

WikiBot

CS6120: Natural Language Processing

Group 30

Ayush Patel

Shivansh Verma

Spandan Maaheshwari

Table of Contents

01

Data Mining

Scrapped data from web for
Wikipedia Level -4 articles

02

Data Pre-processing

Cleaned the data using
various NLP techniques

03

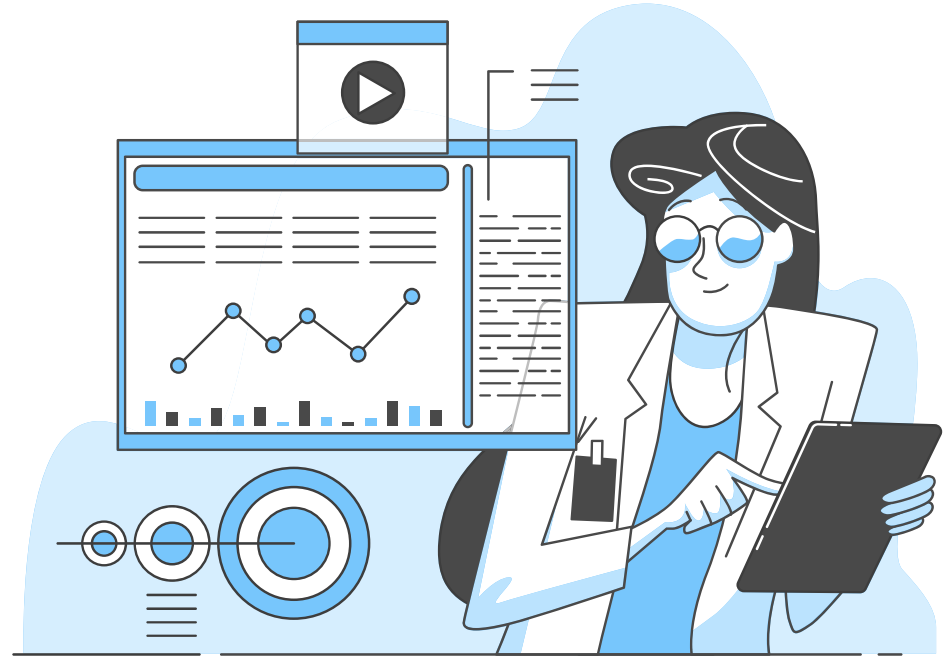
Summarization

Models such as TextRank,
LDA, NMF and Bart

04

Recommendation

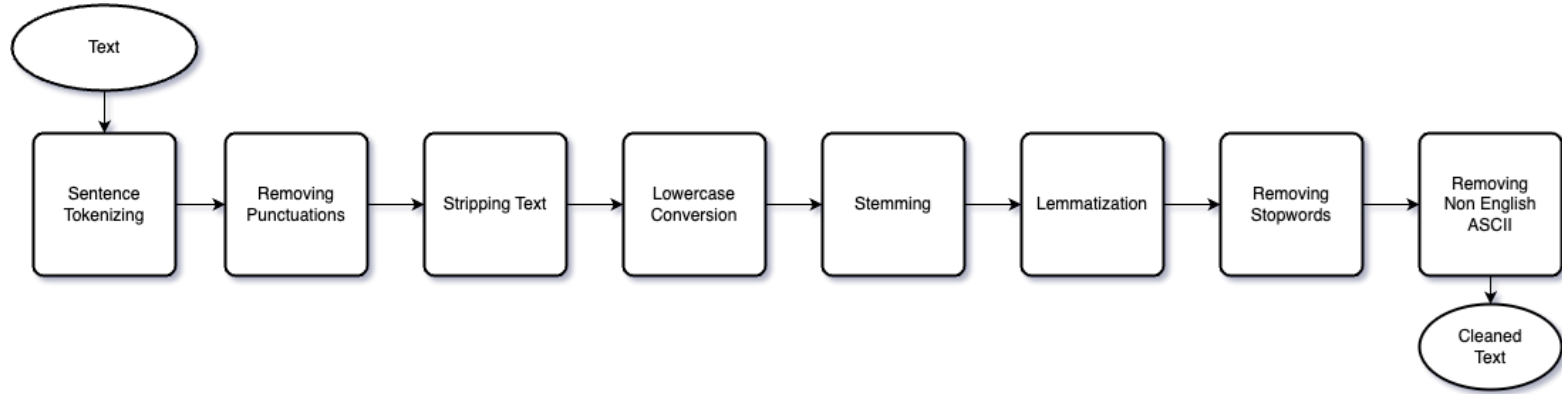
Built a recommendation
system which
recommends users articles
based on their topic of
interest



Data Mining

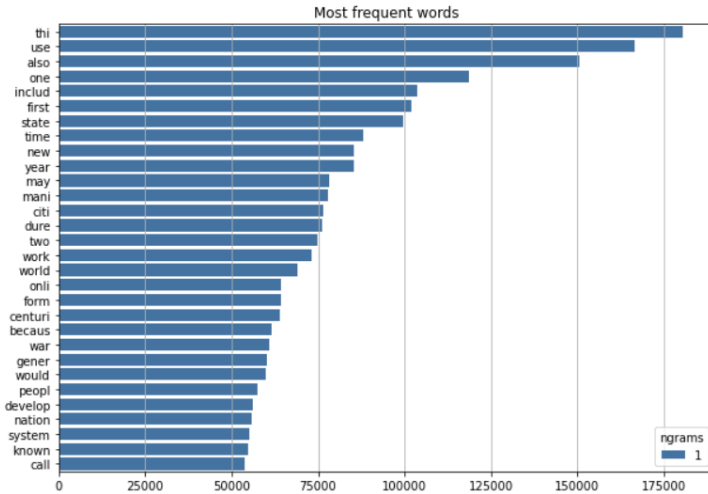
- The Wikipedia Vital Level 4 dataset is a curated collection of over 10,000 articles that cover essential topics in human knowledge. These articles meet certain criteria, such as being notable, widely recognized, and having a significant impact on human knowledge.
- By leveraging BeautifulSoup4 we crawled the Vital Level 4 articles homepage at https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/4
- Then we found all the title headers for all the articles in each of the 11 subsections by identifying correct articles titles using complex BS4 logic and simultaneously removing all the pages that contained noise and improper articles.
- Finally after getting all the article title headers from the webpage we would then feed the article titles to the Python Wikipedia-api to fetch the articles and the summary. Since the API endpoints are rate-limited we have implemented appropriate timeouts in the API calls to avoid requests getting blocked by the endpoint. We are using this API as it is already load balanced for research purposes.

Data Cleaning

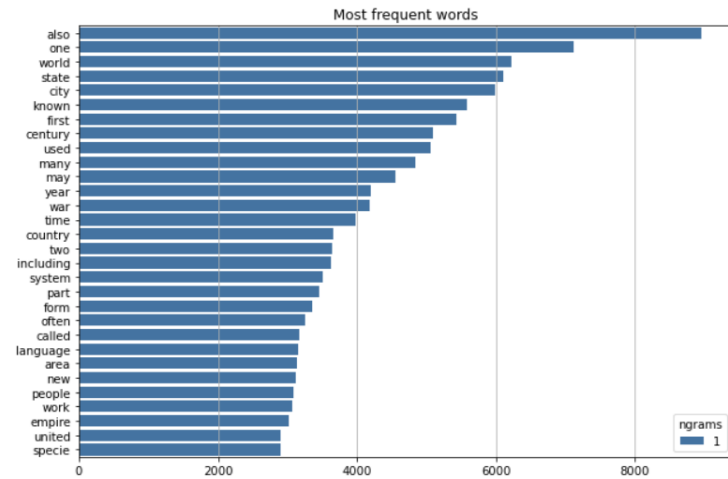


Data Analysis

This is a list of top most frequent words



Top words in article for unigram



Top words in summary for unigram

Data Analysis

	min	mean	max
char_count	421.000000	19482.671296	126076.000000
word_count	60.000000	3371.737236	21625.000000
sentence_count	1.000000	1.000000	1.000000
avg_word_length	4.719607	5.786694	8.22809
avg_sentence_lenght	60.000000	3371.737236	21625.000000

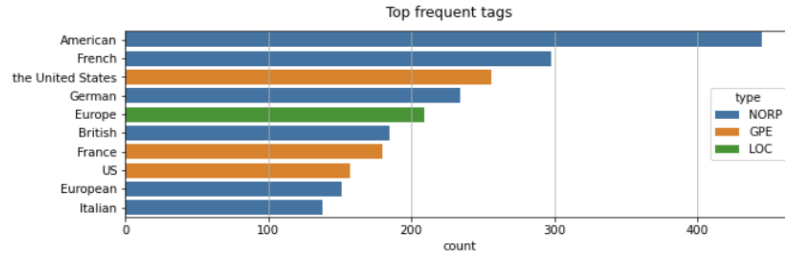


Statistics for word and sentence counts

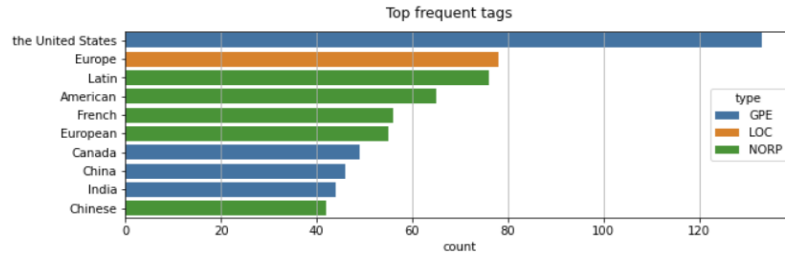
Data Analysis

This is a list of top most frequent words

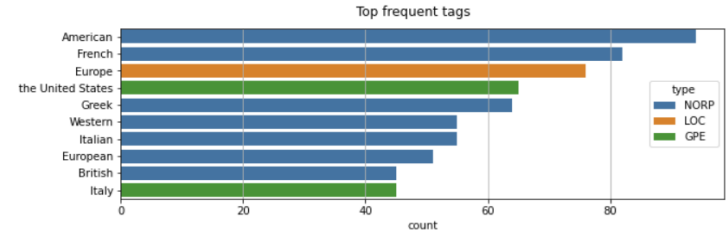
People:



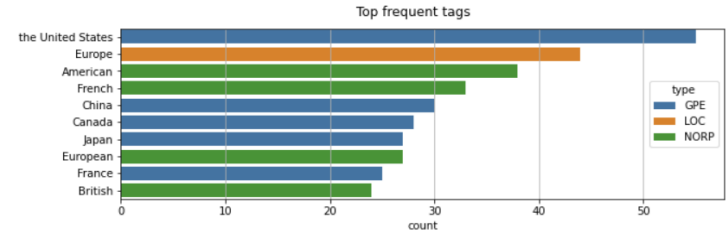
Society_and_social_sciences:



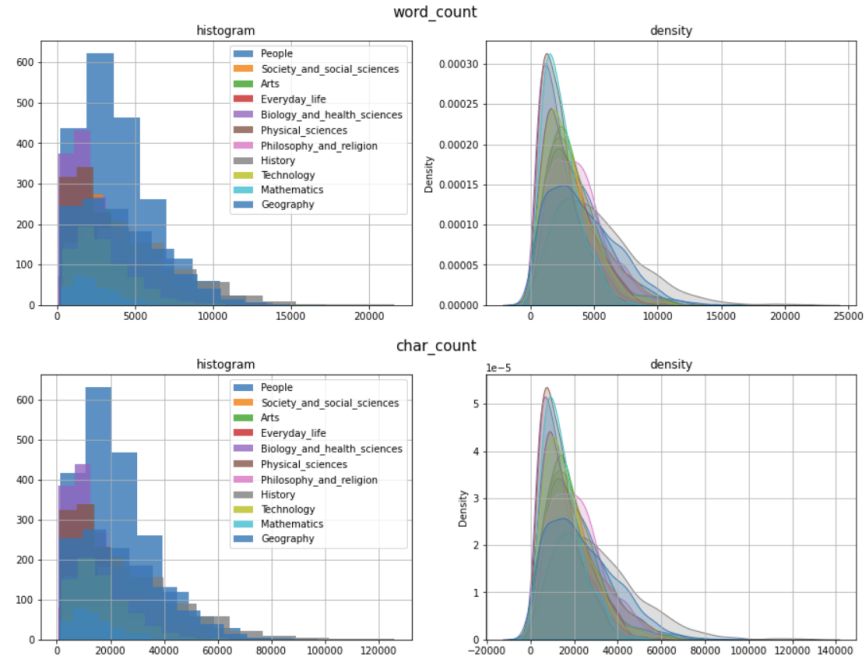
Arts:



Everyday_life:

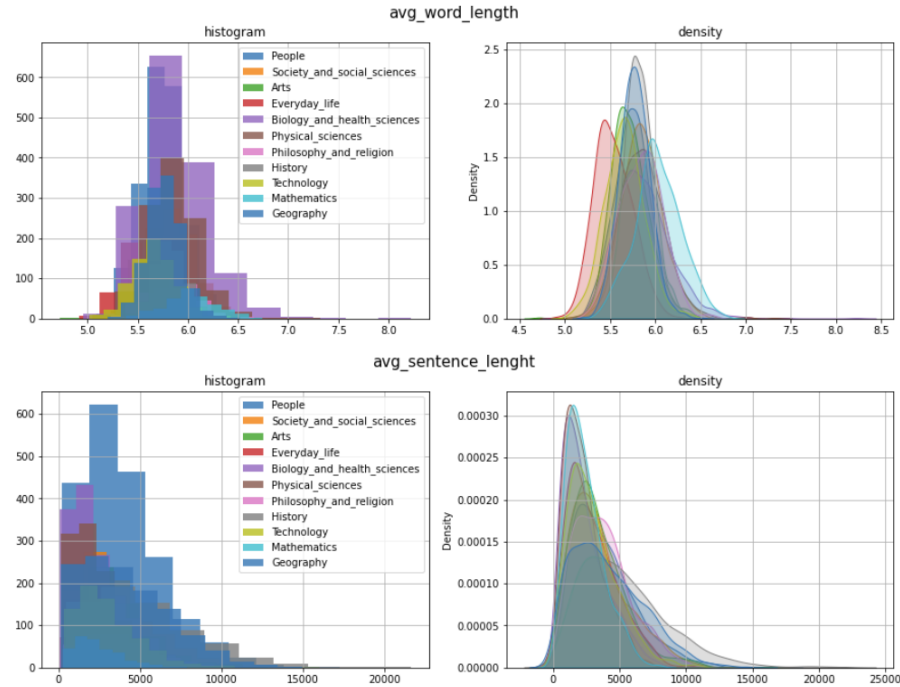


Data Analysis



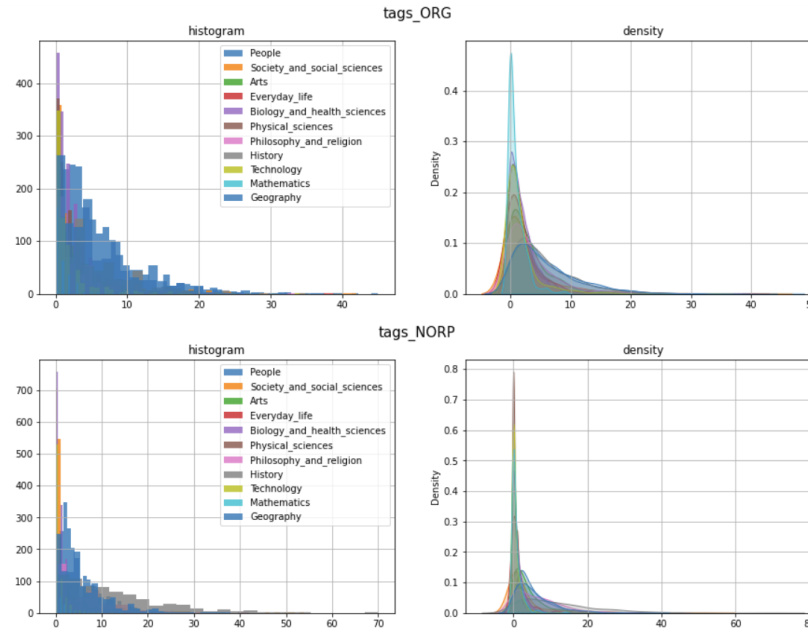
Density Distribution for words and characters for all categories

Data Analysis



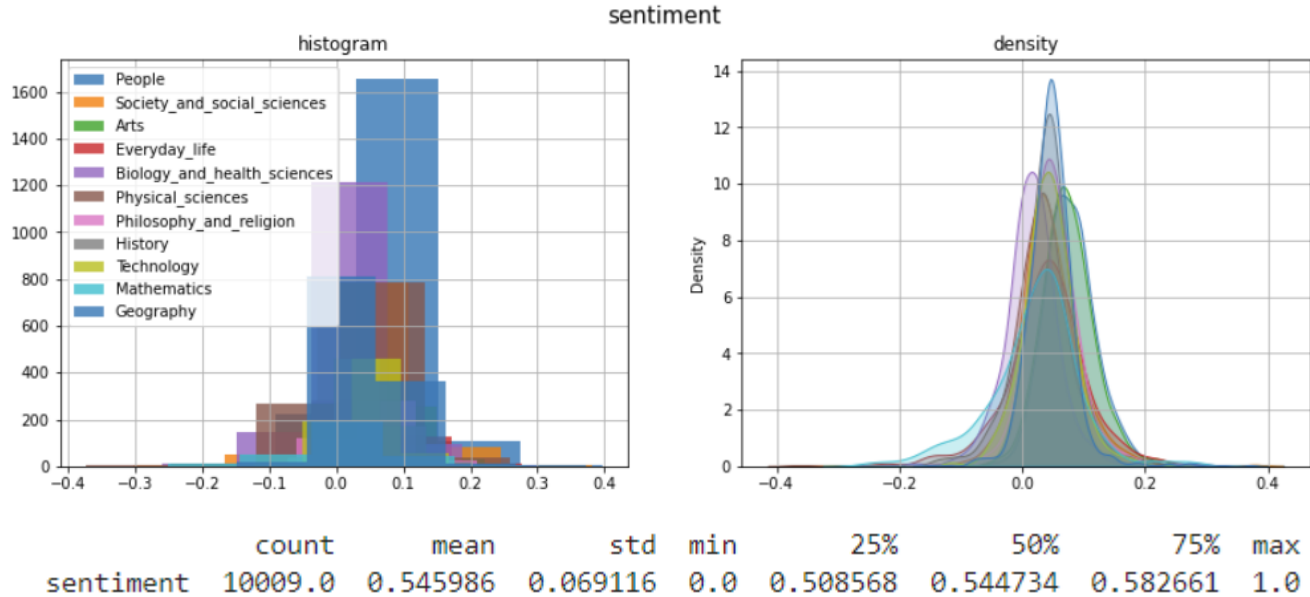
Density Distribution for average word length and average sentence length for all categories

Data Analysis



Density distribution of Organization and Nationalities or religious or political groups for all categories

Data Analysis



Density Distribution of sentiment of articles. Sentiment ranges from -1 to 1 (-1 being negative 1 being positive)

Data Analysis

features selection: from 100,000 to 9,739

Arts:

- . selected features: 739
- . top features: art, film, music, paint, artist, instrument, genre, dance, novel, style

Biology_and_health_sciences:

- . selected features: 1515
- . top features: cell, genus, plant, species, genus, protein, tissue, flower, infection, diseases

Everyday_life:

- . selected features: 629
- . top features: ball, game, player, sport, olympic, ski, saucer, bowl, tournament, sexual

Geography:

- . selected features: 1098
- . top features: city, island, lake, river, sea, airport, park, district, mountain, strait

History:

- . selected features: 862
- . top features: empiricism, war, army, dynasty, ottoman, kingdom, defeat, byzantine, troop, alliance

Mathematics:

- . selected features: 1331
- . top features: algebra, algorithm, axiom, calculus, complex number, \displaystyle , \displaystyle \displaystyle , \displaystyle \mathbb{b} , equate , euclidean

People:

- . selected features: 394
- . top features: career, father, work, award, marriage, son, film, wrote, daughter, friend

Philosophy_and_religion:

- . selected features: 884
- . top features: belief, church, deity, god, myth, religion, philosophy, worship, goddess, judaism

Physical_sciences:

- . selected features: 1435
- . top features: atom, energy, galaxy, hydrogen, magnet, orbit, oxide, particle, star, earth

Society_and_social_sciences:

- . selected features: 582
- . top features: consonant, language, vowel, verb, social, dialect, noun, law, phoneme, linguist

Named Entity Recognition

- Named Entity Recognition involves identifying and categorizing named entities in text into predefined categories such as person names, organizations, locations, medical codes, etc.
- NER has various applications in different fields such as information extraction, sentiment analysis, question answering, and summarization. For example, in our project, we used NER to highlight important entities in Wikipedia articles which could be helpful in generating summaries.

Example

Masaccio **PERSON** (**UK** **GPE** :: **US** **GPE** :: **Italian** **NORP** :: [**maːzattʃo** **PERSON**]; December 21, 1401 – summer 1428), born Tommaso di **PERSON** Ser Giovanni di Simone, was a **Florentine** **NORP** artist who is regarded as the first great **Italian** **NORP** painter of the **Quattrocento** **PRODUCT** period of **the Italian Renaissance** **ORG**. According to **Vasari** **PERSON**, Masaccio **PERSON** was the best painter of his generation because of his skill at imitating nature, recreating lifelike figures and movements as well as a convincing sense of three-dimensionality. He employed nudes and foreshortenings in his figures. This had seldom been done before him. The name Masaccio **PERSON** is a humorous version of **Maso** **LOC** (short for **Tommaso** **PERSON**), meaning "clumsy" or "messy" **Tom** **PERSON**. The name may have been created to distinguish him from his principal collaborator, also called **Maso** **ORG**, who came to be known as **Masolino** **PERSON** ("little/delicate **Tom** **PERSON**").

Despite his brief career, he had a profound influence on other artists and is considered to have started the Early Italian Renaissance in painting with his works in the mid- and late-1420s. He was one of the first to use linear perspective in his painting, employing techniques such as vanishing point in art for the first time. He moved away from the **International Gothic** **ORG** style and elaborate ornamentation of artists like **Gentile da Fabriano** **PERSON** to a more naturalistic mode that employed perspective and chiaroscuro for greater realism.

Masaccio **PERSON** died at the age of twenty-six and little is known about the exact circumstances of his death. Upon hearing of Masaccio **PERSON**'s death, **Filippo Brunelleschi** **PERSON** said: "We have suffered a great loss."

Multi-Class Classification

- The model describes the process of multi-class text classification using Doc2Vec and logistic regression algorithms.
- The Doc2Vec model was used to convert the text data into fixed-length feature vectors, which were then used as input to a logistic regression classifier.
- Doc2Vec is an unsupervised algorithm that can generate fixed-length feature vectors for a given text document by considering the context of words within the document, allowing it to capture the semantic meaning of the text.
- The results showed that the proposed approach achieved an accuracy of over 75%, demonstrating the effectiveness of Doc2Vec and logistic regression for multi-class text classification tasks.




Extractive Vs Abstractive Summarization

- Extraction-based Summarization:

The extractive approach involves picking up the most important phrases and lines from the documents. It then combines all the important lines to create the summary. So, in this case, every line and word of the summary actually belongs to the original document which is summarized. Extractive summaries can often be inconsistent grammatically, as sentences are stitched together haphazardly.

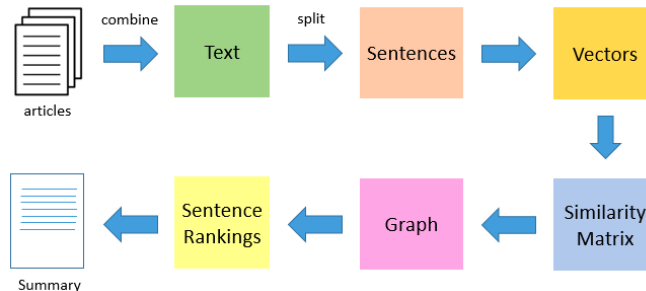
- Abstraction-based Summarization:

The abstractive approach involves summarization based on deep learning. So, it uses new phrases and terms, different from the actual document, while keeping the points the same, just like how we actually summarize. This results in a natural, grammatically accurate summary. This makes it much harder than the extractive approach.



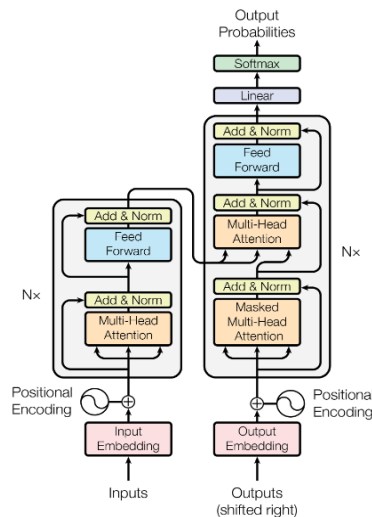
Extractive Summarization Using TextRank

- Like all Extractive Summarization methods TextRank takes parts of the text (Sentences, Phrases, Paragraphs etc). Therefore identifying right sentences for summarization is a vital process for summarization.
- TextRank algorithm is based on PageRank algorithm which was originally developed by Google to rank pages for their search engines.
- Inplace of Web pages, TextRank uses the sentence and its corresponding similarities as a transition is equated to page transition probability.



Abstractive Summarization Using BART

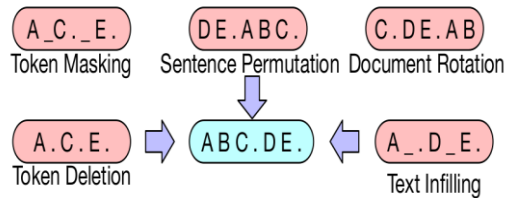
- BART (Denoising Autoencoder from Transformer) is a transformer-based model that was introduced by Facebook AI in 2020.
- BART (**Bidirectional and Auto-Regressive Transformers**) uses a standard seq2seq machine translation architecture with a bidirectional encoder (similar to BERT) and a left-to-right decoder (similar to GPT). The pretraining task involves randomly shuffling the order of the original sentences and a novel in-filling scheme, where spans of text are replaced with a single mask token.
- It is trained to reconstruct the original sentence from a corrupted version of it, which is called denoising auto-encoding. This allows BART to learn more robust representations of the text and to handle more complex language tasks.



How Does BART Summarize Text?

BART is just a standard encoder-decoder transformer model. Its power comes from three ideas:

- BART's encoder is bidirectional. It uses the context from both sides of a word.
- Uses an autoregressive decoder to generate natural-sounding language sequences.
- During pre-training, BART learns a language model that contains all the complexities and nuances of a real-world natural language. By deliberately introducing all kinds of noise, deletions, and modifications in its input data to make it difficult to learn, BART gains the ability to generate linguistically correct sequences even when input text is noisy, erroneous, or missing.
- This denoising language model is versatile and can be adapted for any downstream text generation or understanding tasks like summarization, question-answering, machine translation, or text classification.






Reasons to Use BART

- Most Resilient to Real-World Noisy Data
- Produces Grammatically Correct Summaries
- Simple to Fine-Tune
- Runs on CPUs Without GPUs
- BART's Quality Is Comparable to the Smaller GPT-3 Models
- Efficient Model Size Compared to GPT Models

What sets BART apart is that it explicitly uses not just one but multiple noisy transformations. So, BART learns to generate grammatically correct sentences even when supplied with noisy or missing text.

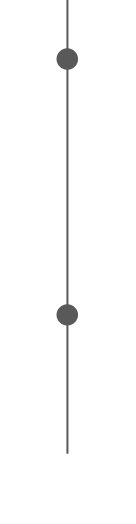
We are using the BART-Large version which is of 1.5GB, it contains 1024 hidden layers and 406M parameters and has been fine-tuned using CNN, on a news summarization dataset.

This is a good trade-off for not using GPT models which are larger in size and require more computational units.





Extractive Summarization using LDA and NMF

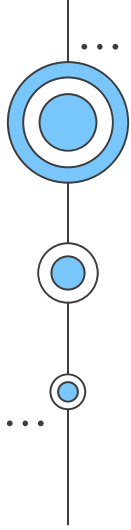
- Latent Dirichlet Allocation (LDA) is a generative statistical model that explains a set of observations through unobserved groups, and each group explains why some parts of the data are similar. The LDA is an example of a topic model. In this, observations (e.g., words) are collected into documents, and each word's presence is attributable to one of the document's topics. Each document will contain a small number of topics. This is a method of topic Modelling.
 - It assumes that documents are generated by a process in which a set of topics are chosen, and then words are generated based on those topics. This means that the topics are latent variables that are inferred from the observed words in the documents.
 - NMF is a form of Topic Modelling – the art of extracting meaningful themes that recur through a corpus of documents. A corpus is composed of a set of topics embedded in its documents. A document is composed of a hierarchy of topics. A topic is composed of a hierarchy of terms.
 - NMF is a matrix factorization technique that decomposes a matrix into two non-negative matrices that represent a set of basis vectors and a set of coefficients. In the context of topic modeling, the vectors represent the topics and the coefficients represent the distribution of topics in each document. It is often used as an alternative to LDA, particularly in cases where the number of topics is small and well-defined.
- 

Article Recommendation

- Article recommendation is a type of recommendation system that suggests relevant articles to users based on their interests and preferences.
- The recommender takes an input article and calculates the cosine similarity between that article and other articles in the same cluster. It then selects the K-nearest neighbors based on the similarity distance and recommends articles that match both the category and the keywords of the input article.
- While cosine similarity is a widely used metric for article recommendation, it has some limitations. For example, it may not work well for articles with very short or very long content. It also does not take into account the temporal aspects of articles, such as the publication date or the popularity of the article. Therefore, it is important to carefully consider the limitations of cosine similarity when designing an article recommendation system.
- The summarizer creates features from the summary of the document in consideration and then finds similar documents based on their cosine similarity

Future Work

- **Multi-lingual support:** Currently, WikiBot only supports English Wikipedia articles. Adding support for other languages would make the tool accessible to a wider audience and expand its usefulness.
- **Fine-tuning summarization algorithms:** While WikiBot's current summarization algorithms work well for most articles, there is always room for improvement. Fine-tuning the algorithms with additional training data and exploring new techniques such as GPT-3 models could improve the quality of the generated summaries.
- **Personalization:** Adding personalization features such as user profiles and learning user preferences could improve the accuracy of the recommendation system and make the tool more engaging for users.
- **Integration with other knowledge sources:** While Wikipedia is a vast and comprehensive knowledge source, integrating WikiBot with other sources such as academic journals, news outlets, and other online databases could provide users with even more information and resources.



Thank you!

