Key Phrase Extraction from given phrase

Abstract— **Traditional approaches to extract useful keywords from a sentence rely heavily on human effort. We would like to propose a data driven approach to extract a key phrase efficiently that reduces the scope for human errors and saves time. Machine Learning- Artificial Intelligence algorithm detects the key/message phrase from a sentence that the user feeds as an input and sets a reminder using the key phrase. We have every intention to use different tools such as Natural Language Processing (NLP), Machine Learning (ML), Deep Learning (DL). Automatic key phrase extraction is typically a two-step process: first, a set of words and phrases that could convey the topical content of a document are identified, then these candidates are scored/ranked and the "best" are selected as a document's key phrases. Various tools and algorithms are available to extract keywords or important terms of a given text ,such as NLTK POS tagging, RAKE(Rapid Automatic Key extraction), TextRank .We intend to apply various such methods to find and analyze the best possible way of extracting the keywords**

efficiently. Data set was collected from web and then the aforementioned models were applied to the data set and train them via the same. An accuracy determining function was used on the built and trained models. With slight modifications to the code, the model can be implemented to serve different purposes such as message or threat decoding in military purposes and can be extended to use in speech-to-text purposes and sentimental analysis of the data.

I. INTRODUCTION

Keyphrase extraction is a fundamental task in natural language processing that facilitates mapping of documents to a set of representative phrases. The concise understanding of the text and grasping the central theme behind the given text can be achieved through key phrase extraction. Spending a huge amount of time in reading can be avoided .Information can be extracted efficiently comparing to the traditional extraction techniques. At present times, where there exists a vast amount of information in the form of text on internet, the generation of keywords or phrase has assumed much wider application and importance. With the growing abundance of resource materials on the internet, the need of information retrieval calls for automatic tagging of a text or document to extract relevant information for aparticular query of a user. Without any doubt, the task of manually tagging or summarizing such texts will be herculean; and this calls for automation in this field to reduce the time and effort and of course to meet the unprecedented volume of information to be exchanged today. The rise of 'Big Data Analysis' will play a prominent role in phrase extraction. Any key phrase model aims to generate words and phrases to summarize the given text.

.

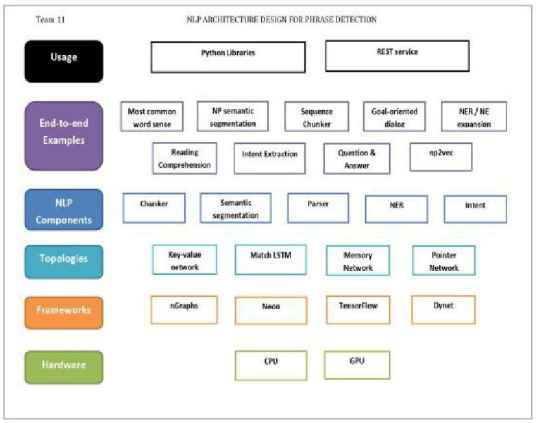
II. VARIOUS APPROACHES TOWARDS KEYPHRASE EXTRACTION

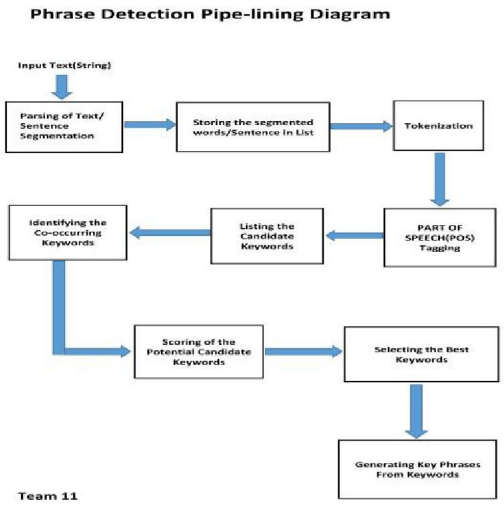
A. Natural Language Processing-NLP

NLP is the widely used technique to extract key phrases from large chunk of data. Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken. NLP is a component of artificial intelligence (AI). Natural language refers to the way we, humans, communicate with each other. Namely, speech and text.

B. Term Frequency-inverse document frequency – TF-IDF

. The tf-idf weight is a weight often used in information retrieval and text mining. Variations of the tf-idf weighting scheme are often used by search engines in scoring and ranking a document's relevance given a query. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus (data-set).





BI. METHODOLOGY IN PHRASE DETECTION

A. NLTK- POS tagging

POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after tokenization process. The dataset has to be pre-processed before adding a tag . The following are the steps to implement POS tagging

- *Parsing of Text/ Sentence Segmentation:*
  Text parsing is a common programming task that splits the given sequence of characters or values(text) into smaller parts based on some rules.
- *Storing the segmented words/Sentence in List:*
  The segmented word is then stored in a list. The sequence is further analyzed ,tokenized and grammar is determined
- *Tokenization:*
  "Tokens" are usually individual words and "tokenization" is taking a text or set of text and breaking it up into its individual words. These tokens are then used as the input for other types of analysis or tasks, like parsing (automatically tagging the syntactic relationship between words).
- *PART OF SPEECH(POS) Tagging:*
  A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'
- *Listing the Candidate Keywords* **:**
  The candidate Keywords are listed based on tags. The co-occurring keywords are identified.
- *Scoring the potential candidate keywords:*
  The potential candidate keywords are scored
  The best keywords are selected and scored .From the given scored words the models generates a phrase.

B. Rapid Automatic Keyword Extraction-RAKE

Candidates are extracted from the text by finding strings of words that do not include phrase delimiters or stop words (a, the, of, etc). This produces the list of candidate keywords/phrases.. A Co-occurrence graph is built to identify the frequency that words are associated together in those phrases. . A score is calculated for each phrase that is the sum of the individual word's scores from the co-occurrence graph. An individual word score is calculated as the degree (number of times it appears + number of additional words it appears with) of a word divided by its frequency (number of times it appears), which weights towards longer phrases.. Adjoining keywords are included if they occur more than twice in the document and score high enough. An adjoining keyword is two keyword phrases with a stop word between them.. The top T keywords are then extracted from the content, where T is 1/3rd of the number of words in the graph..

C. TextRank:

In general, TextRank creates a graph of the words and relationships between them from a document, then identifies the most important vertices of the graph (words) based on importance scores calculated recursively from the entire graph. Candidates are extracted from the text via sentence and then word parsing to produce a list of words to be evaluated. The words are annotated with part of speech tags (noun, verb, etc) to better differentiate syntactic use. Each word is then added to the graph and relationships are added between the word and others in a sliding window around the word. A ranking algorithm is run on each vertex for several iterations, updating all of the word scores based on the related word scores, until the scores stabilize – the research paper notes this is typically 20-30 iterations. The words are sorted and the top N are kept (N is typically 1/3rd of the words). A post-processing step loops back through the initial candidate list and identifies words that appear next to one another and merges the two entries from the scored results into a single multi-word entry.

D. Text Frequency Inverse Document Frequency- TF-IDF:

At a high level, a TF-IDF score finds the words that have the highest ratio of occurring in the current document vs. the frequency of occurring in the larger set of documents.

IV. IMPLEMENTATION

The above proposed was implemented in Python=3.7 and used the NLTK tool kit to preprocess text .RAKE tool is used to produce a list of candidate keywords or phrases and the score calculated for each phrase depending upon features of the word and correlation among them. Adjoining keyword are included if they occur more than twice in the text and given a high score. The top keywords from the contents and displayed to the user . TF-IDF tool is also used to extract the key phrase from the data .

V.CONCLUSION

We implemented all the models for various data text and analyzed the predictions and accuracy. Of all the models ,we infer that RAKE algorithm gives the best results.

REFERENCES

[1] KEA: Practical automatic keyphrase extraction**.** Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G.*In Proceedings of the fourth ACM conference on Digital libraries.* p. 254-255. 1999.

[2] **Improved automatic keyword extraction given more linguistic knowledge.** Anette Hulth. *In Proceedings of EMNLP 2003.* p. 216-223.

[3] **Keyphrase Extraction in Scientific Publications.** Thuy Dung Nguyen and Min-Yen Kan. *In Proceedings of International Conference on Asian Digital Libraries 2007.* p. 317-326.

[4] **Single Document Keyphrase Extraction Using Neighborhood Knowledge.** Xiaojun Wan and Jianguo Xiao. *In Proceedings of AAAI 2008.* pp. 855-860.

[5] **Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods.**Alexander Thorsten Schutz. *Master's thesis, National University of Ireland (2008).*

[6] **Large dataset for keyphrases extraction.** Krapivin, M., Autaeu, A., & Marchese, M. (2009). *University of Trento.*

[7] **Supervised Topical Key Phrase Extraction of News Stories using Crowdsourcing, Light Filtering and Co-reference Normalization.** Marujo, L., Gershman, A., Carbonell, J., Frederking, R., & Neto, J. P. *In Proceedings of LREC 2012.*

[8] **TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction.** Adrien Bougouin, Florian Boudin, Béatrice Daille. *In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), 2013.*

[9] **Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach.** Cornelia Caragea, Florin Bulgarov, Andreea Godea and Sujatha Das Gollapalli. *In Proceedings of EMNLP 2014.* pp. 1435-1446.

[10] **How Document Pre-processing affects Keyphrase Extraction Performance.** Florian Boudin, Hugo Mougard and Damien Cram. *COLING 2016 Workshop on Noisy User-generated Text (WNUT).*

[11] **TermITH-Eval: a French Standard-Based Resource for Keyphrase Extraction Evaluation.** Adrien Bougouin, Sabine Barreaux, Laurent Romary, Florian Boudin and Béatrice Daille. *Language Resources and Evaluation Conference (LREC), 2016.*

[12] **Keyphrase Cloud Generation of Broadcast News.** Luis Marujo, Márcio Viveiros, João Paulo da Silva Neto. *In Proceedings of Interspeech 2011.*

[13] **Human-competitive tagging using automatic keyphrase extraction.** O. Medelyan, E. Frank, I. H. Witten. *In Proceedings of EMNLP 2009.*

[14] **TALN Archives: a digital archive of French research articles in Natural Language Processing.** Florian Boudin. *In Proceedings of TALN 2013.*