srijan:

RAIT acm Chapter

Approaching Text Summarization using ML and DNN

RAIT-ACM STTP

25 May 2020

About Me

Mayank Kumar Jha

Data Scientist | Kaggle Competition Expert

Experience Across
Machine Learning, Deep Learning,
Data Ops, Cloud, Algorithms, Optimization

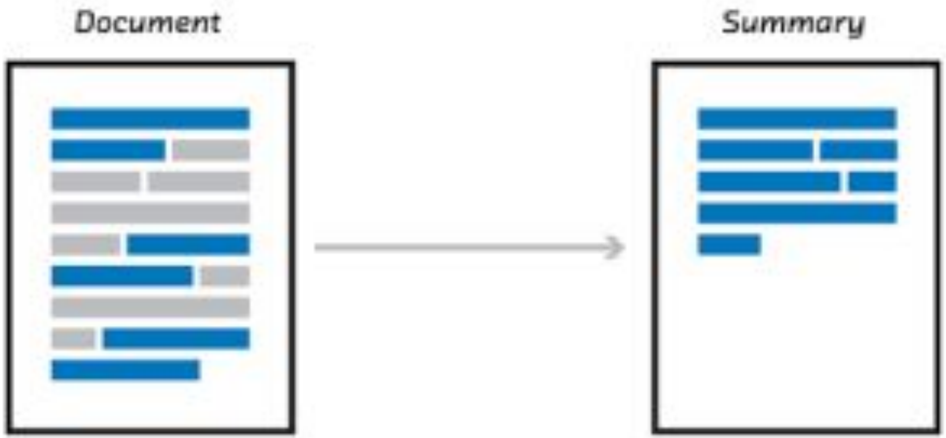
Things to cover

- Introduction to Text Summarization
- Various approaches
- Propose possible solutions
- Create a basic solution
- Code Walkthrough
- Query session

What is Text Summarization?

What is Text Summarization?

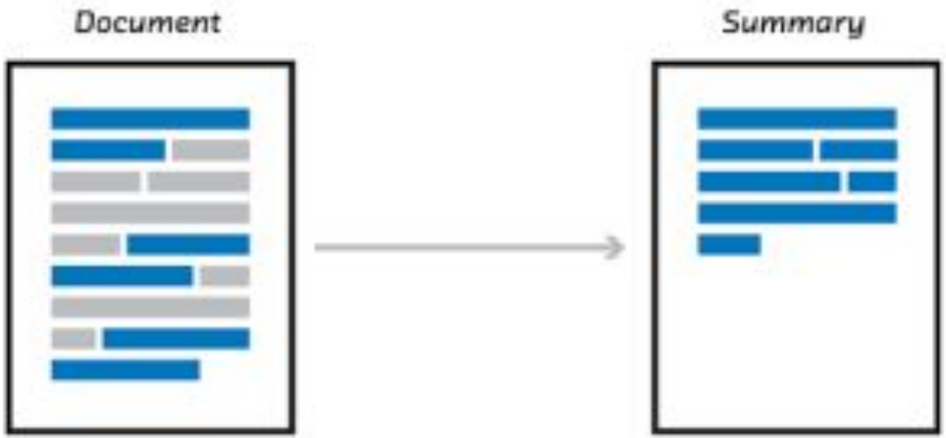
srijan:



ref: https://medium.com/@ondenyi.eric/extractive-text-summarization-techniques-with-sumy-3d3b127a0a32

What is Text Summarization?

- Text summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content.[wikipedia]
- Text summarization is the technique for generating a concise and precise summary of voluminous texts while focusing on the sections that convey useful information, and without losing the overall meaning.[floydhub]



ref: https://medium.com/@ondenyi.eric/extractive-text-summarization-techniques-with-sumy-3d3b127a0a32

Various approaches for text summarization?

- **Extractive Summarization**

- **Abstractive Summarization**

Various approaches for text summarization?

srijan:

- **Extractive Summarization**

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter

- **Abstractive Summarization**

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

ref: https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/

Various approaches for text summarization?

- **Extractive Summarization**
  - Extractive summarization means identifying important sections (paragraphs or sentences or even words) of the text and selecting (copy paste) them producing a subset of the text from the original text.

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter

- **Abstractive Summarization**
  - Abstractive summarization is the technique of generating a summary of a text from its main ideas, not by copying verbatim most salient sentences from text.

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

ref: https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/

Approaching Extractive Summarization?

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
    - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer

Approaching Extractive Summarization?

- **What is TF-IDF Vectorizer:**

Approaching Extractive Summarization?

- **What is TF-IDF Vectorizer:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

Approaching Extractive Summarization?

- **What is TF-IDF Vectorizer:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Count vectorization

	for	sat	around	fire	a	the	man	went	out	walk	children
0	0.142857	0.000000	0.000000	0.000000	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857	0.000000
1	0.000000	0.166667	0.166667	0.166667	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.166667

Approaching Extractive Summarization?

- **What is TF-IDF Vectorizer:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

Count vectorization

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

	for	sat	around	fire	a	the	man	went	out	walk	children
0	0.142857	0.000000	0.000000	0.000000	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857	0.000000
1	0.000000	0.166667	0.166667	0.166667	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.166667

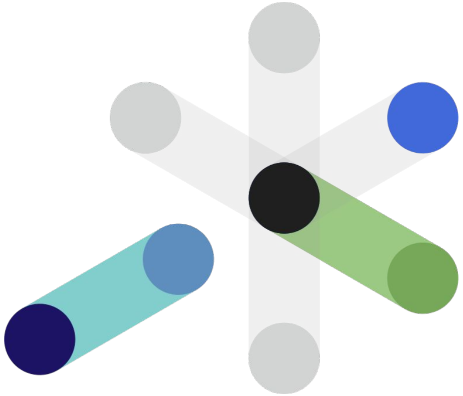
Inverse Document Frequency (IDF)

$$idf(w) = log(\frac{N}{df_t})$$

	for	sat	around	fire	a	the	man	went	out	walk	children
0	0.099021	0.000000	0.000000	0.000000	0.099021	0.0	0.099021	0.099021	0.099021	0.099021	0.000000
1	0.000000	0.115525	0.115525	0.115525	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.115525

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

TF-IDF



Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec

Approaching Extractive Summarization?

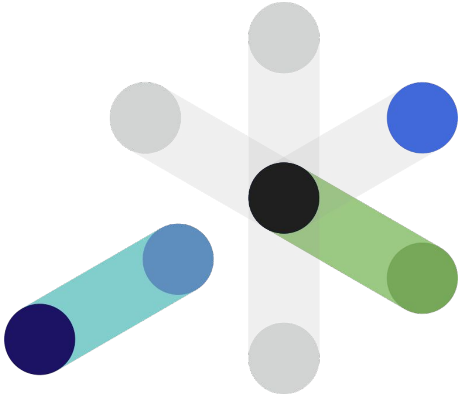
- **What is Word2Vec embedding:**

Approaching Extractive Summarization?

- **What is Word2Vec embedding:**
    - Embedding is something like key-value pair.

Approaching Extractive Summarization?

- **What is Word2Vec embedding:**
  - Embedding is something like key-value pair.
  - For each word in our vocabulary, there will be a learned numerical representation for it.



Approaching Extractive Summarization?

- **What is Word2Vec embedding:**
  - Embedding is something like key-value pair.
  - For each word in our vocabulary, there will be a learned numerical representation for it.
  - To handle unknown words, we will treat each of them as OOV (Out of vocabulary) words and will use the learned OOV representation.

Approaching Extractive Summarization?

- **What is Word2Vec embedding:**
  - Embedding is something like key-value pair.
  - For each word in our vocabulary, there will be a learned numerical representation for it.
  - To handle unknown words, we will treat each of them as OOV (Out of vocabulary) words and will use the learned OOV representation.
  - Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram.[wikipedia]

Approaching Extractive Summarization?

- **What is Word2Vec embedding:**
  - Embedding is something like key-value pair.
  - For each word in our vocabulary, there will be a learned numerical representation for it.
  - To handle unknown words, we will treat each of them as OOV (Out of vocabulary) words and will use the learned OOV representation.
  - Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram.[wikipedia]
  - Both of these techniques learn weights which act as word vector representations.

Approaching Extractive Summarization?

srijan:

- **What is CBOW:**

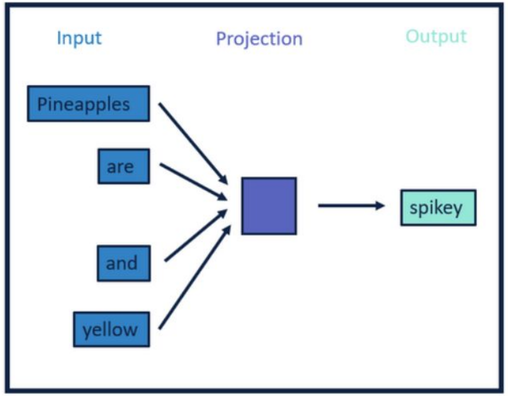
Approaching Extractive Summarization?

srijan:

- **What is CBOW:**
  - CBOW stands for continuous bag-of-words architecture:
    - Task is to predicts the current word from a window of surrounding context words.[wikipedia]
    - The order of context words does not influence prediction (bag-of-words assumption).[wikipedia]

Approaching Extractive Summarization?

- **What is CBOW:**
  - CBOW stands for continuous bag-of-words architecture:
    - Task is to predicts the current word from a window of surrounding context words.[wikipedia]
    - The order of context words does not influence prediction (bag-of-words assumption).[wikipedia]



CBOW

Input	Output
Pineapples	Spikey
are	Spikey
and	Spikey
yellow	Spikey

CBOW

ref: https://community.alteryx.com/t5/Data-Science-Blog/Word2vec-for-the-Alteryx-Community/ba-p/305285

Approaching Extractive Summarization?

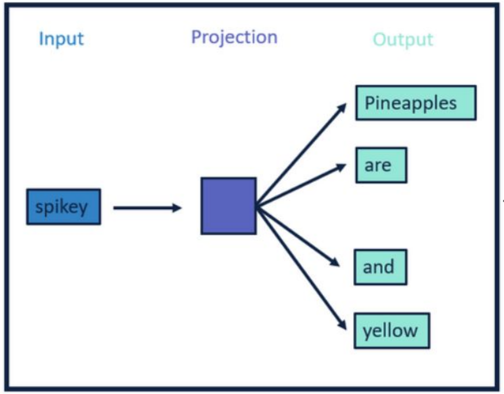
- **What is SKIP-GRAM:**

Approaching Extractive Summarization?

- **What is SKIP-GRAM:**
  - Continuous skip-gram architecture:
    - Task is to use the current word to predict the surrounding window of context words. [wikipedia]
    - The skip-gram architecture weighs nearby context words more heavily than more distant context words.[wikipedia]
    - According to the authors' note, CBOW is faster while skip-gram is slower but does a better job for infrequent words.[wikipedia]

Approaching Extractive Summarization?

- **What is SKIP-GRAM:**
  - Continuous skip-gram architecture:
    - Task is to use the current word to predict the surrounding window of context words. [wikipedia]
    - The skip-gram architecture weighs nearby context words more heavily than more distant context words.[wikipedia]
    - According to the authors' note, CBOW is faster while skip-gram is slower but does a better job for infrequent words.[wikipedia]



Skip-gram

Input	Output
Spikey	Pineapples
Spikey	are
Spikey	and
Spikey	yellow

Skip-gram

ref: https://community.alteryx.com/t5/Data-Science-Blog/Word2vec-for-the-Alteryx-Community/ba-p/305285

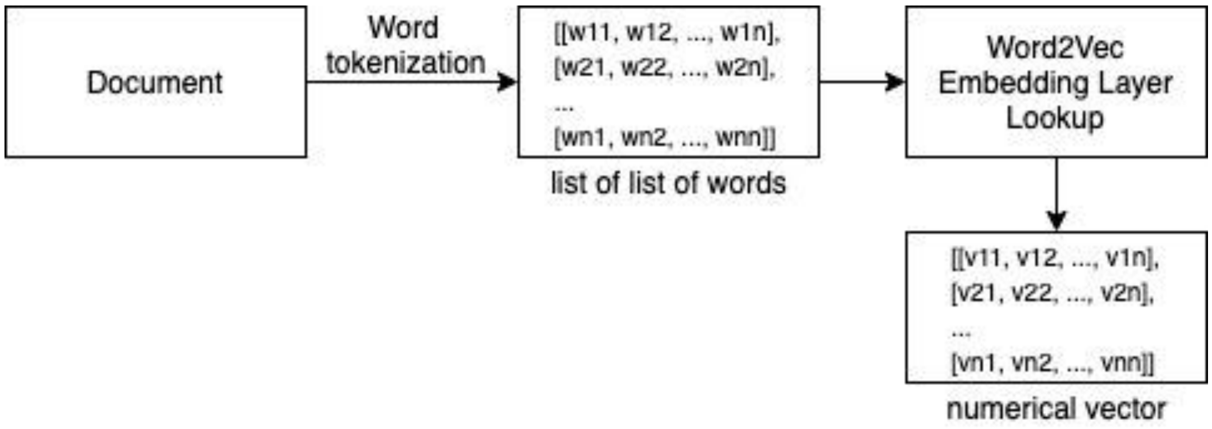
Approaching Extractive Summarization?

- **Using Word2Vec Embedding**

Approaching Extractive Summarization?

srijan:

- **Using Word2Vec Embedding**
  - Flow would be something like below one:



Using Word2Vec to convert text data to numerical semantic vector

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText

Approaching Extractive Summarization?

srijan:

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
  - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.

Approaching Extractive Summarization?

- **What is BERT:**

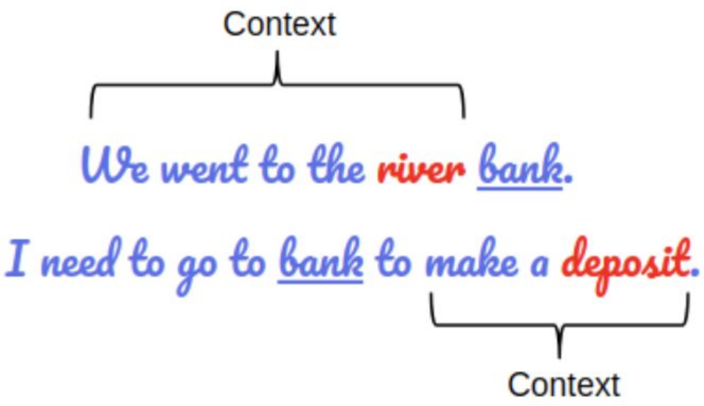
Approaching Extractive Summarization?

- **What is BERT:**
  - BERT stands for Bi-directional Encoder Representation from Transformers.
  - It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context.
  - BERT is pre-trained on two NLP tasks:
    - Masked Language Modeling
    - Next Sentence Prediction

Approaching Extractive Summarization?

srijan:

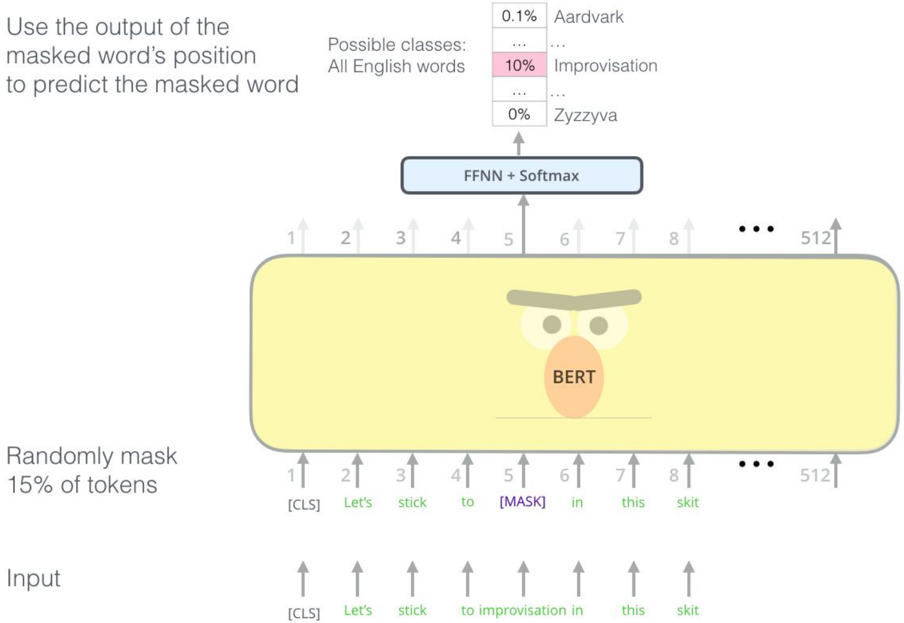
- **What is BERT:**
  - BERT stands for Bi-directional Encoder Representation from Transformers.
  - It has two variants mainly:
    - BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters
    - BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters
  - BERT is pre-trained on two NLP tasks:
    - Masked Language Modeling
    - Next Sentence Prediction

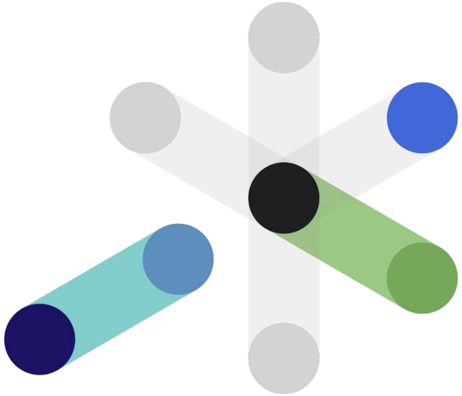


ref: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

Approaching Extractive Summarization?

- **What is Masked Language Modeling:**





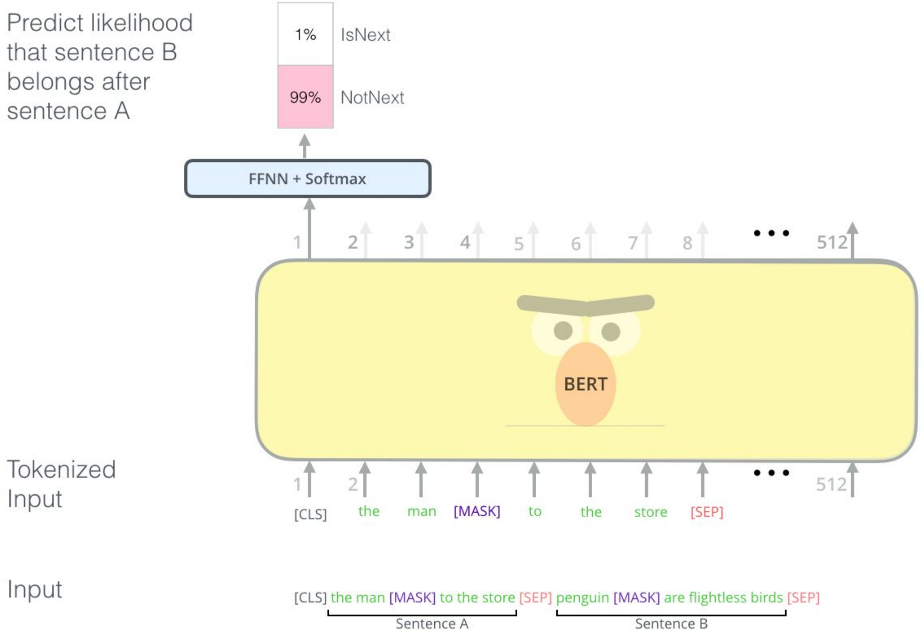
BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word

ref: http://jalammar.github.io/illustrated-bert/

Approaching Extractive Summarization?

srijan:

- **What is Next Sentence Prediction:**



The second task BERT is pre-trained on is a two-sentence classification task. The tokenization is oversimplified in this graphic as BERT actually uses WordPieces as tokens rather than words --- so some words are broken down into smaller chunks.

ref: http://jalammar.github.io/illustrated-bert/

Approaching Extractive Summarization?

- **Using BERT for feature extraction**

Approaching Extractive Summarization?

srijan:

- **Using BERT for feature extraction**
  - Feed the text to BERT Wordpiece Tokenizer

Approaching Extractive Summarization?

srijan:

- **Using BERT for feature extraction**
  - Feed the text to BERT Wordpiece Tokenizer
  - BERT tokenizer will split the sentence as well as words incase needed to match vocabulary.

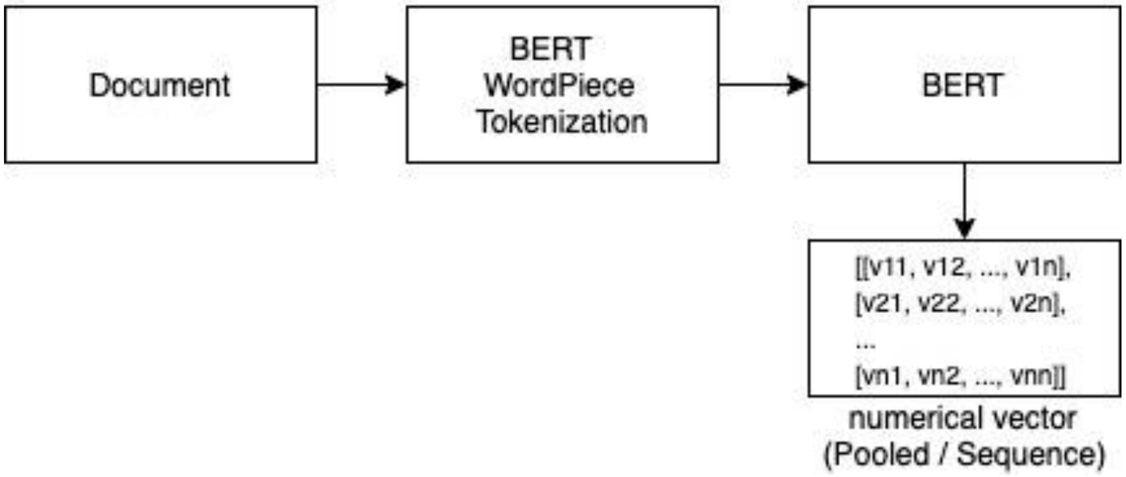
Approaching Extractive Summarization?

- **Using BERT for feature extraction**
  - Feed the text to BERT Wordpiece Tokenizer
  - BERT tokenizer will split the sentence as well as words incase needed to match vocabulary.
  - Words are splitted on the basis of their probability of occurrences in that context which gives BERT advantages to handle contraction as well as spelling errors

Approaching Extractive Summarization?

srijan:

- **Using BERT for feature extraction**
  - Feed the text to BERT Wordpiece Tokenizer
  - BERT tokenizer will split the sentence as well as words incase needed to match vocabulary.
  - Words are splitted on the basis of their probability of occurrences in that context which gives BERT advantages to handle contraction as well as spelling errors
  - Flow would be something like below one:



Using BERT to convert text data to numerical vector

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
  - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
    - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
    - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
    - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.
    - Train your own (quite cumbersome :( )

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
  - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.
  - Train your own (quite cumbersome :( )
- Now comparison can be done between each sentences as they are no more a sequence of characters and words but a numerical vector now.

Approaching Extractive Summarization?

srijan:

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
  - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.
  - Train your own (quite cumbersome :( )
- Now comparison can be done between each sentences as they are no more a sequence of characters and words but a numerical vector now.
- Use the comparisons to score each sentences and pick the top scored ones as your summary.

Approaching Extractive Summarization?

- One approach could be to create a semantic representation of sentences.
- Following can be used to create a semantic representation for texts:
  - Count-based techniques like CountVectorizer, Tf-Idf Vectorizer
  - Pretrained word embeddings based techniques like Word2Vec, Glove, fastText
  - Pretrained SOTA transformers like BERT (or its variants) to better capture context as well.
  - Train your own (quite cumbersome :( )
- Now comparison can be done between each sentences as they are no more a sequence of characters and words but a numerical vector now.
- Use the comparisons to score each sentences and pick the top scored ones as your summary.
- Scoring technique needs to be intelligent enough to properly evaluate what are the information content of a sentence and thus score it accordingly.

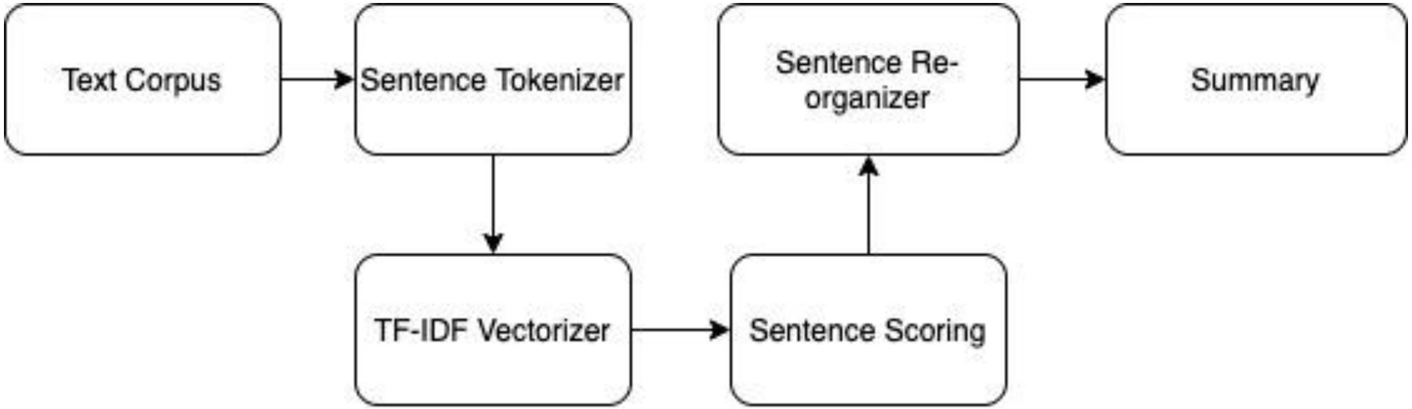
Approaching Extractive Summarization?

srijan:

Let’s build our basic Solution

Approaching Extractive Summarization?

srijan:



A high level solution for frequency based extractive summarizer

Approaching Extractive Summarization?

- **Sentence scoring using TF-IDF:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

Approaching Extractive Summarization?

srijan:

- **Sentence scoring using TF-IDF:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

	for	sat	around	fire	a	the	man	went	out	walk	children
0	0.099021	0.000000	0.000000	0.000000	0.099021	0.0	0.099021	0.099021	0.099021	0.099021	0.000000
1	0.000000	0.115525	0.115525	0.115525	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.115525

Approaching Extractive Summarization?

srijan:

- **Sentence scoring using TF-IDF:**

```
documentA = 'the man went out for a walk'
documentB = 'the children sat around the fire'
```

TF-IDF

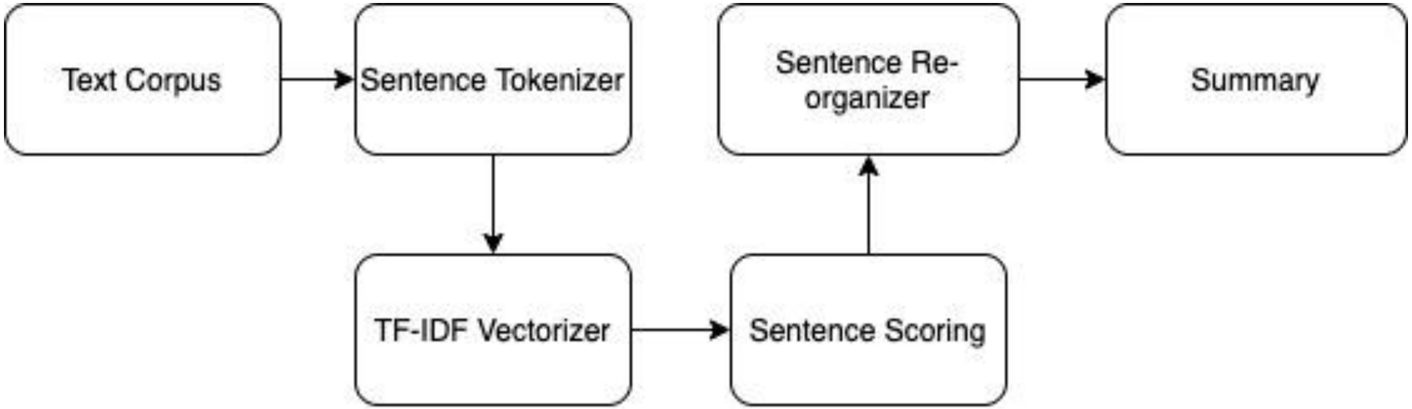
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

	for	sat	around	fire	a	the	man	went	out	walk	children
0	0.099021	0.000000	0.000000	0.000000	0.099021	0.0	0.099021	0.099021	0.099021	0.099021	0.000000
1	0.000000	0.115525	0.115525	0.115525	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.115525

Adding TF-IDF weights to score each sentence

	for	sat	around	fire	a	the	man	went	out	walk	children	score
0	0.099021	0.000000	0.000000	0.000000	0.099021	0.0	0.099021	0.099021	0.099021	0.099021	0.000000	0.594126
1	0.000000	0.115525	0.115525	0.115525	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.115525	0.462098

Approaching Extractive Summarization?



A high level solution for frequency based extractive summarizer

Approaching Extractive Summarization?

- **What is Sentence Reorganizer:**

  Text : Stop Words are words which do not contain important significance to be used in Search Queries. Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information. Each programming language will give its own list of stop words to use.

Approaching Extractive Summarization?

- ## What is Sentence Reorganizer:

Text : Stop Words are words which do not contain important significance to be used in Search Queries. Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information. Each programming language will give its own list of stop words to use.

↓

Sentence Order Indexing : { "Stop Words are words which do not contain important significance to be used in Search Queries": 0,
"Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information": 1 ,
"Each programming language will give its own list of stop words to use": 2 }

Approaching Extractive Summarization?

- **What is Sentence Reorganizer:**

Sentence Order Indexing : { “Stop Words are words which do not contain important significance to be used in Search Queries”: 0,
“Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information”: 1 ,
“Each programming language will give its own list of stop words to use”: 2 }

Scored Sentences : { “Each programming language will give its own list of stop words to use”: 1.32
,
“Stop Words are words which do not contain important significance to be used in Search Queries”: 0.79,
“Usually, these words are filtered out from search queries because they return a vast amount of unnecessary information”: 0.32 }

Result : Stop Words are words which do not contain important significance to be used in Search Queries. Each programming language will give its own list of stop words to use.

Let’s be practical…….

Fire up your Notebooks

srijan:

Any Questions ?

Thank You

srijan:

RAIT acm Chapter