**Problem statement:**

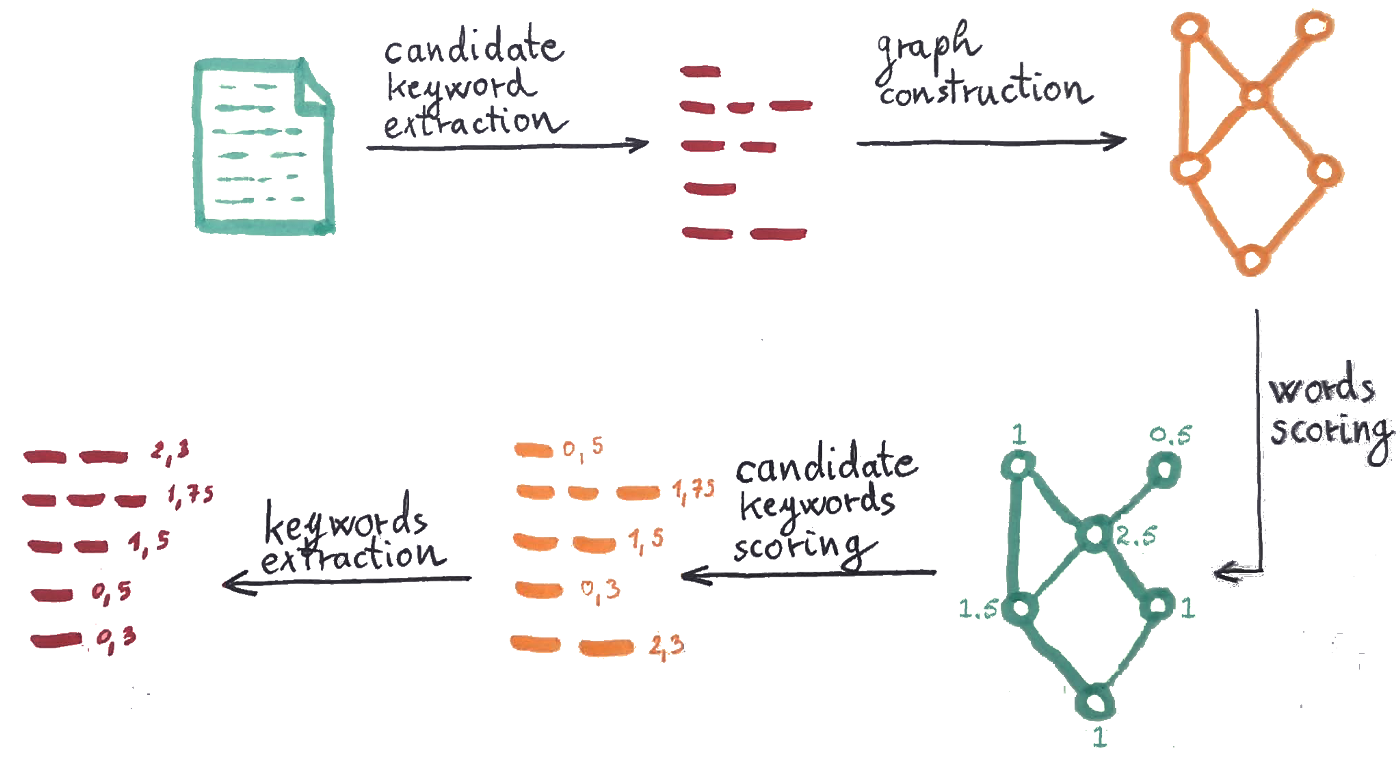
**Given a certain piece of text/paragraph/article you have to develop a topic model for discovering the abstract topics that occur in a collection of documents.**

# **Approach**

Rapid Automatic Keyword Extraction(RAKE) is a Domain-Independent keyword extraction algorithm in Natural Language Processing.

It is an Individual document-oriented dynamic Information retrieval method.

Concept of RAKE is built on three matrices Word Degree (deg(w)), Word Frequency (freq(w)), Ratio of the degree to frequency (deg(w)/freq(w)).

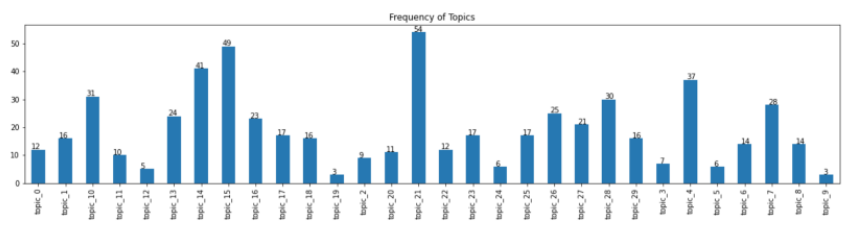


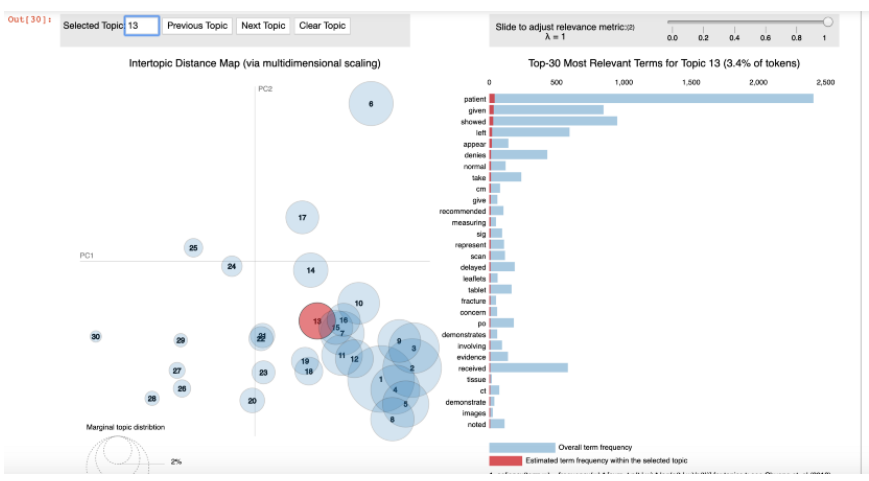
Steps to implement RAKE algorithm..

1. **Candidate keywords extraction** — Text is split on keywords candidates based on stop words and phrase delimiter. A keyword candidate is a phrase that is between two stop-words or phrase delimiters. Phrase delimiters are, for example, punctuation characters.
2. **Keywords co-occurrence graph construction**— Vertices in the graph are words. They are connected if they appear together in candidate keywords. The graph is weighted — weight is the number of times that connected words appear together in the candidate keywords. The graph also includes connections to the vertex itself (each word appears in a candidate keyword with itself).
3. **Words scoring** — Each word in the graph is scored with one of the scores: a) **Word degree** deg(w) — number of words that word w co-occur with (sum of edge weights including an edge that points to vertex itself). Degree favours words that appear more often and in longer keywords. b) **Word frequency** freq(w) — number of times that word appears in any candidate keyword. Frequency favours words that appear more frequent. c) **Ratio of the degree to frequency**deg(w)/freq(w) — This metric favours words that mainly occur in longer candidate keywords. It is suggested to use either word degree or ratio of the degree to frequency. From those two, the degree will favour shorter keywords.
4. **Candidate keywords scoring**— the score of each candidate keyword is the sum of scores of its member words.
5. **Adjoining keywords**— The candidate keywords do not include stopwords. Since sometimes stopwords can be part of the keyword, they are added in this step. The algorithm finds pairs of keywords joined with a stop word in a text and add them to the set of existing stopwords. They must appear at least twice in the text to be added. The score of the new keyword is the sum of its member keywords.
6. **Keywords extraction**— As a result, 1/3 of the best scoring keywords are extracted.

# **Deliverables**

* Sorted data
* Summarized data
* Classified data into abstract topics with sufficient accuracy
* Data analysis and statistics in form of graphs, excel and histogram





# **Technology Stack**

* spaCy
* NumPy
* Pandas
* Natural Language Toolkit (NLTK)
* Matplotlib

For Website (Frontend and Backend)

1. Django
2. SQL
3. HTML/CSS/JS

# **Project Timeline**

22nd Nov to 5th Dec

Learning and implementing tech stack mentioned above.

3rd Jan – 15th Jan

Implementing explored tech stack into the RAKE algorithm.

15th Jan – 30th Jan

Learning required tech stack for website building for tool.

30th Jan – 7th Feb

Compiling all work into one project.

Includes Weekly updates on every Friday.