Scope and opportunities in the areas of

Artificial Intelligence - NLP

Dr. Sarwan Singh

Scientist – D, NIELIT Chandigarh

MSc (III sem) – NLP

SECTION A

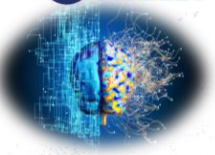
Natural Language Processing – Introduction, Stemming and Lemmatization, Tokenizing, Web Scraping, Part of Speech(POS) Tagging and Sequence Labeling

Named Entity Recognition (NER), Text classification with RNN, Sentiment Analysis, Ethical consideration for NLP, NLTK toolkit

Text Features and TF-IDF Classification

SECTION B

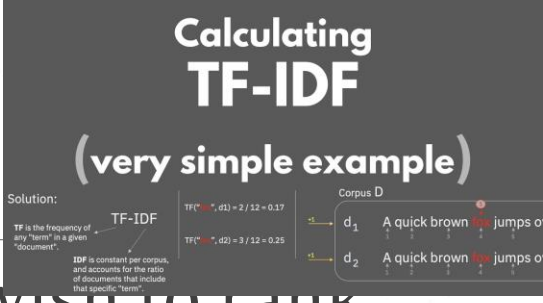
Applications using NLP: Machine translation and learning, summarization and natural language generation, natural language querying, information extraction.



- TF-IDF?
- Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus).

- Terminology :
  - t — term (word)
  - d — document (set of words)
  - N — count of corpus
  - corpus — the total document set

Term Frequency (TF):



- Suppose we have a set of English text documents and wish to rank which document is most relevant to the query , “Data Science is awesome !” A simple way to start out is by eliminating documents that do not contain all three words “Data”,”is”, “Science”, and “awesome”, but this still leaves many documents.
- To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency.
- The weight of a term that occurs in a document is simply proportional to the term frequency.

Formula :

- tf(t,d) = count of t in d / number of words in d

Document Frequency

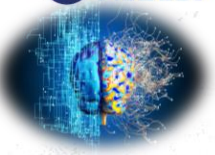
- This measures the importance of document in whole set of corpus, this is very similar to TF.
- The only difference is that TF is frequency counter for a term t in document d, where as DF is the count of occurrences of term t in the document set N.
- In other words, DF is the number of documents in which the word is present.
- We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.
- df(t) = occurrence of t in documents

Inverse Document Frequency(IDF)

- 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 IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as “is” is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.
- idf(t) = N/df

Inverse Document Frequency(IDF)

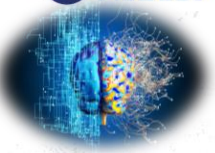
- Now there are few other problems with the IDF , in case of a large corpus,say **100,000,000** , the **IDF** value explodes , to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be **0**. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.
- formula: idf(t) = log(N/(df + 1))
- **tf-idf** now is a the right measure to evaluate how important a word is to a document in a collection or corpus.here are many different variations of TF-IDF but for now let us concentrate on the this basic version.
- Formula : tf-idf(t, d) = tf(t, d) * log(N/(df + 1))



- Natural Language Processing (NLP) is an area of computer science that focuses on the interaction between human language and computers.
- One of the fundamental tasks of NLP is to extract relevant information from large volumes of unstructured data.

TF-IDF – introduction

- TF-IDF is a numerical statistic that reflects the importance of a word in a document. It is commonly used in NLP to represent the relevance of a term to a document or a corpus of documents.
- The TF-IDF algorithm takes into account two main factors: the frequency of a word in a document (TF) and the frequency of the word across all documents in the corpus (IDF).
- The term frequency (TF) is a measure of how frequently a term appears in a document. It is calculated by dividing the number of times a term appears in a document by the total number of words in the document.
- The resulting value is a number between 0 and 1.



- The inverse document frequency (IDF) is a measure of how important a term is across all documents in the corpus. It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents in which the term appears. The resulting value is a number greater than or equal to 0.
- The TF-IDF score is calculated by multiplying the term frequency and the inverse document frequency. The higher the TF-IDF score, the more important the term is in the document.

- TF-IDF = TF * IDF
  TF-IDF = TF * log(N/DF)
- Where:
  . TF is the term frequency of a word in a document
  . N is the total number of documents in the corpus
  . DF is the document frequency of a word in the corpus (i.e., the number of documents that contain the word)

How does TF-IDF work

- TF-IDF is a commonly used technique in Natural Language Processing (NLP) to evaluate the importance of a word in a document or corpus. It works by assigning weights to words based on their frequency and rarity.
- Let's see how TF-IDF works.
- Suppose we have a corpus of five documents
  - Doc1: The quick brown fox jumps over the lazy dog
    Doc2: The lazy dog likes to sleep all day
    Doc3: The brown fox prefers to eat cheese
    Doc4: The red fox jumps over the brown fox
    Doc5: The brown dog chases the fox

- **Step 1: Calculate the term frequency (TF)**
  The term frequency (TF) is the number of times the word appears in the document. We can calculate the TF for the word “fox” in each document as follows:
  - TF = (Number of times word appears in the document) / (Total number of words in the document)
    Doc1: 1 / 9
    Doc2: 0 / 8
    Doc3: 1 / 7
    Doc4: 2 / 8
    Doc5: 1 / 6

- **Step 2: Calculate the document frequency (DF):**
  The document frequency (DF) is the number of documents in the corpus that contain the word. We can calculate the DF for the word "fox" as follows:
  - DF = 4 (Doc1, Doc3, Doc4 and Doc5)
- **Step 3: Calculate the inverse document frequency (IDF):**
  The inverse document frequency (IDF) is a measure of how rare the word is across the corpus. It is calculated as the logarithm of the total number of documents in the corpus divided by the document frequency. In our case, we have:
  - IDF = log(5/4) = 0.2231

- **Step 4: Calculate the TF-IDF score:**
  The TF-IDF score for the word “fox” in each document can now be calculated using the following formula:
  - TF-IDF = TF * IDF

  Doc1: 1/9 * 0.2231 = 0.0247
  Doc2: 0/8 * 0.2231 = 0
  Doc3: 1/7 * 0.2231 = 0.0318
  Doc4: 2/8 * 0.2231 = 0.0557
  Doc5: 1/6 * 0.2231 = 0.0372

- Therefore, the TF-IDF score for the word “fox” is highest in Doc4 indicating that this word is relatively important in this document compared to the rest of the corpus. On the other hand, the TF-IDF score is zero in Doc2, indicating that the word “fox” is not relevant in this document.

Implement TF-IDF in Python using the scikit-learn

- The algorithm works as follows:
  1. Preprocessing: The text data is preprocessed by removing stop words, punctuation, and other non-alphanumeric characters.
  2. Tokenization: The text is tokenized into individual words.
  3. Instantiate TfidfVectorizer and fit the corpus
  4. Transform that corpus to get the representation

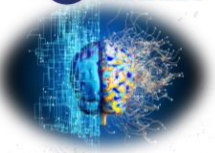
```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import re
from sklearn.feature_extraction.text import TfidfVectorizer

# Sample corpus of documents
corpus = ['The quick brown fox jumps over the lazy dog.',
          'The lazy dog likes to sleep all day.',
          'The brown fox prefers to eat cheese.',
          'The red fox jumps over the brown fox.',
          'The brown dog chases the fox'
         ]
```

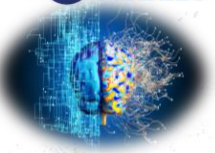
Advantages of TF-IDF

Some of the advantages of using TF-IDF include:

- Measures relevance: TF-IDF measures the importance of a term in a document, based on the frequency of the term in the document and the inverse document frequency (IDF) of the term across the entire corpus. This helps to identify which terms are most relevant to a particular document.
- Handles large text corpora: TF-IDF is scalable and can be used with large text corpora, making it suitable for processing and analyzing large amounts of text data.
- Handles stop words: TF-IDF automatically down-weights common words that occur frequently in the text corpus (stop words) that do not carry much meaning or importance, making it a more accurate measure of term importance.



- Can be used for various applications: TF-IDF can be used for various natural language processing tasks, such as text classification, information retrieval, and document clustering.
- Interpretable: The scores generated by TF-IDF are easy to interpret and understand, as they represent the importance of a term in a document relative to its importance across the entire corpus.
- Works well with different languages: TF-IDF can be used with different languages and character encodings, making it a versatile technique for processing multilingual text data.



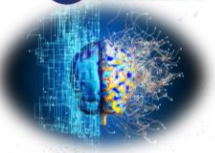
Limitations of TF-IDF

Here are a few limitations of TF-IDF:

- Ignores the context: TF-IDF only considers the frequency of each term in a document, and does not take into account the context in which the term appears. This can lead to incorrect interpretations of the meaning of the document.
- Assumes independence: TF-IDF assumes that the terms in a document are independent of each other. However, this is often not the case in natural language, where words are often related to each other in complex ways.
- Vocabulary size: The vocabulary size can become very large when working with large datasets, which can lead to high-dimensional feature spaces and difficulty in interpreting the results.

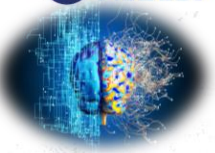
Limitations of TF-IDF

- No concept of word order: TF-IDF treats all words as equally important, regardless of their order or position in the document. This can be problematic for certain applications, such as sentiment analysis, where word order can be crucial for determining the sentiment of a document.
- Limited to term frequency: TF-IDF only considers the frequency of each term in a document and does not take into account other important features, such as the length of the document or the position of the term within the document.
- Sensitivity to stopwords: TF-IDF can be sensitive to stop words, which are common words that do not carry much meaning, but appear frequently in documents. Removing stop words from the document can help to address this issue.



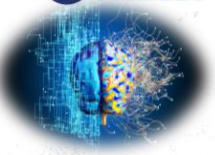
Applications of TF-IDF

- Search engines: TF-IDF is used in search engines to rank documents based on their relevance to a query. The TF-IDF score of a document is used to measure how well the document matches the search query.
- Text classification: TF-IDF is used in text classification to identify the most important features in a document. The TF-IDF score of each term in the document is used to measure its relevance to the class.
- Information extraction: TF-IDF is used in information extraction to identify the most important entities and concepts in a document. The TF-IDF score of each term is used to measure its importance in the document.



Applications of TF-IDF

- Keyword extraction: TF-IDF is used in keyword extraction to identify the most important keywords in a document. The TF-IDF score of each term is used to measure its importance in the document.
- Recommender systems: TF-IDF is used in recommender systems to recommend items to users based on their preferences. The TF-IDF score of each item is used to measure its relevance to the user's preferences.
- Sentiment analysis: TF-IDF is used in sentiment analysis to identify the most important words in a document that contribute to the sentiment. The TF-IDF score of each word is used to measure its importance in the document.



- https://www.kaggle.com/code/yassinehamdaoui1/creating-tf-idf-model-from-scratch