

Document Analysis using LLMs

Presented by

Amay Avasthi

Ishwinderpreet Arora

Sai Murali SN

Shania Jairath

Guided By

Mr Sahibpreet Singh

Mr Puneet Singh



Introduction to Document Analysis

Definition and Importance

- **Document Analysis:** Examining text documents for insights and information.
- **Importance:** Informs decision-making, automates tasks, and aids understanding.

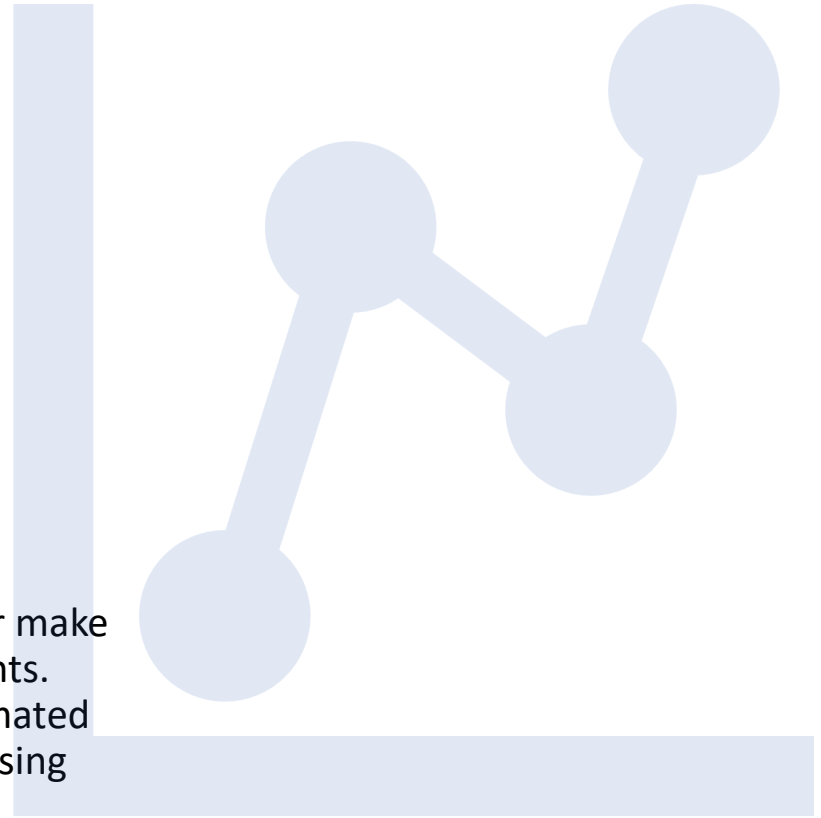
Role of NLP in Document Analysis

- NLP(Natural Language Processing) : AI field for language and understanding.
- NLP in Document Analysis:
 - Preprocessing
 - Information Extraction
 - Sentiment Analysis
 - Document Summarization
 - Language Translation
 - Enhanced Analysis



What is document analysis?

Document analysis refers to the process of examining and evaluating documents to extract information, gain insights, or make decisions based on the content and structure of the documents. This analysis can be performed manually by humans or automated using various technologies, including natural language processing (NLP), machine learning, and computer vision techniques.



Why document analysis?



Information Retrieval: Document analysis helps retrieve relevant information quickly and accurately from vast amounts of unstructured data, such as text documents. This is crucial for search engines, databases, and information retrieval systems, enabling users to find the information they need efficiently.



Knowledge Extraction: Analyzing documents allows organizations to extract valuable knowledge, insights, and trends from textual or visual data. This knowledge can inform decision-making, support research, and facilitate strategic planning.



Automation: Document analysis automates tasks that would be time-consuming and error-prone if performed manually. This can lead to significant cost savings and increased efficiency, especially in industries with large volumes of paperwork, such as finance, healthcare, and legal services.



Data Enrichment: Document analysis can enhance existing datasets by extracting structured information from unstructured documents. For example, extracting customer data from scanned forms or invoices can improve customer relationship management.



Competitive Intelligence: Businesses can analyze documents such as industry reports, news articles, and competitor documents to gain insights into market trends, competitor strategies, and emerging opportunities or threats.



Legal and Compliance: In legal contexts, document analysis is essential for reviewing contracts, court documents, and legal records. It ensures compliance with regulations and supports legal teams in making informed decisions.

USE CASES FOR DOCUMENT ANALYSIS



1

Customer Insights: Analyzing customer feedback, reviews, and surveys provides businesses with valuable insights into customer satisfaction, preferences, and pain points, helping them improve products and services.




2

Healthcare: Document analysis in healthcare involves extracting patient information, diagnoses, and treatment plans from electronic health records (EHRs). This improves patient care, facilitates medical research, and streamlines administrative tasks.




3

Research: In academia and scientific research, document analysis assists researchers in identifying relevant publications, tracking developments in their fields, and synthesizing research findings.



Security and Fraud Detection: Document analysis can be used to identify fraudulent documents, such as forged IDs or counterfeit banknotes. It is also valuable for monitoring and detecting security threats in electronic communications.


Archiving and Preservation: Document analysis is crucial for digitizing and preserving historical documents, manuscripts, and rare materials. It ensures that valuable information is accessible for future generations.



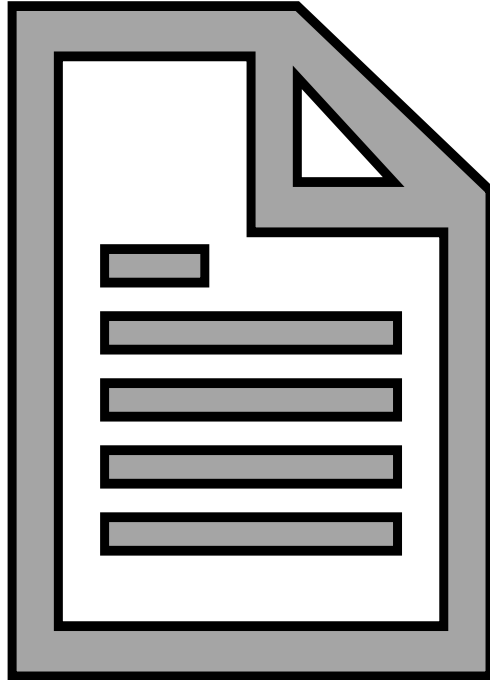
Translation and Localization: Document analysis is used in language translation services to translate documents accurately from one language to another, preserving the original meaning and context.

Media and Entertainment: In the media industry, document analysis supports content recommendation systems and content categorization, making it easier for users to discover relevant content.

Overall, document analysis empowers organizations and individuals to harness the information contained within documents, improve decision-making, and automate tasks, leading to increased efficiency, productivity, and competitiveness in various fields and industries.



Limitations of Document Analysis



- Document analysis, which involves the examination and interpretation of written or printed materials, has its own set of limitations. These limitations can affect the accuracy and comprehensiveness of the analysis. Here are some key limitations of document analysis:
 1. **Incomplete or Biased Information**: Documents may not always provide a complete or unbiased view of a subject. Authors may intentionally omit or distort information, leading to a skewed analysis. Additionally, some documents may be missing important context or details.
 2. **Subjectivity**: Document analysis often relies on the interpretation of the analyst, which can introduce subjectivity into the process. Different analysts may draw different conclusions from the same set of documents, especially when dealing with complex or ambiguous content.
 3. **Lack of Real-time Data**: Documents are static and represent a snapshot of information at a particular point in time. They may not reflect current conditions or developments. This limitation is particularly relevant when analyzing rapidly changing fields or dynamic situations.
 4. **Trustworthiness and Authenticity**: Ensuring the authenticity of documents can be challenging, especially in the age of digital manipulation and forgery. Without proper verification, analysts may inadvertently rely on false or fabricated information.
 5. **Contextual Understanding**: Documents often lack context, making it difficult to fully understand the meaning or significance of the information presented. Analysts may need to rely on external sources or

NLP Libraries for Document Analysis

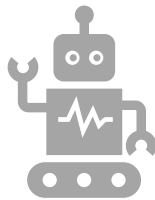


NLTK - Natural Language Toolkit

Overview: Comprehensive Python library for text analysis.

Functions: Tokenization, Stemming, Stop Words, Custom NER.

Use Cases: Preprocessing, Custom NER, Summarization.



spaCy - Industrial-Strength NLP

Overview: Fast, efficient NLP library with pre-trained models.

Functions: Tokenization, Pre-trained NER, Dependency Parsing.

Advantages: Speed, Multilingual Support, Ease of Use.

Applications: Preprocessing, Named Entity Recognition



Integration with LLMs

Role: Both libraries serve as preprocessing steps.

Enhancement: Combining with Large Language Models for advanced analysis.

Benefit: Improved accuracy and expanded capabilities.

NLTK for Text Processing

Tokenization

- **Definition:** Splitting text into words or sentences.
- **Importance:** Structures text for analysis.

Stop Word Removal

- **Purpose:** Eliminating common, less meaningful words.
- **Enhancement:** Focuses analysis on essential content.

Stemming

- **Definition:** Reducing words to their base or root form.
- **Simplifies:** Variations of words to their core.

```
from nltk.tokenize import word_tokenize, sent_tokenize
text = "Tokenization splits text into words or sentences."
words = word_tokenize(text)
sentences = sent_tokenize(text)
```

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word.lower() not in stop_words]
```

```
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in words]
```

NLTK for NER and Summarization

Named Entity Recognition (NER) with NLTK

- **Definition:** Identifying and classifying entities (e.g., names, places) in text.
- **NLTK's Role:** Custom NER models for specific domains.
- **Use Cases:** Healthcare, e-commerce, finance.

Document Summarization with NLTK

- **Methods:** Extractive and abstractive summarization.
- **Applications:** Research papers, legal documents, news articles.
- **Customization:** Tailor NLP for domain-specific tasks.



spaCy - Tokenization and Preprocessing



Efficient Tokenization:

spaCy's fast tokenization efficiently breaks text into words or sentences. Ideal for processing large volumes of text data quickly.



Lemmatization for Consistency:

spaCy's lemmatization ensures words are reduced to their base form. Enhances accuracy and consistency in subsequent analysis.



Part-of-Speech Tagging:

spaCy tags words with their grammatical categories (e.g., noun, verb). Helps understand text structure and content.



Stop Word Removal:

Easily remove common stop words for more meaningful analysis.

spaCy - Named Entity Recognition(NER)



Named Entity Recognition (NER):

Identifies and classifies entities in text.

Entities include people, organizations, locations, dates, currency, etc.

Crucial for structured data extraction.



spaCy's Pre-trained NER Models:

Includes pre-trained models for accurate entity recognition.

Trained on extensive text data.



Entity Types Recognized:

Recognizes persons, organizations, locations, dates, currency, and more.



Applications in Document Analysis:

Supports information extraction, categorization, summarization, and relationship analysis.

Enhances understanding of document content.



Custom Entity Recognition:

Can be customized for domain-specific or unique entity types.

Multilingual Support(spacy)

Multilingual Capability:

spaCy offers robust support for multiple languages.

Ideal for global document analysis projects.

Language Model Availability:

Highlight the availability of pre-trained spaCy models for various languages.

Mention specific languages supported.

Benefits for Global Analysis:

Discuss how multilingual support enhances cross-border understanding.

Mention potential applications in international markets.

Integration with LLMs



Preprocessing Bridge:

spaCy preprocesses text for LLMs.
Ensures clean and formatted data for analysis.



LLMs for Advanced Analysis:

LLMs like BERT and GPT-3 offer robust NLP capabilities.
Excel in summarization, sentiment analysis, content generation, and more.



Practical Applications:

Summarization: Combines spaCy's preprocessing with LLMs for concise document summaries.
Sentiment Analysis: LLMs gauge document emotions.
Content Generation: LLMs automate content creation.



Enhanced Understanding:

spaCy and LLMs together improve document comprehension.
Accurate and context-aware text analysis.



Custom Pipelines:

Create tailored NLP pipelines by combining spaCy and LLMs.
Adapt to project-specific needs.



Example Use Case:

Use Case: Document Summarization

- spaCy tokenizes and cleans a lengthy research paper.
- An LLM generates a concise summary, preserving key findings.
- Result: Efficient understanding of complex documents.

Bag of words (BOW)

- The **Bag of Words (BoW)** model is a fundamental technique in natural language processing (NLP) and text analysis. It simplifies the representation of text data by treating each document as a 'bag' of individual words, disregarding word order and considering only the frequency of words in the text.
- This approach converts text into numerical vectors, making it suitable for various NLP tasks such as text classification, sentiment analysis, and document clustering. BoW serves as a foundational concept for understanding and working with textual data in machine learning and NLP applications

Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up... very very very good movie.



('the', 8),
(',', 5),
('very', 4),
(':', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
.....

Why is the Bag-of-Words algorithm used?



One of the biggest problems with text is that it is messy and unstructured, and machine learning algorithms prefer structured, well defined fixed-length inputs and by using the Bag-of-Words technique we can convert variable-length texts into a fixed-length **vector**.



Also, at a much granular level, the machine learning models work with numerical data rather than textual data. So to be more specific, by using the bag-of-words (BoW) technique, we convert a text into its equivalent vector of numbers.



Let's understand this with an example

Suppose we wanted to vectorize the following:

- *the cat sat*
- *the cat sat in the hat*
- *the cat with the hat*

We'll refer to each of these as a text **document**.

Step 1: Determine the Vocabulary

We first define our **vocabulary**, which is the set of all words found in our document set.

The only words that are found in the 3 documents above are: the, cat, sat, in, the, hat, and with.

Step 2: Count

To vectorize our documents, all we have to do is **count how many times each word appears**:

Now we have length-6 vectors for each document!

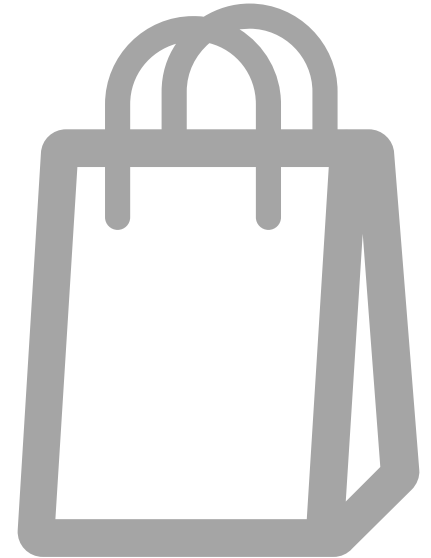
- *the cat sat*: [1, 1, 1, 0, 0, 0]
- *the cat sat in the hat*: [2, 1, 1, 1, 1, 0]
- *the cat with the hat*: [2, 1, 0, 0, 1, 1]

Notice that we lose contextual information, e.g. where in the document the word appeared, when we use BOW.

It's like a literal **bag**-of-words: it only tells you *what* words occur in the document, not *where* they occurred.

Limitations of Bag-of-Words

- Although **Bag-of-Words** is quite efficient and easy to implement, still there are some disadvantages to this technique which are given below.
1. The model **ignores the location information** of the word. The location information is a piece of very important information in the text. For example "today is off" and "Is today off", have the exact same vector representation in the BoW model.
 2. Bag of word models **doesn't respect the semantics** of the word. For example, words 'soccer' and 'football' are often used in the same context. However, the vectors corresponding to these words are quite different in the bag of words model. The problem becomes more serious while modeling sentences. Ex: "Buy used cars" and "Purchase old automobiles" are represented by totally different vectors in the Bag-of-words model.
 3. The **range of vocabulary** is a big issue faced by the Bag-of-Words model. **For example**, if the model comes across a new word it has not seen yet, rather we say a rare, but informative word like Biblioklept (means one who steals books). The BoW model will probably end up ignoring this word as this word has not been seen by the model yet.



TF-IDF

Term Frequency

x

Inverse Document Frequency

The number of times
a word appears in a document

A measure of whether a term is common or rare in a
collection of documents

TF-IDF(Term Frequency-Inverse Document frequency)

- **TF-IDF (term frequency-inverse document frequency)** is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.
- This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.
- **TF-IDF** is much more preferred than **Bag-Of-Words**, in which every word, is represented as **1 or 0**, every time it gets appeared in each Sentence, while, in TF-IDF, gives weightage to each Word separately, which in turn defines the importance of each word than others.

Terminologies

Calculating TF IDF

Calculating Term Frequency

$$tf(t, d) = \frac{\text{Frequency of term } t, \text{ in document } d}{\text{Total number of terms in document } d}$$

Calculating Inverse Document Frequency

$$idf(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$

1.TF: Term Frequency which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

- **TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)**

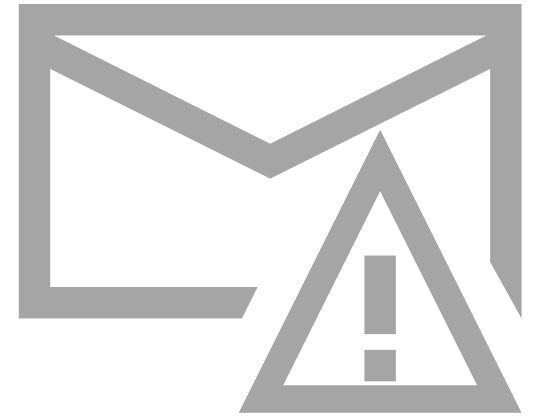
2.IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as “is”, “of”, and “that”, may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

Applications of TF-IDF

Some common applications of TF-IDF in NLP:-

1. **Information Retrieval:** TF-IDF is widely used in search engines and information retrieval systems to rank and retrieve documents based on their relevance to a user's search query. It helps identify documents that contain the most relevant keywords and terms.
2. **Text Classification:** TF-IDF features are often used as input features for machine learning models in text classification tasks. By weighing the importance of words in a document, TF-IDF can improve the performance of classifiers, making them more accurate in categorizing text data into predefined categories..
3. **Document Summarization:** TF-IDF can be employed as part of the process to generate extractive summaries. It helps identify the most significant sentences or phrases within a document, which can be included in the summary.
4. **Spam Detection:** TF-IDF features can be used to build classifiers for spam detection. By analyzing the importance of words and phrases in an email or text message, it's possible to identify spammy content.
5. **Language Processing:** TF-IDF can assist in language identification by comparing the word frequencies in a document to language-specific TF-IDF profiles.
6. **Content Recommendation:** In content recommendation systems, TF-IDF can be used to find content similar to what a user has interacted with in the past, based on shared keywords and terms.
7. **Data Preprocessing:** TF-IDF can be employed as part of the data preprocessing pipeline in NLP tasks. It helps transform raw text data into a more informative and relevant representation for downstream analysis.

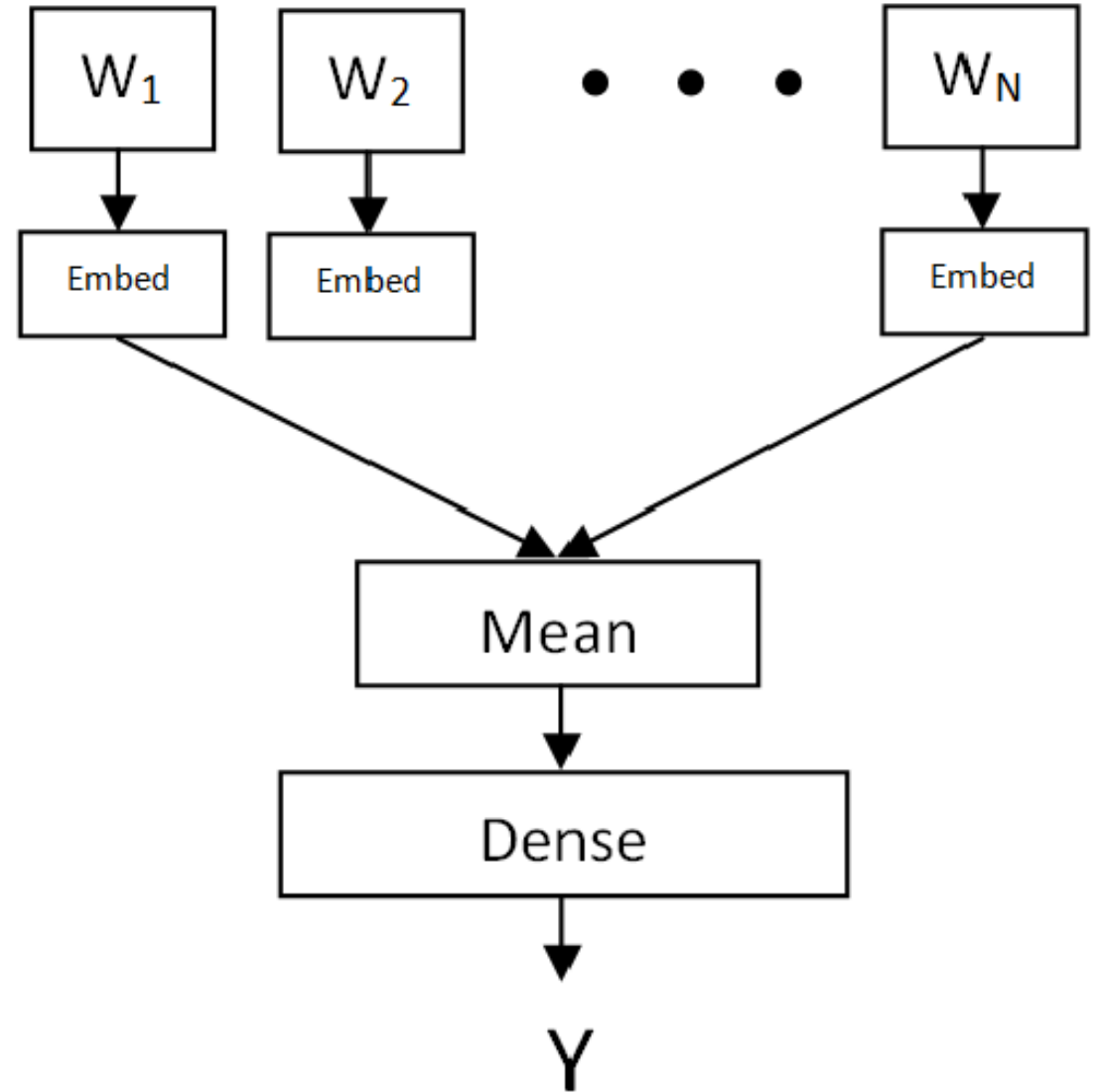


"Evolving Text Analysis: From Bag of Words to TF-IDF"

- The transition from BoW to TF-IDF in text analysis is driven by the need for more sophisticated and effective ways to represent text data. TF-IDF addresses the limitations of BoW by assigning weights to words based on their importance and uniqueness, making it more suitable for various NLP tasks, including information retrieval, text classification, and content recommendation. TF-IDF helps improve the accuracy and relevance of results in these tasks.
- Here are some key reasons why TF-IDF has gained popularity over BoW:
 1. **Term Weighting:** BoW simply counts the frequency of words in a document, which can lead to issues with common words dominating the representation. TF-IDF, on the other hand, assigns weights to terms based on their importance. It considers not only the term frequency (how often a word appears in a document) but also the inverse document frequency (how unique or important a word is across the entire corpus of documents). This weighting helps to highlight important and distinctive terms in a document.
 2. **Dimensionality Reduction:** BoW can result in high-dimensional and sparse representations, which can be computationally expensive and can lead to overfitting in some machine learning models. TF-IDF helps reduce dimensionality by focusing on the most relevant terms, making it more efficient for many NLP tasks.
 3. **Semantic Meaning:** TF-IDF takes into account the distribution of terms across documents, which can capture some aspects of semantic meaning. Terms that are highly specific to a particular document may be indicative of its content, even if those terms don't appear frequently in the corpus.
 4. **Stop Words Handling:** TF-IDF naturally down weights common stop words (e.g., "the," "and" "in"), which are usually not very informative but might appear frequently in documents. BoW would give these words equal importance.

Fast Text

FastText is an open-source, free, lightweight library developed by Facebook's AI Research (FAIR) lab for efficient text classification and text representation learning. It was designed to work well with text data and is particularly useful for tasks like text classification, language identification, and word representation (word embeddings).



Use Cases For FastText

**Text
Classification**

**Language
Identification**

**Text
Clustering**

**Named Entity
Recognition
(NER)**

**Word
Embeddings**

**Document
Similarity**

**Search
Engines**

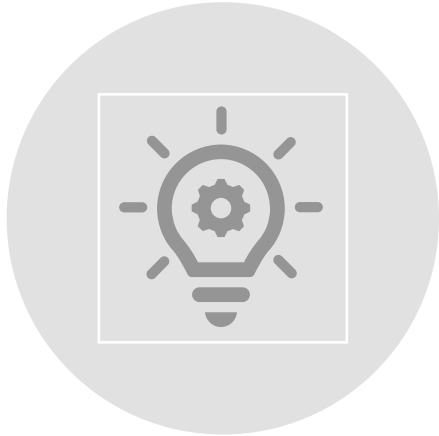
**Topic
Modeling**

**Chatbots and
Virtual
Assistants**

**Cross-lingual
Applications**

**Anomaly
Detection**

Importance of FastText in Document Analysis



EFFICIENT PREPROCESSING:- LLMs LIKE GPT-3.5 ARE POWERFUL FOR UNDERSTANDING AND GENERATING TEXT, BUT THEY ARE COMPUTATIONALLY INTENSIVE AND REQUIRE SIGNIFICANT RESOURCES. FASTTEXT CAN BE USED AS A PREPROCESSING STEP TO EFFICIENTLY CLEAN, TOKENIZE, AND PREPARE THE TEXT DATA FOR ANALYSIS, REDUCING THE COMPUTATIONAL LOAD ON LLMs.



TEXT CLASSIFICATION: FASTTEXT'S STRENGTH LIES IN TEXT CLASSIFICATION, WHICH IS A FUNDAMENTAL TASK IN DOCUMENT ANALYSIS. IT CAN QUICKLY CATEGORIZE DOCUMENTS INTO PREDEFINED CLASSES OR LABELS. WHEN COMBINED WITH AN LLM, FASTTEXT CAN HELP FILTER AND CLASSIFY DOCUMENTS AT SCALE, ENSURING THAT ONLY RELEVANT DOCUMENTS ARE PASSED TO THE LLM FOR MORE IN-DEPTH ANALYSIS.



MULTILINGUAL SUPPORT: FASTTEXT SUPPORTS MULTIPLE LANGUAGES, MAKING IT VALUABLE FOR DOCUMENT ANALYSIS TASKS THAT INVOLVE DOCUMENTS IN DIFFERENT LANGUAGES. THIS IS PARTICULARLY IMPORTANT WHEN WORKING WITH DIVERSE DATASETS OR GLOBAL ORGANIZATIONS.



Speed and Efficiency:

FastText is known for its efficiency in training and inference, which is crucial when dealing with large volumes of documents. It can rapidly process and classify documents, enabling real-time or near-real-time document analysis, which may not be feasible with LLMs alone.



Subword Information:

FastText's ability to represent words as the sum of their subword embeddings can be beneficial when dealing with out-of-vocabulary words or documents in languages with complex morphology. This can improve the LLM's understanding of the document content.



Content Filtering:

FastText can be used to filter and prioritize documents based on criteria like sentiment, topic, or relevance. This ensures that the LLM focuses on documents that are most important for the specific document analysis task, saving computational resources.



Scalability: Document analysis often involves processing large volumes of text data. FastText can be distributed and parallelized easily, allowing for scalable document processing, classification, and feature extraction when working with LLMs.



Customization:

FastText can be fine-tuned or customized for specific document analysis requirements. This allows you to adapt its classification models to the nuances of your document dataset, improving accuracy and relevance.

StopWords

- Stopwords are words that are commonly used in natural language but are typically filtered out or ignored when processing or analyzing text data. These words are considered to have little or no informational value in the context of text analysis, and their removal can help improve the efficiency and accuracy of various natural language processing (NLP) tasks, such as text classification, information retrieval, and text summarization.



Word2Vector



"Word2Vec" (short for "Word to Vector") is a popular natural language processing (NLP) technique used to convert words into numerical vectors. These vectors represent the semantic meaning of words in a high-dimensional space. Word2Vec was introduced by Tomas Mikolov and his team at Google in a series of research papers in 2013.

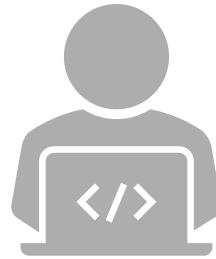


The fundamental idea behind Word2Vec is to create dense vector representations for words in such a way that words with similar meanings or contexts have vectors that are close to each other in this high-dimensional space. Word vectors capture semantic relationships between words, making them valuable for various NLP tasks, including text classification, machine translation, sentiment analysis, and more.

UseCases of Word2Vector



Semantic Understanding: Word2Vec captures semantic meaning and context, allowing it to understand the relationships between words. This is crucial for understanding the nuances of human language.



Feature Extraction: Word2Vec generates numerical vectors for words, which can be used as features in machine learning models. These word embeddings often lead to better model performance in NLP tasks.



Dimensionality Reduction: By representing words in a lower-dimensional vector space, Word2Vec reduces the dimensionality of textual data. This not only saves computational resources but also helps in visualizing and analyzing word relationships.



Improving Search Engines: Word2Vec can enhance the quality of search engines by understanding user queries and matching them with relevant documents or web pages. It helps in better information retrieval.



Text Classification: Word embeddings generated by Word2Vec improve the accuracy of text classification tasks, such as sentiment analysis, spam detection, and topic classification, by providing a more meaningful representation of the text.



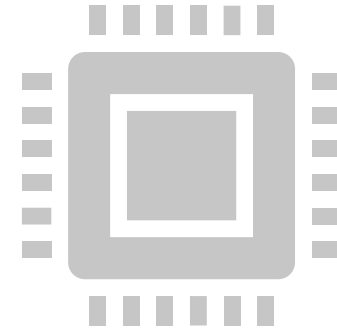
Machine Translation: In machine translation models, Word2Vec embeddings can help identify equivalent words or phrases in different languages, contributing to better translation accuracy.



Recommendation Systems: Word2Vec can be used to recommend products or content based on user preferences and historical behavior. It can capture semantic similarities between items and user profiles.



Topic Modeling: Word embeddings can be utilized in topic modeling techniques like Latent Dirichlet Allocation (LDA) to improve the identification and interpretation of topics in large text corpora.



Multilingual Applications: Word2Vec can be extended to handle multiple languages, allowing it to play a vital role in multilingual NLP applications and cross-lingual tasks.

THANK
YOU!

