CS6370: Natural Language Processing Final Project

Boneshwar V K (BS20B012)[1], Shivaram V (BE20B032)[2], and Aakash (BS20B001)[3]

Indian Institute of Technology, Madras, Chennai, India
bs20b012@smail.iitm.ac.in[1]
be20b032@smail.iitm.ac.in[2]
bs20b001@smail.iitm.ac.in[3],

Abstract. The final project for the CS6370 Natural Language Processing course, which consists of multiple subproblems, is contained in this article. The project focuses on methods and techniques related to intrinsic natural language processing. Precursor to the second and final part, which creates an effective search engine, is the first section.

Keywords: Natural Language Processing, Search Engine, Cranfield dataset

1 Introduction

The project harnesses Natural Language Processing (NLP) techniques to enhance information retrieval. Initially, it employs IDF (Inverse Document Frequency) to rank documents based on TF-IDF (Term Frequency-Inverse Document Frequency) matrix scores, capturing the importance of terms within documents across a corpus. Subsequently, documents are evaluated using specific metrics, and an Information Retrieval system is constructed. This system leverages cosine similarity to rank documents, effectively measuring the similarity between a query and documents in the corpus. In the subsequent sections, we will delve into the various components of the project, detailing its architecture and enhanced features aimed at designing an efficient search engine.

2 Information Retrieval

One essential tool for locating and obtaining pertinent data from a sizable dataset or collection of documents is an information retrieval (IR) system. These systems play a key role in many different fields, such as enterprise knowledge management platforms and internet search engines. Fundamentally, an IR system employs an intuitive interface to enable smooth communication between people and data. It effectively sorts through enormous volumes of data to give users the most relevant results by utilising strategies like indexing, querying, and ranking. IR systems facilitate efficient information retrieval for both individuals and organisations by arranging and evaluating data in response to user inquiries. This improves decision-making and productivity.

3 Cosine Similarity

Cosine similarity is a metric used to measure the similarity between two vectors in a multi-dimensional space. In Natural Language Processing (NLP), it is widely used in Information Retrieval (IR) systems for comparing documents or text snippets based on their content.

The formula for cosine similarity between two vectors **A** and **B** is given by:

$$CosineSimilarity(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \tag{1}$$

Where $\mathbf{A} \cdot \mathbf{B}$ denotes the dot product of vectors **A** and **B**, and $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ represent the Euclidean norms of vectors **A** and **B**, respectively.

A score between -1 and 1, representing perfect dissimilarity, 0 representing no similarity, and 1 representing perfect similarity, is obtained by computing cosine similarity. In information retrieval systems, documents that score higher on the cosine similarity metric relative to a query are deemed more pertinent to the query and are consequently displayed higher in search results.

4 Evaluation Metrics

Metrics for evaluation are crucial tools for evaluating the efficacy and efficiency of information retrieval (IR) systems. With the use of these metrics, one can objectively assess how well a system responds to user inquiries by obtaining pertinent information. Evaluation criteria that are frequently used include normalised discounted cumulative gain (nDCG), average precision (AP), F1 score, precision, recall, and recall. Whereas recall counts the percentage of relevant documents that are retrieved, precision counts the percentage of relevant documents that are retrieved. A fair assessment of retrieval performance is given by the F1 score, which is the harmonic mean of recall and precision. The system's capacity to rate pertinent documents highly is seen in AP's evaluation of the average precision across various memory levels. Relevance and position are taken into account by DCG while evaluating the calibre of the ranked list of documents. A handful of these metrics will be briefly reviewed in the following subsections, offering an understanding of their applicability and interpretation in assessing IR systems.

4.1 Precision

Precision measures the proportion of retrieved documents that are relevant.

$$Precision = \frac{Numberofrelevantdocumentsretrieved}{Totalnumberofdocumentsretrieved} \tag{2}$$

4.2 Recall

Recall quantifies the proportion of relevant documents that are retrieved.

$$Recall = \frac{Numberofrelevantdocumentsretrieved}{Totalnumberofrelevantdocuments} \tag{3}$$

4.3 F0.5 score

F0.5 score is the harmonic mean of precision and recall with more weight given to precision.

$$F_{0.5} = \frac{(1 + 0.5^2) \times Precision \times Recall}{(0.5^2 \times Precision) + Recall} \tag{4}$$

4.4 AP (Average Precision)

Avreage Precision or AP evaluates the average precision across different recall levels.

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{Totalnumberofrelevantdocuments} \tag{5}$$

4.5 nDCG (Normalized Discounted Cumulative Gain)

nDCG assesses the quality of the ranked list of documents by considering both relevance and position.

$$nDCG = \frac{DCG}{IDCG} \tag{6}$$

Where DCG (Discounted Cumulative Gain) is calculated as:

$$DCG = \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{\log_2(i + 1)} \tag{7}$$

And IDCG (Ideal Discounted Cumulative Gain) is the maximum achievable DCG for the given set of documents.

5 Methodologies

We looked into an introduction of Information Retrieval followed by Cosine Similarity and Evaluation Metrics till now. In this section, we will look into the different methodologies we could use for information retrieval starting with the basic TF-IDF-based document retrieval and its shortcomings followed by several other techniques and a qualitative and quantitative comparison of these methodologies using evaluation metrics.

5.1 Term Frequency - Inverted Document Frequency

In information retrieval and text mining, TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical metric that assesses a term's significance in a document in relation to a corpus of documents. It integrates the two measurements, IDF (Inverse Document Frequency) and TF (Term Frequency).

1. **Term Frequency (TF)**: It measures how frequently a term occurs in a document. It is calculated by dividing the number of times a term appears in a document by the total number of terms in the document. The formula for TF is:

$$TF(t, d) = \frac{Number of times term t appears in document d}{Total number of terms in document d} \tag{8}$$

2. **Inverse Document Frequency (IDF)**: It measures how important a term is across the entire corpus. Terms that occur rarely across the corpus are given higher IDF scores, while terms that occur frequently are given lower IDF scores. IDF is calculated by taking the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the term, and then adding 1 to the divisor to avoid division by zero. The formula for IDF is:

$$IDF(t, D) = \log\left(\frac{Total number of documents in corpus D}{Number of documents containing term t} + 1\right) \tag{9}$$

   Once TF and IDF are calculated, TF-IDF is obtained by multiplying TF and IDF for each term in each document. This results in a matrix where each row represents a document, each column represents a term, and each cell represents the TF-IDF score of a term in a document.

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \tag{10}$$

   The significance of a term in a document in relation to the entire corpus is reflected in the TF-IDF score. Higher TF-IDF scores indicate that a term is more significant or pertinent to the text.

Limitations There are several limitations for an Information Retrieval system which uses TF-IDF to retrieve documents. A few ahve been listed below.

1. **Bag of Words:** The VSM ignores word order and grammar in favor of treating documents like bags of words. Loss of context and ambiguity may result from this. Take the query "apple computer" for example, and examine it within the framework of the Cranfield dataset. Without taking word order into account, papers that contain the terms "apple" and "computer" but do not specifically mention the company "Apple Computers" can incorrectly be given a higher ranking.
2. **Dimensionality ineffectiveness:** Distances between points lose significance in high-dimensional spaces, while individual dimensions lose significance. This may result in problems with sparsity and a decrease in the efficiency of document relationship capture. For instance, the effectiveness of the VSM may decline in the Cranfield dataset if there are a lot of uncommon phrases that have no bearing.

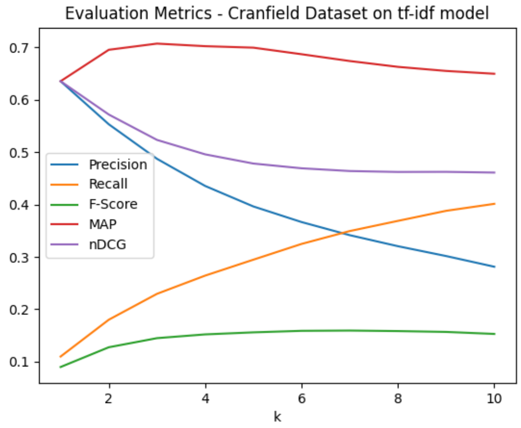

Fig. 1. Evaluation Metric Plot - TF-IDF Retrieval

3. **Scalibility:** For big datasets, VSM may involve keeping a sizable matrix of word frequencies or TF-IDF weights, which can be resource-intensive. Scalability problems may arise while handling large document collections as a result.
4. **Lack of semantic Interpretability:** VSM ignores the meanings and relationships between terms, treating them as separate entities. When similar terms are used differently in the query and the documents, this can lead to erroneous retrieval. For example, if the query is ”heart attack,” any documents stating ”cardiac arrest” may be pertinent, but the lack of semantic comprehension may cause them to be overlooked.s

5.2 Latent Semantic Analysis

In natural language processing and information retrieval, latent semantic analysis (LSA) is a technique used to examine the connections between a collection of documents and the terms they include. Singular value decomposition (SVD) is applied to a matrix representing the frequency of terms in documents to represent documents and terms in a lower-dimensional space. The detection of hidden associations between terms and texts is made possible by this approach, which captures the corpus' latent semantic structure. Following is a summary of the essential LSA equations: Given a term-document matrix A, where A_{ij} represents the frequency of term i in document j, LSA performs singular value decomposition (SVD) on A to factorize it into three matrices:

$$A = U \Sigma V^T \tag{11}$$

where: - U is a matrix representing the relationships between terms in the reduced space, - Σ is a diagonal matrix containing the singular values, indicating the importance of each dimension, and - V^T is a matrix representing the relationships between documents in the reduced space.

By reducing the dimensionality of the space via truncating the matrices U, Σ, and V^T, LSA effectively captures the underlying semantic structure of the document-term space, facilitating tasks such as information retrieval, document clustering, and text summarization.

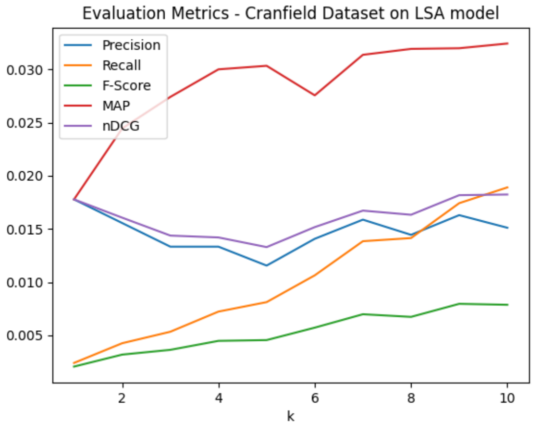

Fig. 2. Evaluation Metric Plots - Latent Semantic Analysis

5.3 Word Embed Retrieval

Word embedding is a technique used in natural language processing to represent words as dense vectors in a continuous vector space, where semantically similar words are located closer to each other. These embeddings capture the contextual meaning of words based on their distributional patterns in large corpora. They are learned using algorithms like Word2Vec, GloVe, or FastText, which process large amounts of text data to generate high-dimensional vector representations for each word.

In this section, the embeddings utilized were Word2Vec embeddings obtained from the Google News corpus, which contains 300 million words or tokens, each represented by a 300-dimensional vector. These embeddings were employed using Python's Gensim library [2]. To integrate these embeddings into the document space of the Cranfield dataset, a projection was made from the document space, where TF-IDF vectors represent documents, to the word embedding space. This was achieved by multiplying the TF-IDF vectors by the Word2Vec matrix, resulting in a final set of vectors representing the words in the Cranfield dataset. This operation effectively mapped the words in the document space to the word embedding space, leveraging the semantic information encoded in the embeddings for downstream tasks such as information retrieval or text classification

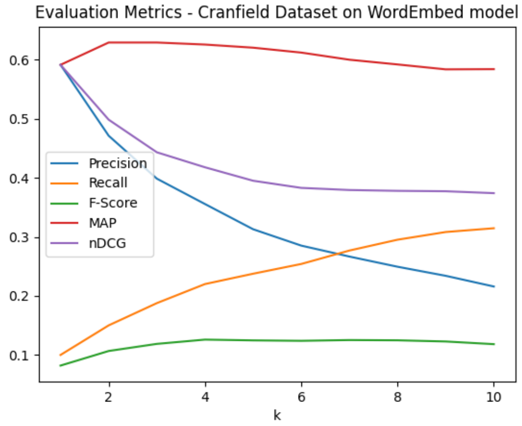

Fig. 3. Evaluation Metric Plot - Word Embed Retrieval

5.4 UMAP-based Retrieval

UMAP, similar to PCA, reduces dimensionality but preserves complex non-linear relationships better. In an IR system, UMAP offers advantages over the Vector Space Model (VSM) by generating dense, semantic embeddings that capture non-linear structures, leading to more accurate similarity measurements and robustness to noise. In the context of an Information Retrieval (IR) system, UMAP can offer several advantages over the traditional Vector Space Model (VSM):

1. **Semantic Embeddings:** Compared to the sparse, high-dimensional representations employed in VSM, UMAP can produce dense, low-dimensional embeddings of documents or concepts that better reflect semantic links. Better similarity metrics and more precise retrieval outcomes may arise from this.
2. **Non-linear Relationships**: UMAP is able to model complex semantic structures that may not be captured by the linear transformations used in VSM since it is able to capture non-linear relationships between terms or documents.
3. **Robustness to Noise:** The ability of UMAP to withstand noise and outliers can help enhance embedding quality and lessen the impact of terms that are irrelevant or noisy in the document collection.

Semantic Embeddings: UMAP can generate dense, low-dimensional embeddings of documents or terms that capture semantic relationships more effectively than the sparse, high-dimensional representations used in VSM. This can lead to better similarity measurements and more accurate retrieval results.

Non-linear Relationships: UMAP can capture non-linear relationships between documents or terms, allowing it to model complex semantic structures that may not be captured by the linear transformations used in VSM.

Robustness to Noise: UMAP's robustness to noise and outliers can help improve the quality of the embeddings and mitigate the impact of noisy or irrelevant terms in the document collection.

Overall, UMAP offers a more flexible and powerful approach to dimensionality reduction and semantic embedding compared to traditional techniques like PCA, making it a promising alternative for enhancing the effectiveness of IR systems.

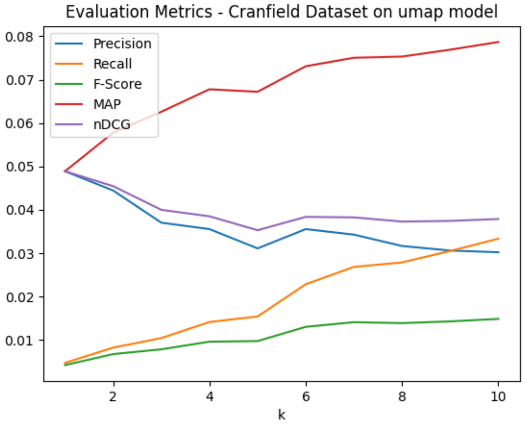

Fig. 4. Evaluation Metric Plot - UMAP Retrieval

6 Comparison

From the given time elapsed comparison, we can infer that the LSA elapses more time followed by UMAP and we have kind of similar time elapsed for both TF-IDF and Word Embed Search.

From the evaluation metrics plot of the methods given, we observe that the LSA performs poorly compared to the other methods. Comparing the other methods we can infer that UMAP performs well than the other algorithms from the MAP metric. TF-IDf, Word Embed Search and UMAP have competitive performance with respect to the other evaluation metrics.

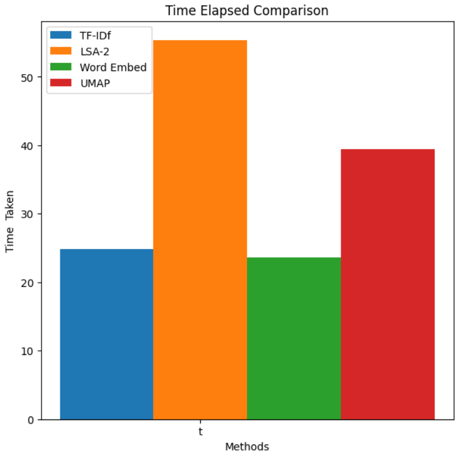

Fig. 5. Time Elapsed

7 Conclusion

In this project, we implement a simple Information Retrieval method TF-IDF and show its shortcomings. We also went through various methods such as LSA, Word Embed and UMAP based methods for IR and evaluate their performance.

References

1. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
2. PyPi library: Gensim. https://pypi.org/project/gensim/
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)