Natural Language processing(CS760)

Mini Project

Text Summarization Using NLP

Submitted By: Rishi Anand Arya
M.Tech[DS],SC&SS,JNU

Submitted To:Prof.Piyush Pratap Singh
SC&SS,JNU

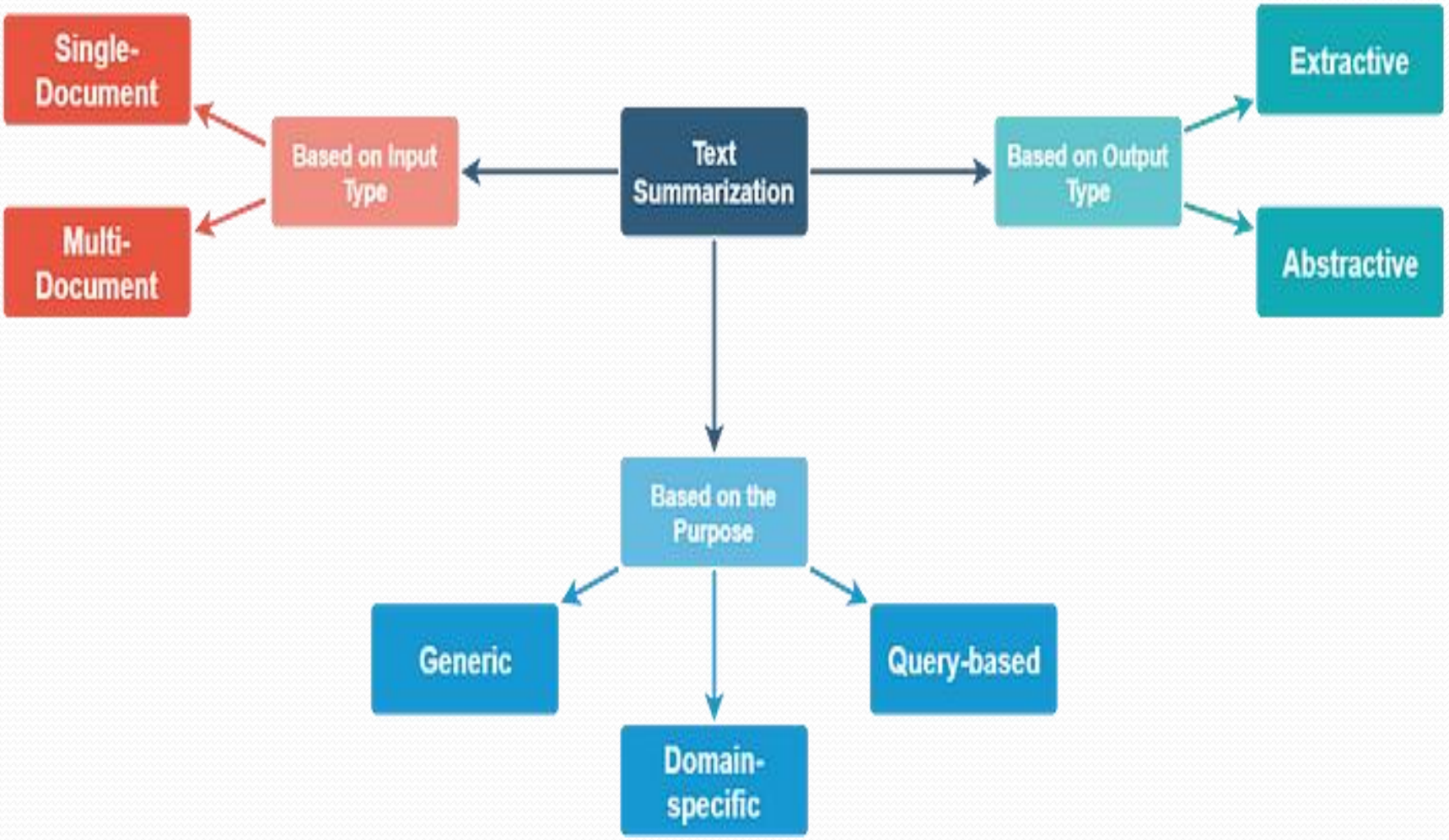
Text Summarization

Text summarization is the process of condensing large piece of text to smaller text retaining the relevant information. The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format.

Why Automated Text summarization?

- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarization.
- Personalized summaries are useful in question-answering systems as they provide personalized information.
- Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of text documents they are able to process.

Types of Text Summarization



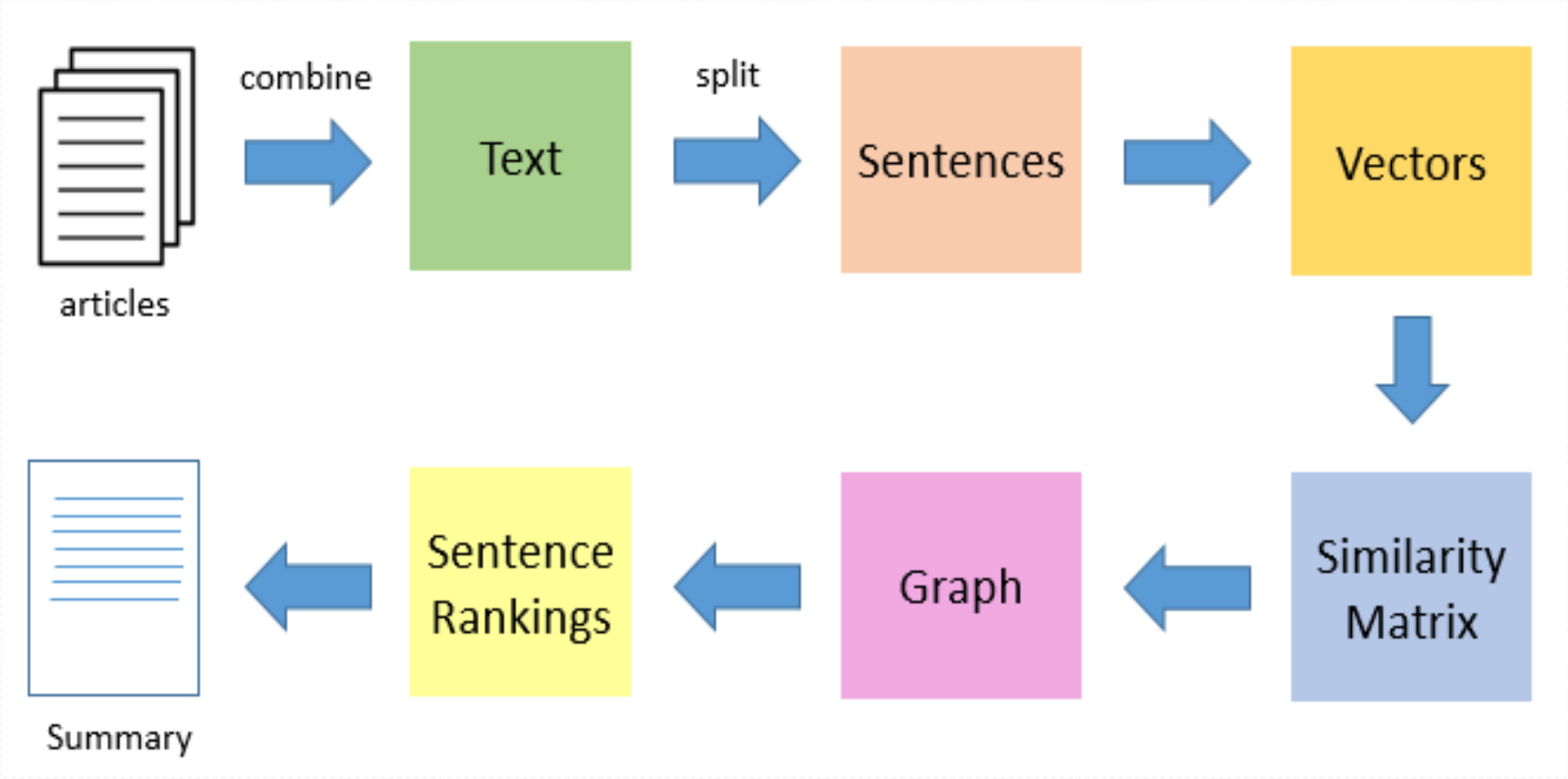
Approaches of Text Summarization

There are two approaches of text summarization :

- Extractive approach : *It aims at extracting and concatenating important span of the source text.This is akin to skimming the text.This approach has been shown to maintain a reasonable degree of grammaticality and accuracy.*

- Abstractive approach: *This approach focuces on generating new summaries that paraphrase the source text.It is more challenging as the model must be able to represent semantic information of the source text to generate a paraphrase.*

Flow chart of Text Summarization



Steps of Text Summarization

- Text Cleaning and Preprocessing
- Sentence tokenization
- Word tokenization
- Word-frequency table
- Summarization

Text cleaning

```
# !pip install -U spacy
# !python -m spacy download en_core_web_sm
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
stopwords = list(STOP_WORDS)
nlp = spacy.load('en_core_web_sm')
doc = nlp(text)
```

Word Tokenization

```
1.  tokens = [token.text for token in doc]
2.  print(tokens)
3.  punctuation = punctuation + '\n'
4.  Punctuation
5.  word_frequencies = {}
6.  for word in doc:
7.  If word.text.lower() not in stopwords:
8.  if word.text.lower() not in punctuation:
9.  if word.text not in word_frequencies.keys():
10. word_frequencies[word.text] = 1
11. else:
12. word_frequencies[word.text] += 1
13. print(word_frequencies)
```

Sentence tokenization

```
1. max_frequency = max(word_frequencies.values())
2. max_frequency
3. for word in word_frequencies.keys():
4. word_frequencies[word] = word_frequencies[word]/max_frequency
5. print(word_frequencies)
6. sentence_tokens = [sent for sent in doc.sents]
7. print(sentence_tokens)
```

Word-Frequency Table

```
1. sentence_scores = {}
2. for sent in sentence_tokens:
3. for word in sent:
4. if word.text.lower() in word_frequencies.keys():
5. if sent not in sentence_scores.keys():
6. sentence_scores[sent] = word_frequencies[word.text.lower()]
7. else:
8. sentence_scores[sent] += word_frequencies[word.text.lower()]
9. sentence_scores
```

Summarization

```
1. from heapq import nlargest
2. select_length = int(len(sentence_tokens)*0.3)
3. select_length
4. summary = nlargest(select_length, sentence_scores, key = sentence_scores.get)
5. summary
6. final_summary = [word.text for word in summary]
7. summary = ' '.join(final_summary)
```

Working of Text Summarization Algorithm

- Text summarization is typically approached as a supervised machine learning issue in NLP.

This is how the approach should be taken:

- Create a method for extracting the important keys from the original document.
- Collect text documents with keywords that are favorably labeled. The keys must be compatible with the extraction method specified. One may also build negatively labeled keys to improve accuracy.
- To produce the text summary, train a binary machine learning classifier. Finally, in the test phrase, generate all of the relevant words and phrases and classify them accordingly.

Machine learning models for Text summarization

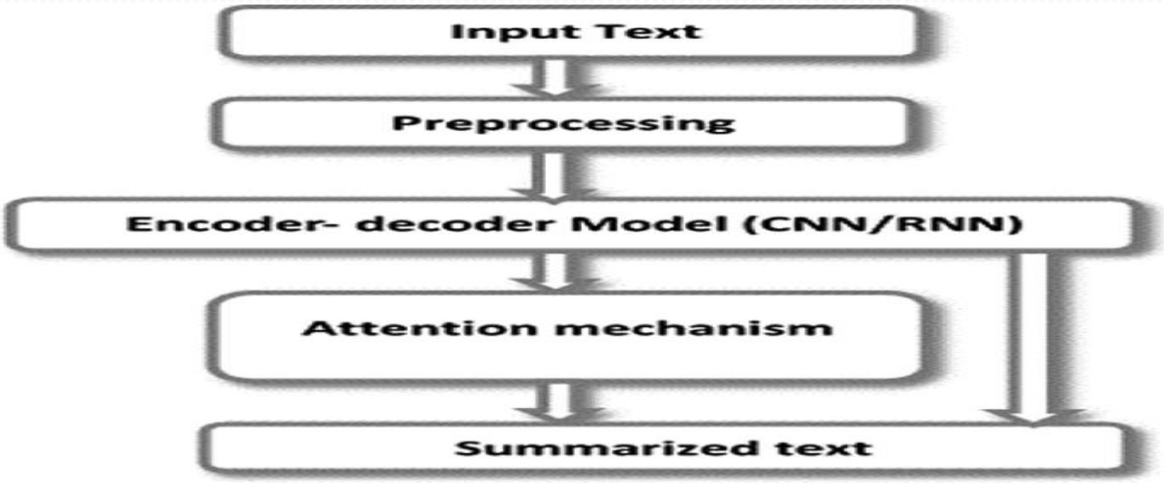
Sequence-to-Sequence modeling(Seq2Seq)

We may use a Seq2Seq model to solve any problem involving sequential data.

Sentiment categorization, Neural Machine Translation, and Named Entity Recognition are examples of popular sequential information applications.

- The input is a text in one language, and the output is also a text in another language in the case of Neural Machine Translation.
- The input for Named Entity Recognition is a list of words, and the output is a list of tags for each of the words in the list.

The two major components of Seq2Seq modeling are Encoder and Decoder.



Challenges to Text summarization

- The main challenges associated with text summarization is topic identification,interpretation,summary generation and evaluation of generated summary.
- Another challenge is multi-document summarization, in which multiple documents are summarized into a single summary.
- The other main challenge with extractive text summarization is that it only chooses the most important words, sentences,and paragraphs to produce a summary. Extractive summarization has a weakness in terms of coherence between sentences in the summary.

Solutions to the challenges

Semantic representation of the corpus of text/documents is

one of the solutions to the challenges we face in text summarization.

- Semantic representation is the process of encoding the content, structure, and context of a text into a formal system, such as a graph, a matrix, or a vector.
- Semantic representation can help you measure the similarity, relevance, and coherence of the summary and the original text.
- It can also help you generate summaries that are more informative, diverse, and concise.
- However, semantic representation is not a trivial task, as it requires a lot of knowledge, resources, and computation. Moreover, semantic representation can be affected by ambiguity, vagueness, or inconsistency in natural language.

List of some Text Summarization models with reference

- A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS).ref.:https://arxiv.org/ftp/arxiv/papers/1805/1805.07799.pdf
- Learning to Extract Coherent Summary via Deep Reinforcement Learning.ref:https://arxiv.org/pdf/1804.07036v1.pdf
- Abstractive Sentence Summarization with Attentive Recurrent Neural Networks.ref.:https://aclanthology.org/N16-1012.pdf
- Selective Encoding for Abstractive Sentence Summarization ref.:https://arxiv.org/pdf/1704.07073v1.pdf
- Sample Efficient Text Summarization Using a Single Pre-Trained Transformer. Ref.:https://arxiv.org/pdf/1905.08836v1.pdf

Thanks