the similarity scores of pairs of words. To solve this, we first loaded a Word2Vec model using the Gensim library’s “downloader” module:

import gensim.downloader as api

model = api.load('word2vec-google-news-300')

After this, we input our keywords into the given code that contains the functions for determining the most similar words and the similarity scores between two words. The final code-based task was to look for hidden patterns or codes in the data. The provided code gives a function to check for potential codes and email addresses which we then applied to our dataset with a simple modification:

my\_text = str(newsgroups\_train.data)

From this, we got a list of potential email addresses and the possible code, “MC68HC16.”

**Reflection**

In our project on sci.electronics, we faced several challenges. Extracting keywords using TF-IDF from technical texts about electromagnetic waves was complex due to our limited experience with such dense material.

Using semantic analysis tools like Word2Vec to find connections between terms also proved difficult. We often struggled to determine if the relationships identified were relevant to our clues. Additionally, employing regular expressions to detect patterns in technical specifications required multiple attempts to get right.

This project tested our ability to apply theoretical knowledge practically, underscoring the need for detailed analysis and persistence in handling specialized content. Despite the difficulties, it provided valuable insights into the real-world applications of text processing in electronics.

**Conclusion**

Through this project, our exploration into text processing within the electronics domain emphasized the critical role of transforming textual information into numerical data—akin to the process of text-to-vector transformations essential in NLP. This transformation is crucial as it allows the intricate details and nuances of electronic texts to be analyzed and understood in the computational language of AI. Every challenge we faced, from keyword extraction to semantic analysis, underscored the necessity of converting textual language into a format that AI algorithms can process mathematically. This experience not only solidified our understanding of the theoretical aspects of NLP but also illuminated its practical applications in deciphering and solving complex problems within electronics. The project was a profound testament to the indispensable nature of NLP in bridging human knowledge and machine understanding.

**Resources**

AWS Machine Learning University: Applications of Deep Learning to Text and Image Data - Module 2 Labs 2, 3.

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html>

<https://radimrehurek.com/gensim/downloader.html>

<http://qwone.com/~jason/20Newsgroups/>