1: Task description

a. Describe the task: Summarize long text into a clear and complete short text, informative texts. Shorten the reader's reading time. Minimize unnecessary redundant information.

b. Motivation: explain why the task is important

Faced with the trend that people spend more and more time reading emails, online newspapers, and social networks, algorithms that use machine learning to automatically summarize long documents concisely and accurately are becoming increasingly popular. necessary and has a great role in any field.

Automatic summarization will be one of the important technologies that can help people reduce the time they spend reading emails and new information and knowledge to spend time on other tasks, while still being able to grasp them concisely. its contents.

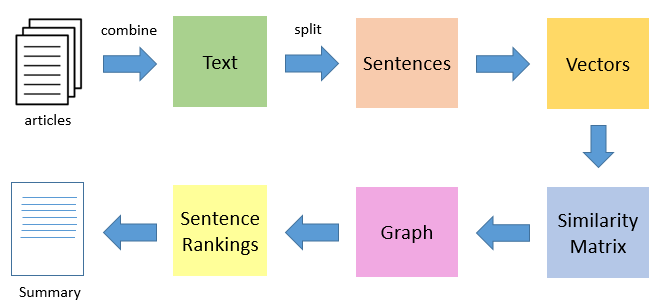
2: Survey

method for the task

Extractive Summarization: These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method.

3: method

a. How did you model the task?



b. How did you solve the task?

* The first step would be to concatenate all the text contained in the articles
* Then split the text into individual sentences
* In the next step, we will find vector representation (word embeddings) for each and every sentence
* Similarities between sentence vectors are then calculated and stored in a matrix
* The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation
* Finally, a certain number of top-ranked sentences form the final summary

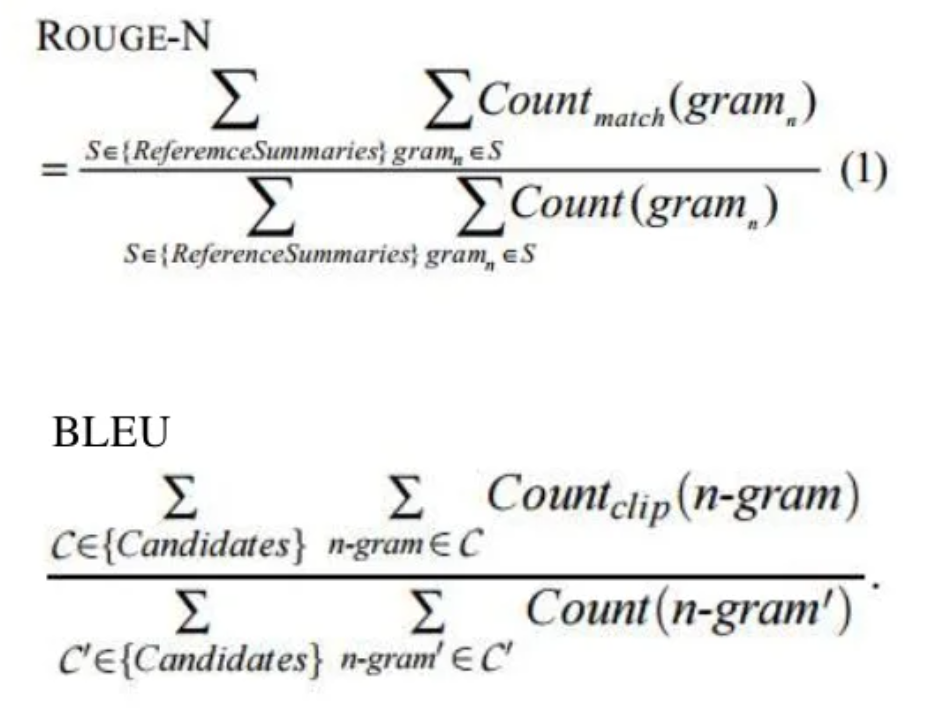
4. Evaluation: how do you evaluate your methods

a. dataset: NLTK Natural Language Toolkit is a set of libraries and programs for English symbolic and statistical natural language processing, built in the Python language.

b. Evaluation measures

**Rouge measures recall**: how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries.

**Bleu measures precision**: how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries.



these results are complementing, as is often the case in precision vs recall. If you have many words/ngrams from the system results appearing in the human references you will have high Bleu, and if you have many words/ngrams from the human references appearing in the system results you will have high Rouge.

Finally, you could use the F1 measure to make the metrics work together: F1 = 2 \* (Bleu \* Rouge) / (Bleu + Rouge)

c. result

Rouge measures recall:

[{'rouge-1': {'r': 0.3392857142857143, 'p': 1.0, 'f': 0.5066666628835556},

'rouge-2': {'r': 0.3125, 'p': 1.0, 'f': 0.4761904725623583},

'rouge-l': {'r': 0.3392857142857143, 'p': 1.0, 'f': 0.5066666628835556}}]

Bleu measures precision: 0.18315341689655779

F1 = 2 \* (Bleu \* Rouge) / (Bleu + Rouge)

d. point out what are solved and not solved; what are challenging problem in the task

what are solved:

* software was able to summarize,
* Clear communication
* concise and minimize redundant information.

what are not solved:

* some information is still lost.
* The evaluation of the model by bleu has many shortcomings.

what are challenging problem in the task

* find data to evaluate the model
* evaluate the model

5. conclusion