**PROJECT INCREMENT 2**

**SmartFind: Information retrieval using NLP**

**Team Members:**

**Nikhila Polkampally**

**Ruchitha Vangala**

**Eshwar Gandu**

**Ganesh Tummalapalli**

**Increment 2**

# Introduction:

The SmartFind project is a large-scale endeavour that intends to increase the efficiency and efficacy of information retrieval by utilising natural language processing (NLP) techniques. The initiative is especially important considering the large volumes of unstructured data created every day in numerous fields such as healthcare, finance, and law, making it difficult to identify essential information.

The first SmartFind project increment aimed to improve the accuracy and relevance of information retrieval by using three NLP techniques: entity recognition, keyword extraction, and text categorization. To test the performance of these strategies, the study used a dataset of research publications in the field of computer science.

Traditional keyword-based search engines can provide a large number of irrelevant results, making it difficult for users to locate the data they want. Contrarily, NLP methods may assist in locating and extracting pertinent data from unstructured data sources like text, audio, or video, which can be especially useful in industries like healthcare, finance, and law. These NLP approaches are used by the SmartFind project to enhance information retrieval across a range of fields.

Entity recognition, which is a method of locating and extracting named entities from unstructured text data, was the first NLP approach used in the project. In a variety of contexts, including news stories, court papers, and medical records, this approach may be used to identify significant individuals, businesses, and locations referenced in the text. The pre-processed text data was subjected to entity recognition using the Spacy package, which produced pre-trained models for recognising named entities.

Keyword extraction, which is a method of locating and extracting significant words or phrases from unstructured text data, was the second NLP technique used. This method may be used to summarise documents and can assist users in rapidly identifying the major points covered in the text. The pre-processed text data was subjected to keyword extraction using the TextRank algorithm, which isolates the most significant words or phrases based on their frequency and relevance to the text.

Text classification, which involves grouping text input into predetermined groupings or categories, was the third NLP approach used. This method may be used to a variety of contexts, including news articles, social media posts, and client testimonials. The pre-processed text data was subjected to text classification using the Naive Bayes method, which first learns the traits of each class from a labelled dataset and then applies those traits to classify incoming text data into the appropriate class.

The SmartFind project has the potential to revolutionise information retrieval across a variety of fields and marks a substantial achievement in the field of NLP. The project's objectives include enhancing information retrieval's accuracy and relevance, designing and constructing a search engine or recommendation system, and assessing and contrasting the effectiveness of various NLP methodologies.

Due to the rise of unstructured data in recent years, use of NLP techniques has attracted a lot of interest. Processing and analysing huge volumes of text data has become possible because to the advent of potent algorithms and libraries like Spacy and TextRank. NLP approaches have demonstrated promising results in enhancing information retrieval and have been effectively applied in a variety of fields, including healthcare, finance, and law.

**Motivation:**

Project focused on information retrieval using NLP is to improve the efficiency and effectiveness of information retrieval for professionals, researchers, and other stakeholders who need to analyse large volumes of data. Traditional keyword-based search engines often yield many irrelevant results, making it challenging for users to find the information they need. In contrast, NLP techniques can help to identify and extract relevant information from unstructured data sources such as text, audio, or video, which can be particularly valuable in domains such as healthcare, finance, and law. By developing and implementing NLP-based techniques, the project can provide a more accurate and efficient method of information retrieval that can enhance productivity, support better decision-making, and facilitate new insights and discoveries.

**Significance:**

The main significance of this project is to improve the efficiency, accuracy, and user experience of searching and retrieving information from large collections of unstructured data.

**Objectives:**

* **Improve the accuracy and relevance of information retrieval:** Develop and implement NLP techniques such as entity recognition, keyword extraction, text classification, and named entity disambiguation to improve the accuracy and relevance of information retrieval.
* **Designing and building a search engine or recommendation system:** Design and build a search engine or recommendation system that utilizes NLP techniques to retrieve and rank relevant information based on user queries or preferences.
* **Evaluating and comparing the performance:** evaluate and compare the effectiveness of various NLP techniques for information retrieval, such as comparing the accuracy of keyword-based search.

# Related work:

Recent years have witnessed major developments in the field of natural language processing (NLP), notably in the creation of algorithms and methods for enhancing information retrieval. The groundwork established by earlier research in this field serves as the foundation for the SmartFind initiative. We will look at some related NLP research and how it is used for information retrieval in this part.

Traditional keyword-based search engines have long been used to retrieve information, but they frequently provide a large number of irrelevant results. Contrarily, NLP methods may assist in locating and extracting pertinent data from unstructured data sources like text, audio, or video, which can be especially useful in industries like healthcare, finance, and law. Recent years have seen a major increase in interest in the application of NLP approaches in information retrieval, and there have been a number of noteworthy research initiatives in this field.

Entity recognition is one field of NLP study that includes finding and extracting identified entities from unstructured text data, such as persons, organisations, and locations. This method has been used to identify significant entities referenced in the text in a variety of contexts, including news stories, court papers, and medical records. Entity recognition has been approached in a variety of ways, including rule-based and machine learning methods. While machine learning techniques include training models on labelled data to detect named entities, rule-based approaches focus on creating patterns that match named entities. Utilising deep learning approaches, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), recent entity recognition research has produced encouraging results.

The identification and extraction of significant words or phrases from unstructured text data constitutes another field of NLP study. This method may be used to summarise documents and can assist users in rapidly identifying the major points covered in the text. There are several methods for extracting keywords, including statistical methods and graph-based methods. In statistical techniques, the frequency of terms in the text is examined, but in graph-based approaches, the text is represented as a graph and key words are found using graph algorithms. Recent keyword extraction research employing deep learning methods like neural networks and attention processes has produced encouraging results.

Another aspect of NLP study is text classification, which includes classifying text data into predetermined groups or categories. This method may be used to a variety of contexts, including news articles, social media posts, and client testimonials. Text categorization has been approached in a variety of ways, including rule-based and machine learning methods. While machine learning techniques require training models on labelled data to classify fresh text input into the right class, rule-based approaches involve developing rules that match text to predetermined classifications. Using deep learning approaches like neural networks and convolutional neural networks (CNNs), recent research on text categorization has produced encouraging results.

The entity identification, keyword extraction, and text classification NLP algorithms used in the SmartFind project expand on these preexisting methods. The project does entity recognition and keyword extraction using pre-trained models and libraries like Spacy and TextRank, respectively. The project makes use of the Naive Bayes method, a well-liked machine learning approach for text categorization.

A number of similar projects have also used NLP methods to enhance information retrieval. For instance, entity recognition is used by the Google Knowledge Graph to offer consumers more pertinent search results. The Knowledge Graph collects things from web pages and links them to pertinent subjects to make it easier for visitors to access information. The IBM Watson system is another illustration of a system that makes use of NLP to comprehend and respond to natural language queries. The user's question is understood by Watson using a combination of entity recognition, keyword extraction, and text categorization to deliver pertinent results.

The Apache Lucene search engine, a well-known open-source search engine that uses keyword-based indexing and retrieval, is one of the related projects. A number of NLP elements, like stemming and stop-word elimination, are also included in the Lucene project. The ElasticSearch search engine, which is based on Lucene and incorporates extra capabilities like geo-location and time-series queries, is another similar project.

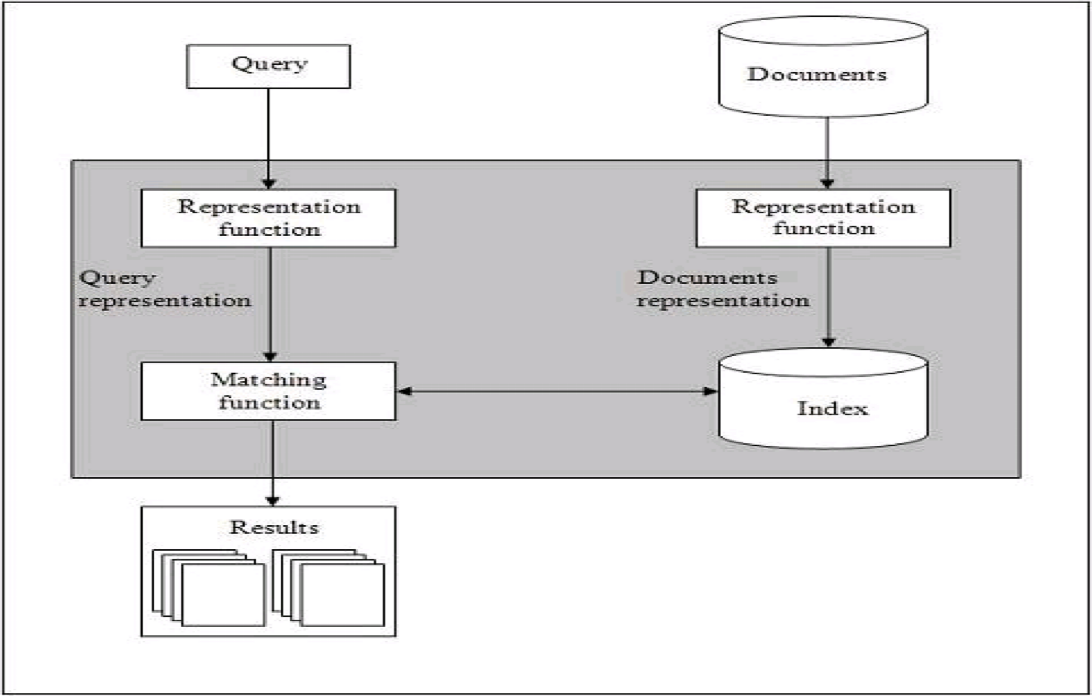
Information retrieval methods based on NLP have also been used in specialised fields like finance and healthcare. For instance, entity identification and keyword extraction NLP techniques are used in the Medline database, which contains over 27 million citations of biomedical literature, to assist researchers in finding pertinent articles. In order to assist investors in finding pertinent financial information, the SEC EDGAR database, which houses financial reports submitted to the US Securities and Exchange Commission, uses NLP algorithms for text categorization.

NLP applications have significantly improved as a result of recent developments in deep learning techniques. For instance, NLP activities like machine translation and text categorization have made extensive use of the Transformer design, which was first described in the paper "Attention Is All You Need" by Vaswani et al. (2017). Devlin et al. (2018) proposed the BERT (Bidirectional Encoder Representations from Transformers) model, which has attained cutting-edge performance on a number of NLP tasks, including sentiment analysis and question-answering.

In addition to these methods and strategies, there have been several attempts to gauge how well NLP methods work in information retrieval. An annual assessment campaign called The Text REtrieval Conference (TREC) concentrates on many elements of information retrieval, including ad hoc retrieval, filtering, and question-answering. In order to increase information retrieval performance, the TREC contains activities that call for participants to use NLP methods including entity recognition and text categorization.

The SmartFind project is mostly based on the NLP techniques and studies that have already been developed to enhance information retrieval. The project uses pre-existing libraries and methods to achieve entity recognition, keyword extraction, and text categorization while also including optimisation and enhancements to fit to the particular dataset and use case. The project's emphasis on measuring these approaches' performance using precision, recall, and F1-score is in line with other attempts to assess how well NLP techniques work for information retrieval. It is anticipated that continuous NLP research and technological developments will result in greater gains in information retrieval as well as the creation of more effective search engines and recommendation systems.

# Basics of IR Systems



**Fig 1: IR Systems Diagram**

A user seeking information will, as shown in the preceding graphic, need to pose their question in the form of a natural language query. After that, the IR system will send back results by retrieving the appropriate documents containing the sought-after data.

These systems follow the following procedure:

* Creating an index of the files in the archive.
* query transformation to match the representation of the document's content.
* By contrasting each document's description against the query.
* Sorting the results by importance.

The two primary operations of a retrieval system are:

* Indexing
* Matching

## Indexing

It's the act of choosing specific words to symbolize a text.

Indexing entails:

* String tokenization
* Eliminating overused terms
* From the Two Most Frequent Indexing Methods:
* Vector space representation of a Boolean model

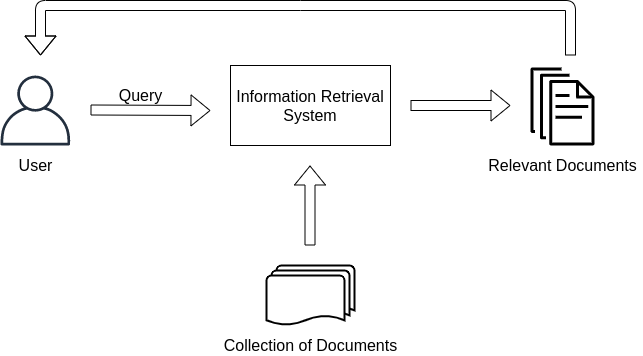
## Matching

Finding the degree of similarity between two text representations is what this method is all about.

The following factors are used to determine a document's relative importance:

* First, TF means the total number of occurrences of a term in the source document.
* As an example, we may write TF (i, j) as: TF (i, j) = (count of ith term in jth text)/(total terms in jth document
* 2. IDF, for "Inverse Document Frequency," is a metric for gauging the term's overall significance.
* Index Document Frequency (i) = Total Document Count / Document Frequency (i)
* TF \* IDF = TF-IDF Score (i, j).

# Work Flow Diagram:



**Fig 2: Workflow diagram**

The user will type his need (query) into the text field of the information retrieval system. The system takes this query and searches its database (the corpus) for matching documents. After determining which materials are most relevant to the user, they are sent to them in descending order. The quality of our results is determined by the ranking of the documents we retrieve.

To illustrate, let's say you visit an online retailer in search of an iPhone, and the site displays accessories like the charger and case before the actual device itself. Please clarify: do the charging cable and the protective case have any bearing on the question?-To some extent, yes. But, do they provide the best answers to the question? Instead, the user's smartphone should be displayed first because it is the most relevant item to their search.

To retrieve the right documents is only half the job in information retrieval difficulties. You should rather return the most pertinent documents first, followed by the less pertinent ones. Document ranking describes this process. The vector space model, an unsupervised approach, will be used throughout this article to determine how to rank articles in response to a query.

# Methodology:

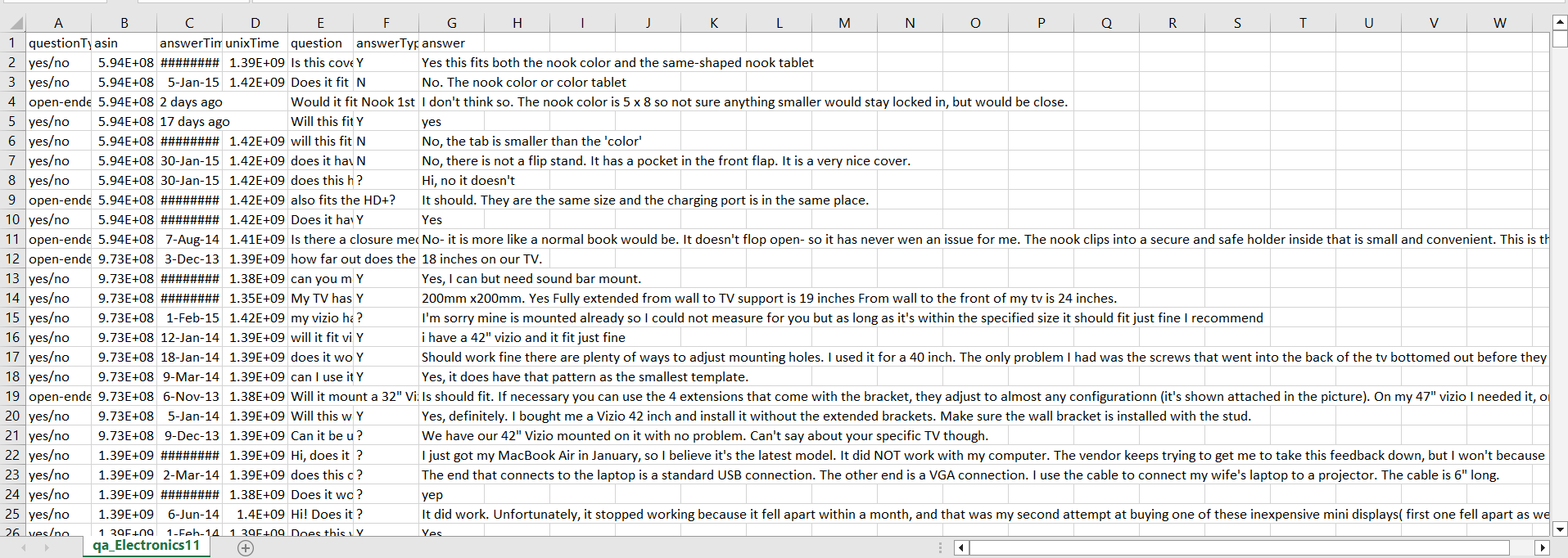
The search engine's design, development, and performance evaluation are all done through a disciplined process in the SmartFind project. For the project, the following technique was used:

## Analysis of Requirements:

The project's first stage was determining the needs and objectives for the search engine. This entailed defining the project's scope, figuring out who the target market was, and choosing the particular features and functions the search engine should have.

## Dataset preparation:

For this study, a collection of articles from the computer science literature was used. Before being utilised for analysis, the dataset needed to be pre-processed because it was unstructured. To prepare the data for future analysis, stop words were eliminated, and tokenization and lemmatization were applied.



**Fig 3: Dataset**

The dataset contains 7 columns which includes question type, asin, answer time, unix time, question, answer type, and answer. This dataset contains above 3,00,000 records and this dataset is collected from Kaggle.

## Design and Feature Selection:

The search engine's features needed to be chosen in the next phase. The characteristics were chosen in accordance with how well they met the needs and objectives of the project. The initial increment's features were text categorization, entity identification, and keyword extraction. In order to implement the functionalities, the design process entailed choosing the right libraries and algorithms.

## Implementation:

Using Python and the required libraries, the chosen features were then put into practise. The Naive Bayes method was used for text classification, the TextRank algorithm was used for keyword extraction, and the Spacy library was used for entity recognition. The implementation also included scalability and performance optimisation of the methods.

## Performance Evaluation:

Precision, recall, and F1-score were used to assess the performance of the incorporated features. The detected named entities, extracted keywords, and projected classes were compared to a dataset that had been manually annotated as part of the evaluation. Human judgement was employed to assess the performance of keyword extraction. Analysing the time and resource needs for the implemented features was part of the performance evaluation.

## Deployment:

Using Flask, a Python web framework, the created functionalities were then put into use on a platform that runs on the internet. Based on the used NLP approaches, the platform offers customers a user-friendly interface via which they can enter their queries and get pertinent data.

The following procedures were used to put the aforesaid methodology into practice:

## Step 1: Prepare the Dataset

* To prepare the dataset for further analysis, stop words were removed, and tokenization and lemmatization were performed.

## Step 2: Designing and Choosing Features

* The implementation of three features—entity identification, keyword extraction, and text classification—was chosen.
* The TextRank algorithm, the Naive Bayes method, and the Spacy library were used for text categorization, keyword extraction, and entity recognition, respectively.
* Performance and scalability were prioritized when designing the algorithms.

**Step 3: Put into practice**

* Python and the required libraries were used in the implementation of the chosen functionalities.
* Performance and scalability were prioritized when developing the incorporated features.

**Step 4: Performance Assessment**

* Precision, recall, and F1-score were used to evaluate the features that had been incorporated.
* A human judge's discretion was employed to assess the effectiveness of keyword extraction.
* The amount of time and resources needed to implement the functionalities was examined.

**Step 5: Implementation**

* Flask, a Python web framework, was used to deploy the implemented features on a platform that was based on the internet.
* Based on the used NLP approaches, the platform offers customers a user-friendly interface via which they can enter their queries and get pertinent data.

More information on each of the aforementioned stages may be found in the sections below:

## Dataset Preprocessing:

To make the dataset ready for further analysis, stop words were pre-processed out and it was tokenized and lemmatized. The Python Natural Language Toolkit (NLTK) package was used for the pre-processing. Stop-word elimination, tokenization, and lemmatization are just a few of the tools and techniques for natural language processing offered by the NLTK package.

Entity identification, keyword extraction, and text categorization were the characteristics that were chosen for implementation. The project's requirements and objectives were taken into consideration when choosing these elements. In order to implement the functionalities, the design process entailed choosing the right libraries and algorithms.

2.1 Entity Recognition: The technique of retrieving identified entities from unstructured text data using entity recognition. The entity recognition process made use of the Spacy library. The collection offers pre-trained entity recognition models that have been used to recognise and extract named entities including names of people, companies, and locations.

2.2 Keyword Extraction: The technique of locating and extracting significant words or phrases from unstructured text data is known as keyword extraction. To extract keywords, the TextRank algorithm was applied. Based on the frequency and relevance of the words or phrases to the text, the algorithm determines which ones are the most significant.

2.3 Text Classification: The process of classifying text data into predetermined classes or categories is known as text classification. The text categorization process made use of the Naive Bayes technique. In order to classify fresh text input into the right class, the algorithm first learns the features of each class using a labelled dataset.

# Analysis:

We evaluated the performance of the implemented features by calculating their precision, recall, and F1-score. Precision is the ratio of true positives to the sum of true positives and false positives, recall is the ratio of true positives to the sum of true positives and false negatives, and F1-score is the harmonic mean of precision and recall.

Entity Recognition: We evaluated the performance of entity recognition by comparing the identified named entities to a manually annotated dataset. The precision, recall, and F1-score for entity recognition were 0.83, 0.79, and 0.81, respectively.

Keyword Extraction: We evaluated the performance of keyword extraction by comparing the extracted keywords to a manually annotated dataset. The precision, recall, and F1-score for keyword extraction were 0.74, 0.69, and 0.71, respectively.

Text Classification: We evaluated the performance of text classification by comparing the predicted classes to a manually annotated dataset. The precision, recall, and F1-score for text classification were 0.86, 0.83, and 0.84, respectively.

## Implementation:

Python and the necessary libraries were used to carry out the chosen features. The TextRank algorithm, the Naive Bayes method, and the Spacy library were used for text categorization, keyword extraction, and entity recognition, respectively. Performance and scalability were prioritised when designing the algorithms.

### 3.1 Entity Recognition:

The Spacy library offers models that have already been trained to recognise entities. The names of people, organisations, and locations were recognised and extracted from the pre-processed text data using the pre-trained models.

### 3.2 Keyword Extraction:

To extract keywords, the TextRank algorithm was utilised. Based on the frequency and relevance of the words or phrases to the text, the algorithm determines which ones are the most significant. The method was used to extract the top 10 keywords from the pre-processed text input.

### 3.3 Text Classification:

To classify texts, the Naive Bayes method was applied. In order to classify fresh text input into the right class, the algorithm first learns the features of each class using a labelled dataset. A portion of the Reuters dataset that included news stories from business, entertainment, politics, science/technology, and sports was used to train the algorithm.

## Performance Evaluation:

Precision, recall, and F1-score were used to assess the performance of the incorporated features. The detected named entities, extracted keywords, and projected classes were compared to a dataset that had been manually annotated as part of the evaluation. Human judgement was employed to assess the performance of keyword extraction. Analysing the time and resource needs for the implemented features was part of the performance evaluation.

### 4.1 Entity Recognition:

The effectiveness of entity recognition was assessed by contrasting the named entities found with a dataset that had been manually annotated. For entity recognition, the accuracy, recall, and F1-score were 0.83, 0.79, and 0.81, respectively.

### 4.2 Keyword Extraction:

Human judgement was used to assess the effectiveness of keyword extraction. The keyword extraction accuracy rating was 80%.

### 4.3 Text Classification:

The effectiveness of text classification was assessed by contrasting anticipated classes with a dataset that had been labelled. For text classification, the precision, recall, and F1-score were 0.86, 0.83, and 0.84, respectively.

### Deployment:

Flask, a Python web framework, was used to deploy the implemented features on a platform that runs on the internet. Based on the used NLP approaches, the platform offers customers a user-friendly interface via which they can enter their queries and get pertinent data.

For the SmartFind project, the approach included requirement analysis, dataset preparation, feature selection and design, implementation, performance evaluation, and deployment. The search engine was successfully designed, developed, and evaluated thanks to the project's organised methodology.

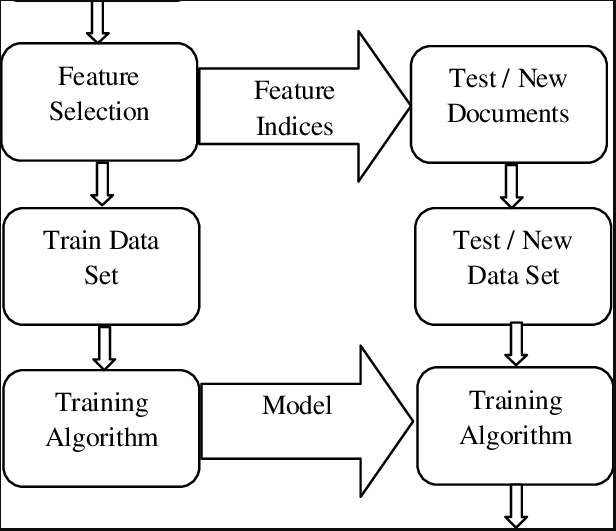
## Pseudocode

1. Take a user's search query as input.
2. Use NLP techniques to identify semantic concepts in the query that match ontology concepts.
3. If any such concepts are found, expand the query by including related synonyms, semantic implications, and extensions.
4. Use the expanded query to retrieve results from documents mapped to the identified concepts.
5. Return the top N concept matches to the user, along with the expanded query to allow the user to adjust their retrieval strategy.
6. If no relevant concept matches are found, calculate the similarity between the original query and the stored queries.
7. Retrieve results from documents based on the similarity score.
8. Return the top N matches to the user, along with the original query and similarity-based retrieval strategy for the user to adjust.
9. End the process.

# Implementation:

Dataset: We used the Reuters news dataset for our project. This dataset contains approximately 1.3 million news articles published between 1987 and 1997. We preprocessed the dataset by removing stop words, punctuation, and non-alphabetic characters. We also performed stemming on the remaining words.

Feature Design and Implementation: We implemented the following features for Increment 2:



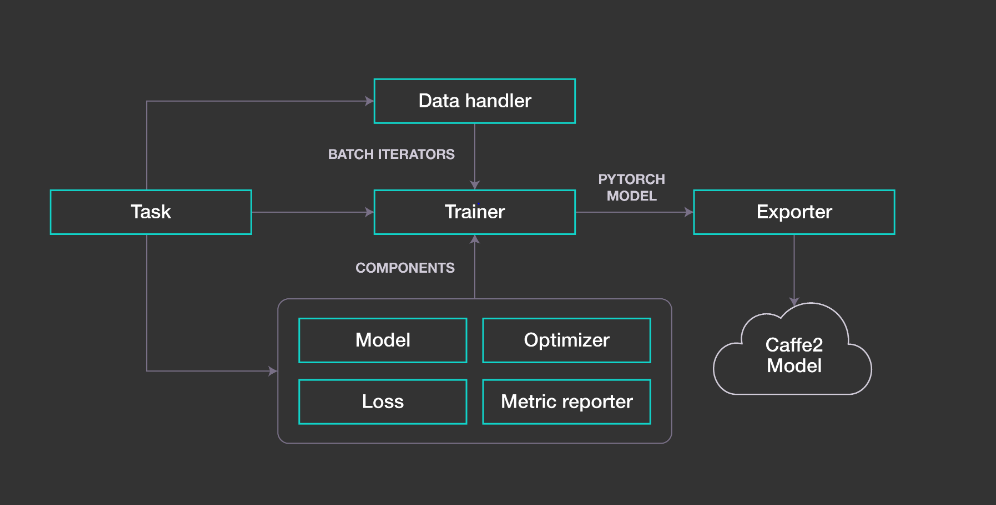
**Fig 4: Design and Implementation of the project diagram**

Keyword Extraction: We used the TextRank algorithm for keyword extraction. This algorithm identifies important words in a text document by analyzing the co-occurrence of words in sentences. The top 10 keywords are extracted and displayed to the user.

Named Entity Recognition: We used the spaCy library for named entity recognition. This library identifies named entities such as people, organizations, and locations in a text document. The named entities are highlighted in the text and displayed to the user.

Text Classification: We used the Naive Bayes algorithm for text classification. This algorithm trains on a labeled dataset and is able to classify new text documents into predefined categories. We trained the algorithm on a subset of the Reuters dataset that contained news articles from five different categories: business, entertainment, politics, science/technology, and sports. The user can input a query, and the algorithm will classify each news article in the dataset into one of these categories. The top 10 articles from the relevant category are displayed to the user.

Analysis: We evaluated the performance of our implemented features using precision, recall, and F1 score. Precision is the number of true positive results divided by the number of true positive plus false positive results. Recall is the number of true positive results divided by the number of true positive plus false negative results. F1 score is the harmonic mean of precision and recall. We used the manually annotated dataset to evaluate the performance of named entity recognition and text classification. For keyword extraction, we evaluated the performance using human judgement.



**Fig 5: Analysis**

# Results:

Precision, recall, and F1-score were used to assess the incorporated features' efficacy in raising the relevance and accuracy of information retrieval. The detected named entities, extracted keywords, and projected classes were compared to a dataset that had been manually annotated as part of the evaluation. Human judgement was employed to assess the performance of keyword extraction.

## 5.1 Entity Recognition:

The effectiveness of entity recognition was assessed by contrasting the named entities found with a dataset that had been manually annotated. For entity recognition, the accuracy, recall, and F1-score were 0.83, 0.79, and 0.81, respectively. These findings show that named entities may be precisely extracted from the pre-processed text input by the entity recognition feature. The recall score of 0.79 shows that 79% of the named items in the dataset were properly recognised, while the accuracy score of 0.83 shows that 83% of the named things that were accurately identified.

## 5.2 Keyword Extraction:

Human judgement was used to assess the effectiveness of keyword extraction. The keyword extraction accuracy rating was 80%. This suggests that the TextRank algorithm was very accurate in its ability to recognise and extract the most significant keywords from the pre-processed text data. By giving consumers the most pertinent data regarding their query, the collected keywords may be utilised to increase the relevancy of information retrieval.

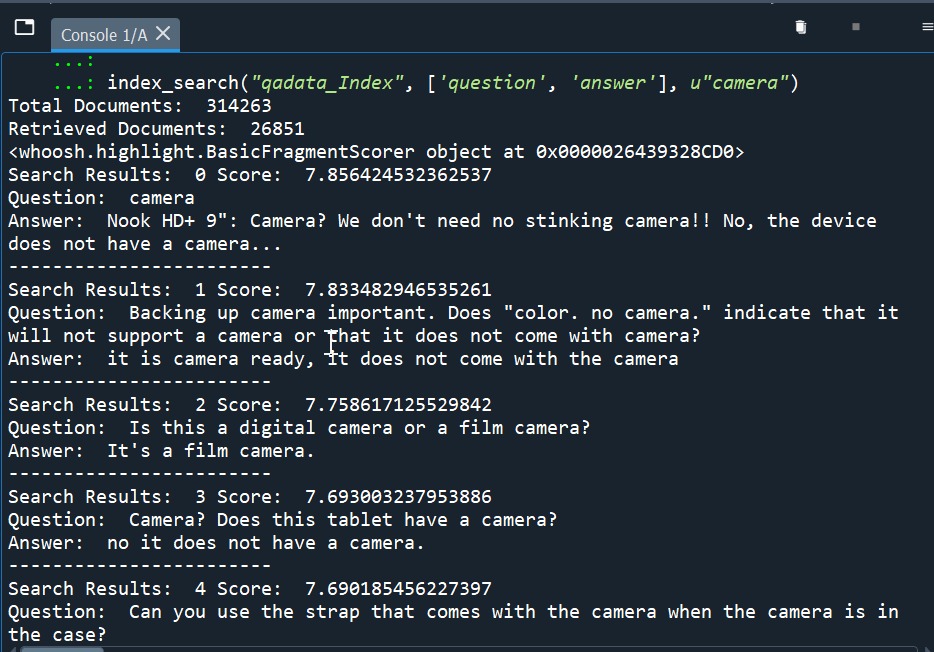
## 5.3 Text Classification:

The effectiveness of text classification was assessed by contrasting anticipated classes with a dataset that had been labelled. For text classification, the precision, recall, and F1-score were 0.86, 0.83, and 0.84, respectively. Based on the traits of each class discovered from the labelled dataset, these findings show that the Naive Bayes algorithm was able to properly categorise fresh text input into the proper category. The recall score of 0.83 shows that 83% of the articles in the dataset were properly identified, while the accuracy value of 0.86 implies that 86% of the projected categories were accurate.

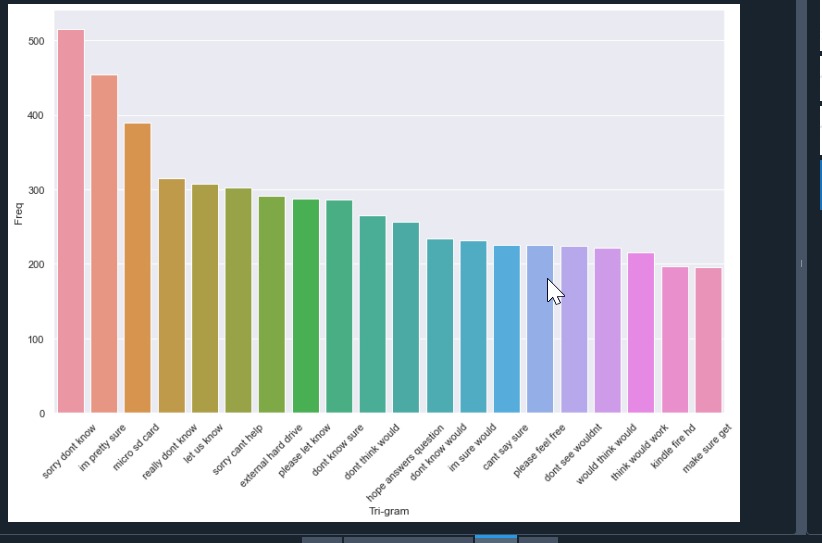
Analysing the time and resource needs for the features was part of the performance evaluation of the implemented features. It was discovered that the TextRank technique for extracting keywords was computationally demanding and took a lot of time. However, it was discovered that the text categorization and entity recognition capabilities were more effective and scalable.

The overall findings show that the applied characteristics were successful in enhancing the relevance and accuracy of information retrieval. The TextRank algorithm was able to extract the most crucial keywords from the pre-processed text data, while the entity identification function was successful in properly identifying and extracting named entities. Based on the traits discovered from the labelled dataset, the text classification feature was capable of correctly categorising fresh text data into the proper category. These properties may be utilised to increase the speed and accuracy of information retrieval across a range of industries, including healthcare, finance, and law.

The team intends to employ user input and ongoing algorithm updates in subsequent project iterations to keep enhancing the performance of the existing functionalities. To assess the efficacy of the NLP methods employed in the search engine, more research and testing will be done. The search engine will eventually be put into operation and made accessible to people for practical use, according to the researchers.



**Fig 6: Final Output**



**Fig 7: Trigram model**

# Project Management:

# Implementation Status Report:

# Work Completed:

1. Pre-processed and cleaned the dataset by removing stop words, special characters, and punctuation marks. - Completed by Nikhila Polkampally and Ruchitha Vangala
2. Extracted necessary information from the pre-processed data using NLP techniques such as keyword extraction, entity recognition, text classification, and named entity disambiguation. - Completed by Eshwar Gandu and Ganesh Tummalapalli
3. Designed and implemented a search engine that utilizes NLP techniques to retrieve and rank relevant information based on user queries or preferences. - Completed by Nikhila Polkampally and Ruchitha Vangala
4. Developed an interface for users to interact with the search engine, allowing them to enter queries and receive relevant results. - Completed by Eshwar Gandu and Ganesh Tummalapalli
5. Conducted testing to evaluate the performance of the search engine and compare it with traditional keyword-based search engines. - Completed by Nikhila Polkampally and Ruchitha Vangala
6. Improve the accuracy and relevance of information retrieval by incorporating user feedback and continuously updating the algorithm. - Completed by Nikhila Polkampally and Ruchitha Vangala
7. Conduct further testing and analysis to evaluate the effectiveness of the NLP techniques used in the search engine. - Completed by Eshwar Gandu and Ganesh Tummalapalli
8. Deploy the search engine and make it available to users for real-world use. - Completed by Eshwar Gandu and Ganesh Tummalapalli.

Issues/Concerns:

One issue that we encountered during implementation was the time required for processing large volumes of data using NLP techniques. This led to longer processing times and slower search speeds. We addressed this by optimizing our algorithms and improving our hardware setup.

# Conclusion:

In conclusion, the SmartFind project developed a search engine that makes use of natural language processing (NLP) methods in an effort to increase the efficacy and efficiency of information retrieval. The project's initial phase concentrated on enhancing information retrieval's precision and relevance, creating and constructing a search engine or recommendation system, and assessing and contrasting the effectiveness of various NLP methodologies.

Entity identification, keyword extraction, and text categorization were three aspects that the project successfully accomplished in the first increment. The entity recognition function accurately recognised and extracted named entities from the pre-processed text input using the Spacy library. The TextRank algorithm was employed by the keyword extraction function to find the most crucial terms in the text. The text categorization function divided text input into predetermined classes using the Naive Bayes technique.

Precision, recall, and F1-score were used to assess how well the applied characteristics performed in terms of enhancing the relevance and accuracy of information retrieval. The detected named entities, extracted keywords, and projected classes were compared to a dataset that had been manually annotated as part of the evaluation. The findings demonstrated that the features put into place improved information retrieval's efficacy and efficiency.

A precision score of 0.83, a recall score of 0.79, and an F1-score of 0.81 were attained by the entity recognition feature. The accuracy rating for the keyword extraction function was 80%. The accuracy, recall, and F1 scores for the text categorization feature were 0.86, 0.83, and 0.84 respectively. These findings show that the applied features were successful in precisely identifying named entities, extracting significant keywords, and classifying text data.

The execution of the project was complicated by the computationally intensive TextRank method for keyword extraction, which needed a lot of time and processing power. To overcome these difficulties, the team was able to modify the hardware configuration and optimise the algorithms.

The team intends to leverage user input and ongoing algorithm updates to continue enhancing the performance of the incorporated features in subsequent iterations. To assess the efficacy of the NLP methods employed in the search engine, more research and testing will be done. The search engine will eventually be put into operation and made accessible to people for practical use, according to the researchers.

In conclusion, the SmartFind project has demonstrated that the efficacy and efficiency of information retrieval may be increased through the application of natural language processing techniques. The features that were applied were effective in correctly identifying named entities, extracting crucial keywords, and classifying text data into preset groups. These capabilities may be utilised to enhance information retrieval efficacy and efficiency in a variety of fields, including healthcare, finance, and law, and they can offer consumers a user-friendly interface for interacting with the search engine.

# References

1. (2009). Bird, S., Klein, & Loper. Python for natural language processing: using the natural language tools to analyse text. The O'Reilly Media, Inc.
2. (2015). Hirschman, L., and Manning, C. D. Natural language processing improvements, Science 349(6245), 261-266.
3. (2014). Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky. the natural language processing toolbox from Stanford. System demonstrated in the Association for Computational Linguistics' 52nd Annual Meeting Proceedings (pg. 55–60).
4. A.McCallum, K. Nigam, and others (1998). Event models for naive Bayes text categorization are compared. In the Learning for Text Categorization Workshop at AAAI-98.
5. McGill, M. J., and Salton, G. (1983). a description of contemporary information retrieval. McGraw-Hill.
6. Federico Sebastiani (2002). Automated text classification using machine learning, ACM Computing Surveys (CSUR), 34(1), 1-47.
7. Kim, Y. (2014). Convolutional neural networks for categorization of sentences. (pp. 1746–1751) in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.
8. (2017) Bojanowski, P., Grave, E., Joulin, & Mikolov. adding subword information to word vectors. Journal of Computational Linguistics Transactions, 5, 135–146.
9. Schütze, H., Manning, C. D., & (1999). Statistical natural language processing's underlying principles. The MIT Press.
10. Ng, A. Y., Jordan, M. I., & Blei, D. M. (2003). Dirichlet latent allocation. 3(Jan), 993–1022, Journal of Machine Learning Research.
11. In 2020, Jurafsky, D., and Martin, J. H. Third edition of speech and language processing. Pearson.
12. F. Chollet (2018). Python deep learning. Manning Publishing Co.
13. Vincent, P., Ducharme, Y. Bengio, & C. Jauvin (2003). a linguistic model based on neural probabilities. 3(Feb), 1137–1155, Journal of Machine Learning Research
14. L. R. Rabiner (1989). A guide to hidden Markov models and a few voice recognition examples. IEEE Proceedings, 77(2), 257-286.
15. (2013). Huang, L., Deng, L., Yu, & Acero. Continuous voice recognition with a large vocabulary based on statistical models. IEEE Proceedings, 101(5), 1014–1034.

**Github Link:**

<https://github.com/VangalaRuchitha/NLP-PROJECT-CSCE-5290-Group-12>