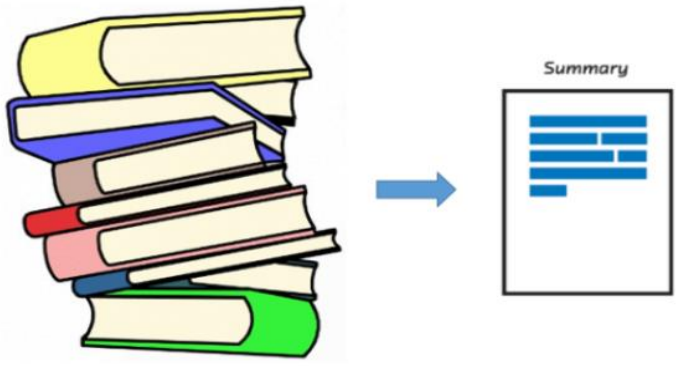
Abstractive summarization



Mandate-I

- ***Problem Statement:***

Given a set of tweets pertaining to a trending topic, create an abstractive prose summary of the tweets. Do not just string the tweets together to form the summary. The summary will need to paraphrase and/or say more than what is directly said in the tweets. Propose a rubric to evaluate the accuracy of your summarization.

- ***Introduction:***

Whether it's sharing breaking news, posting updates about their company or following their favourite celebrities, people are using Twitter to connect with others and to discover new things every day. The data on twitter is the type of ***single domain multi-document*** **text data i.e., several tweets collectively represent the opinion on a single topic in trend.** But size of these document corpuses are too huge for an individual to read. With the overload of information available today I always wished for a way I can read just the summary instead of going through all the tweets**. There are two types of summarizations: extractive and abstractive. Extractive summarization selects a subset of sentences from the text to form a summary; abstractive summarization reorganizes the language in the text and adds novel words/phrases into the summary if necessary.** So, in this project I introduce a novel approach to generate abstractive summaries of a set of tweets pertaining to a trending topic using Natural Language Processing.

- ***Problem Formulation:***

We formalize the issue as follows. Given an event e, a tweet stream T[e,t] concerning e between 2 timestamps t0 and t, our aim is to build a summary S[e,t0,t] of T[e,t], such that S[e,t0,t] = f(T[e,t]). As T[e,t] may represent a huge volume of information (several million of tweets), it does not seem reasonable to start T[e,t+1] from scratch, i.e., by considering the whole stream T[e,t+1]. As a consequence, the problem can be considered as an incremental one as follows: S[e,t0,t+1] = S[e,t0,t] ∪ S[e,t,t+1]

- ***Challenges:***

Twitter summarization can be challenging due to the following reasons:

- **Informal language.**
- **Noise and irrelevant information**.
- Twitter is a platform where people express their opinions, therefore, **sentiment analysis** is crucial for understanding the overall sentiment of tweets, but this can be challenging because of the use of sarcasm, irony, and other nuances in language.
- Twitter also allows users to include **emojis, pictures and videos** in their tweets, which can make it difficult to extract information from them.
- Twitter is a real-time platform, tweets can contain **time-sensitive information**, which can make it challenging to extract the most relevant information from tweets as the topic and sentiment can change quickly.

- ***Basics of Linguistics:***

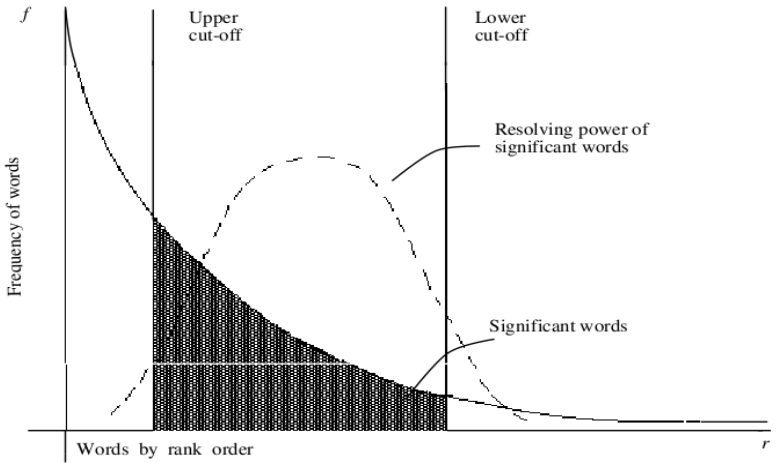
While generating summary we have to keep few points in mind:

- In languages like English, **word orders** are more restricted, and appear mostly in the following form (active voice):

  Subject – Verb(Predicate) --Object

  Shyam complimented Soora

- **Zipf Distribution:** Word distribution in a document corpus found to be very skewed – approximated by a Zipf distribution which says **the frequency of any word is inversely proportional to its rank in the frequency table .**



Most highly occurring terms are “language builders” (terms like is, an, the, for, etc. in English). Such terms having no relevance to document content are called **“stop words”** and are typically removed from consideration. As well as too rarely occurring terms are less significant to the document. Word significance computed by a lower and upper cutoff (typically first and fourth quartile) over the rank ordering.

- A number of linguistic processing require the use of “common sense” reasoning. A common class of coreference resolution problems that require common sense understanding, are the so-called “**Winograd schemas**”.

Example: The pizza was warmer than the hot dog because **it** was in the oven for a ***longer*** amount of time.

→ Here ‘**it**’ refers to the ‘**pizza**’.

The pizza was warmer than the hot dog because **it** was in the oven for a ***shorter*** amount of time.

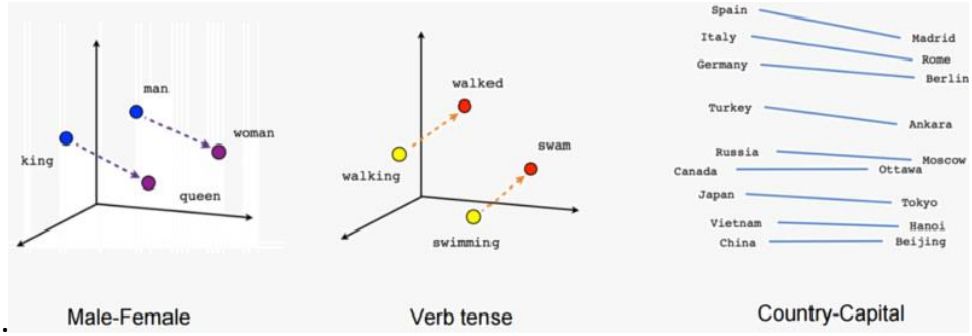
→ Here ‘**it**’ refers to the ‘**hot dog**’.

- ***Approach:***
- **Collection of a large set of tweets on trending topics and their summaries (Document Corpus)**
- **Text Preprocessing:**

1. **Tokenization**: Breaking the text into individual words or phrases.
2. **Lowercasing**: Converting all the text to lowercase to reduce the dimensionality of the problem.
3. **Stop word removal**: Removing common words such as "the", "is", "and", etc., which do not add much meaning to the text.
4. **Stemming or Lemmatization**: Reducing words to their root form to reduce the dimensionality of the problem.
5. Removing special characters and numbers, removing punctuation marks such as ".", ",", "!", etc., removing HTML tags and removing extra whitespaces.

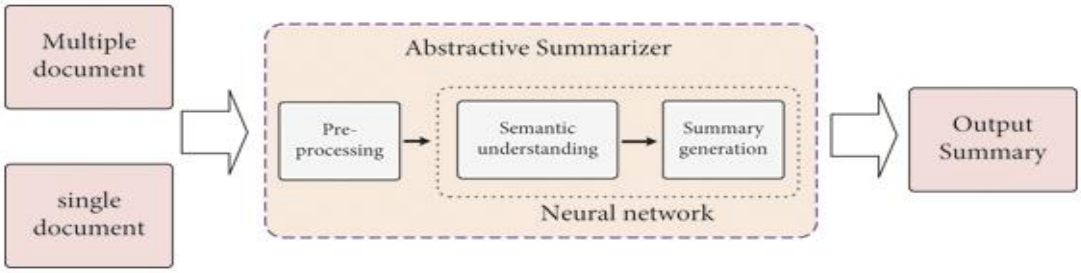
These preprocessing steps help to clean and standardize the text data, making it easier to extract useful information and generate a summary.

- **<u>Statistical NLP and Neural NLP are two approaches to natural language processing (NLP) that are based on different underlying techniques. Both of them can be used for abstract text summarization.</u>**
- **Statistical NLP-based approaches** for abstract text summarization typically involve analyzing the frequency and co-occurrence of words and phrases in the text (**Distributional Hypothesis: "Words with similar meanings tend to appear in similar contexts"**). These models are typically based on assigning probabilities to sequence of n-words(**n-gram**) & linguistic constructs derived from differences between the joint probability P(w1, w2, ...wn) of a sequence as against the product of their marginal probabilities: P(w1)P(w2)...P(wn). **if P(w1,w2,..wn)>P(w1)P(w2)..P(wn)** then **n-gram** can be used. Then those n-grams are represented as vectors in higher dimension using various methods like **bag of words, word2vec, tf-idf** etc. such that semantically similar words will have a higher cosine similarity. The higher the dimension of the vector is the more complex relations it can learn.



But higher dimension introduces **'curse of dimensionality'**.

- **Neural NLP-based approaches** use neural networks to learn the underlying structure of the text and generate a summary. Moreover, Neural NLP is able to handle larger and more complex datasets, which enables them to learn more robust and accurate representations of language. These approaches usually use a combination of **unsupervised and supervised learning techniques, such as sequence-to-sequence (Seq2Seq) models, attention mechanisms, and transformer-based models,** to generate a new condensed version of the input text.
- In summary, both statistical NLP and neural NLP can be used for abstract text summarization, but neural NLP-based approaches tend to be more sophisticated and accurate as they use **representation learning eliminating the task of manual feature engineering**. Additionally, neural NLP models are able to handle more diverse and varied inputs, and can better handle tasks that involve understanding context and relationships between words. **Finally, neural NLP models are generally easier to train and fine-tune for specific tasks, which makes them more widely applicable.**



- ***<u>Evaluation Metrics:</u>***

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

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