Event keyword extraction from research articles

Mini-Project

(Natural Language Processing)

Rohit Burnwal (IIT2020175)

I. ABSTRACT

In today's age of information overload, it has become increasingly important to accurately and efficiently identify the most relevant information from large datasets. One area where this is especially true is in research articles, which often contain vast amounts of data, making it difficult for readers to quickly and effectively identify the key points of the article. In order to address this issue, keyword extraction techniques have been developed, which can automatically identify the most important concepts and topics discussed within an article.

This project focuses on developing a keyword extraction algorithm specifically designed for research articles. The algorithm utilizes natural language processing and machine learning techniques to analyze the text of an article and identify the most significant keywords. The resulting keywords can then be used to summarize the article, aid in search and retrieval, and provide a quick overview of the content for readers.

II. INTRODUCTION

Considering research & news articles, keywords form an important component since they provide a concise representation of the article's content. They also play a crucial role in locating the article from information retrieval systems, bibliographic databases and for search engine optimization. Keywords also help to categorize the article into the relevant subject or discipline. Conventional approaches of extracting keywords involve manual assignment of keywords based on the article content and the authors' judgment. This involves a lot of time & effort and also may not be accurate in terms of selecting the appropriate keywords. With the emergence of Natural Language Processing (NLP), keyword extraction has evolved into being effective as well as efficient. And in our project, we will combine the two — we apply NLP on a collection of articles to extract keywords.

III. DATASET USED

The dataset used for this article is a subset of the papers.csv dataset provided in the NIPS paper datasets on Kaggle. Neural Information Processing Systems (NIPS) is one of the top machine learning conferences in the world. This dataset includes the title and abstracts for all NIPS papers to date (ranging from the first 1987 conference to the current 2016 conference). The original dataset also contains the article text. However, since the focus is on understanding the concept of event keyword extraction and using the full article text could be computationally intensive, only abstracts have been used for NLP modelling.

IV. METHODOLOGY

There are various ways to compute the frequent keywords for event extractions. In our project, we used n-gram method to calculate the most frequent sequence patterns, and TF-IDF value to identify the most frequent features. The detailed steps are mentioned below:

- The dataset used in this project was obtained from the NIPS (Neural Information Processing Systems) conference, which is an annual machine learning and computational neuroscience conference.
- The dataset was loaded into the Jupyter notebook using the Pandas library in Python.
- To fetch the word count for each abstract, the 'word_count' column was created in the dataset using the 'apply' function in Pandas.
- Descriptive statistics of the word count were obtained using the 'describe' function in Pandas.
- The most common and uncommon words in the dataset were identified using the 'value_counts' function in Pandas.
- The NLTK (Natural Language Toolkit) library was used for text preprocessing, which involves removing stop words, stemming and lemmatization.
- A list of stop words was created using the 'stopwords' corpus in NLTK, and custom stop words were added to it.

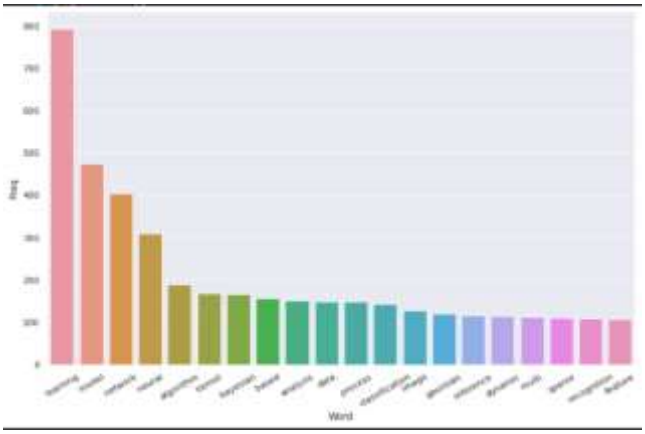
- Punctuations, special characters, digits and HTML tags were removed from the text using regular expressions.
- The text was converted to lowercase, and then to a list of words.
- Stemming and lemmatization were applied to each word in the list using the 'PorterStemmer' and 'WordNetLemmatizer' classes in NLTK.
- The preprocessed text was stored in a list called 'corpus', which was used to generate a word cloud using the 'WordCloud' class in the 'wordcloud' library.
- The most frequently occurring words and bi-grams in the corpus were identified using the 'CountVectorizer' class in Scikit-Learn.
- The top 20 most frequent words and bi-grams were plotted using the 'barplot' function in the 'seaborn' library.
- Finally, we calculated TF-IDF value for individual features, sorted them in increasing order, and finally printed the top 5 features of a document.

V. RESULTS

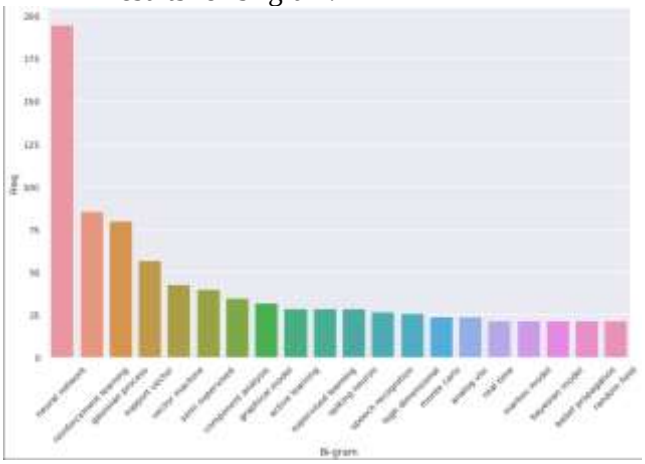
The following section presents the results of the analysis, which includes word cloud and graphs for unigram, bigram, and trigram:

- Word Cloud:

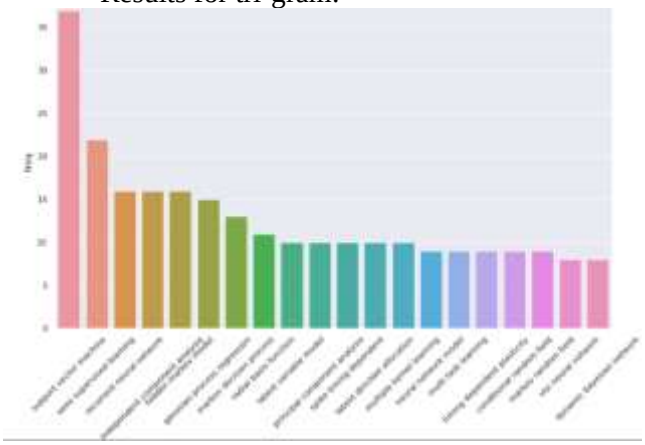
- Results for unigram (top 20 most frequent words):



- Results for bi-gram:



- Results for tri-gram:



VI. CONCLUSIONS

- The dataset used for analysis is NIPS papers dataset.
- The word count for each abstract is fetched and descriptive statistics of word count are obtained.

- The most common and uncommon words are identified from the dataset.
- Text preprocessing techniques such as stemming, lemmatization, removing stop words, and converting to lowercase are applied to the dataset.
- A word cloud is generated to visualize the most common words in the dataset.
- The most frequently occurring words and bi-grams are identified and their frequency is plotted using bar graphs.