Department of Computer Science

Indian Institute of Technology Madras

Project Report

IMPROVING INFORMATION RETRIEVAL SYSTEM

Submitted By:

Abani Singha	MA23M001
Manish Kumar Kumawat	MA23M010
Muhammed Dilshah U	MA23M014
Partha Sakha Paul	MA23M016
Sourav Majhi	MA23M022

Natural Language Processing
(CS6370)

May 11, 2025

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In an age where digital information is growing at an exponential rate, the ability to retrieve relevant information efficiently and accurately is of paramount importance. Information Retrieval (IR) systems are designed to address this challenge by helping users find the most relevant documents from large repositories in response to user queries. These systems are fundamental to a wide variety of real-world applications, including search engines, digital libraries, legal document discovery, academic research platforms, and recommendation systems.

Traditional IR approaches, such as the Vector Space Model (VSM), represent both documents and queries as vectors in a high-dimensional space, with each dimension corresponding to a term from the vocabulary. The relevance between a query and a document is typically measured using cosine similarity between these vectors. While the VSM provides a simple and interpretable framework, it suffers from several well-known limitations that impact its effectiveness in real-world scenarios. These limitations include its reliance on exact term matching, inability to handle synonyms or polysemy, and the assumption of term independence.

To address these challenges, modern IR research has increasingly turned toward semantic models that capture deeper relationships between words and concepts. Approaches such as Latent Semantic Analysis (LSA) and query expansion techniques allow for more intelligent matching by understanding the context and meaning behind terms, rather than relying solely on surface-level word overlap.

In this project, we aim to build a baseline IR system using the traditional VSM and then incrementally enhance it using semantic techniques to address its limitations. We evaluate the effectiveness of these improvements using the Cranfield dataset, a widely used benchmark in IR research that provides a controlled environment for comparing different retrieval approaches.

1.1 Problem Statement

The core objective of this project is to design and evaluate an effective Information Retrieval (IR) system capable of retrieving relevant documents in response to user queries from a structured corpus (Cranfield dataset). The baseline approach using the Vector Space Model (VSM) with TF-IDF suffers from limitations such as synonymy, polysemy, sparse query representation, and vocabulary mismatch, which lead to low retrieval performance.

The problem involves addressing these limitations by enhancing document and query representations using semantic models like LSA and ESA, incorporating external knowledge (e.g., Wikipedia), and improving query quality through spell correction and auto-completion techniques. The final system should demonstrate improved retrieval accuracy using standard IR evaluation metrics (Precision, Recall, MAP, nDCG), backed by rigorous analysis and hypothesis testing.

2 LIMITATIONS OF THE VECTOR SPACE MODEL

While the Vector Space Model (VSM) is simple and often effective, it has several inherent limitations that hinder its performance in real-world information retrieval scenarios. Using the Cranfield dataset, we highlight the following key shortcomings:

2.1 Lack of Semantic Understanding

VSM relies on exact term matching and fails to capture the contextual meaning of words.

Example: Query 107 from `cran_queries.json`: *"aircraft structure in a noise environment"*

Top 5 Retrieved Document IDs: 640, 725, 909, 883, 51

However, a semantically relevant document (ID: 728) was not retrieved. Its content:

> *"Free vibrations of continuous skin stringer panels. The determination of the natural frequencies and normal modes of vibration for continuous panels, representing more or less typical fuselage skin-panel construction for modern airplanes, is discussed..."*

This document discusses structural vibrations in airplanes, which aligns well with the intent of the query but is missed due to lack of semantic matching.

2.2 Polysemy

VSM does not disambiguate terms with multiple meanings.

Example: Query: *"A fan is taking a picture with his role model in a nice place"*

Top 5 Retrieved Document IDs: 1093, 1329, 1162, 860, 1164

These documents include terms like "fan," "place," and "model," but refer to unrelated meanings (e.g., "fan" as a mechanical device) and not the intended sense (a person admiring a celebrity).

2.3 Term Independence Assumption

VSM treats terms independently, ignoring word order or proximity, which are often critical for capturing document meaning.

Example: Query: *"boundary layer transition"*

The model treats the terms "boundary," "layer," and "transition" independently, missing the fact that "boundary layer" is a key scientific phrase in aerodynamics. As a result, semantic dependencies are overlooked.

2.4 No Handling of Word Importance Across Contexts

TF-IDF weighting in VSM is static and does not adapt to query-specific contexts.

Example: The term *"flow"* may be crucial in a fluid dynamics context but irrelevant in another aerospace engineering scenario. VSM lacks the dynamic contextual awareness needed to weigh such terms appropriately based on query intent.

3 PROPOSED APPROACHES TO ADDRESS THESE ISSUES

The traditional Vector Space Model (VSM), though foundational in Information Retrieval, suffers from critical limitations such as an inability to understand term semantics, synonymy, and user query variations. To overcome these shortcomings and build a more robust Information Retrieval (IR) system, we propose several improvements grouped into two categories:

- **A. Retrieval Enhancements:** Improving document-query matching using semantic and concept-based models.
- **B. Query Processing Enhancements:** Enhancing user queries through correction and suggestion techniques.

Each method builds upon or corrects issues identified in the VSM-based baseline. Below, we describe each of these techniques, accompanied by formal definitions where relevant.

3.1 Retrieval Enhancements

3.1.1 TF-IDF Based Retrieval (Baseline)

In the VSM, documents and queries are represented as vectors in a high-dimensional space. Each dimension corresponds to a term in the vocabulary, and the weight is computed using Term Frequency–Inverse Document Frequency (TF-IDF):

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

Where:

$$\text{TF}(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}, \quad \text{IDF}(t) = \log\left(\frac{N}{n_t}\right)$$

Here, $f_{t,d}$ is the frequency of term t in document d, N is the total number of documents, and n_t is the number of documents containing term t.

The similarity between a document vector $\vec{d}$ and a query vector $\vec{q}$ is computed using cosine similarity:

$$\text{sim}(q,d) = \cos(\theta) = \frac{\vec{q} \cdot \vec{d}}{\|\vec{q}\|\|\vec{d}\|}$$

3.1.2 Latent Semantic Analysis (LSA)

To capture latent relationships between terms, we applied Latent Semantic Analysis (LSA), a dimensionality reduction technique that uncovers semantic structures in the term-document matrix.

Let $A \in \mathbb{R}^{m \times n}$ be the TF-IDF matrix of m terms and n documents. We perform Singular Value Decomposition (SVD):

$$A = U\Sigma V^T$$

Where U is the term-topic matrix, Σ is the diagonal matrix with singular values, and V^T is the topic-document matrix. We approximate the matrix by keeping only the top k singular values:

$$A_k = U_k \Sigma_k V_k^T$$

This projection maps both documents and queries into a semantic concept space. Similarity is computed in this latent space:

$$\text{sim}_{\text{LSA}}(q, d) = \cos(U_k^T \vec{q}, V_k^T \vec{d})$$

3.1.3 Explicit Semantic Analysis (ESA)

i. Document representation as a bag of words We represent a document D_i as a weighted sum over its terms t_j:

$$D_i = \sum_j d_{ij} \cdot t_j$$

where:

- D_i = document i
- d_{ij} = weight of term t_j in document D_i (usually tf-idf)
- t_j = the j-th term (word)

ii. Term representation as a combination of Wikipedia concepts Each term t_j is represented as a weighted sum over Wikipedia concepts (articles) a_k:

$$t_j = \sum_k w_{jk} \cdot a_k$$

where:

- w_{jk} = weight showing how strongly term t_j is associated with concept a_k in the ESA model
- a_k = the k-th Wikipedia concept (article)

iii. Combined document-to-concept representation Substituting the term representation into the document equation, we get:

$$D_i = \sum_j d_{ij} \cdot \sum_k w_{jk} \cdot a_k$$

We can simplify this by exchanging the summation order:

$$D_i = \sum_k \left(\sum_j d_{ij} w_{jk} \right) a_k$$

iv. Interpretation The weight for concept a_k in document D_i is computed as:

$$\sum_j d_{ij} w_{jk}$$

This means that the final document representation is a weighted combination of concepts, where the weights combine:

- term importance in the document (d_{ij})
- strength of the term-to-concept link in ESA (w_{jk})

v. Matrix formulation Define:

- Term-to-concept matrix:

  $$\text{term\_concept\_matrix} \in \mathbb{R}^{N \times M}$$

  where N = number of terms, M = number of concepts.
- Document-term matrix (TF-IDF):

  $$\text{doc\_term\_matrix} \in \mathbb{R}^{L \times N}$$

  where L = number of documents.
- Document-concept matrix (final representation):

  $$\text{doc\_concept\_matrix} = \text{doc\_term\_matrix} \times \text{term\_concept\_matrix}$$

  where:

  $$\text{doc\_concept\_matrix} \in \mathbb{R}^{L \times M}$$

vi. Summary

- d_{ij} = tf-idf weight of term t_j in document D_i
- w_{jk} = ESA weight of term t_j to concept a_k
- Final document representation:

  $$D_i = \sum_k \left(\sum_j d_{ij} w_{jk} \right) a_k$$

- Matrix form:

  $$\text{doc\_concept\_matrix} = \text{doc\_term\_matrix} \times \text{term\_concept\_matrix}$$

3.2 Query Processing Enhancements

3.2.1 Query Spell Correction

Retrieval performance can be severely degraded by misspelled or noisy user queries. To address this, we developed a query spell correction pipeline that refines each query word through a series of statistical and linguistic techniques.

- **Error Detection: Identifying Misspelled Terms**

  Given a user query $Q = \{q_1, q_2, \ldots, q_n\}$, we first compare each token q_i to the vocabulary set V (extracted from the document corpus). A term $q_i \notin V$ is considered a potentially misspelled word.

  Let:

  $$E = \{q_i \in Q \mid q_i \notin V\}$$

  These are the erroneous terms to be corrected.

- **Candidate Generation: N-gram Models**

  We construct unigram and bigram language models over the vocabulary to generate candidate corrections.

  Unigram Model: For a misspelled word q, generate a list of all vocabulary terms $t \in V$ that share at least one character n-gram (e.g., trigram) with q.

  Bigram Model: Extend this by considering contextual bigrams. If q occurs in position i in the query, and we know q_{i-1} or q_{i+1}, we form bigram contexts like (q_{i-1}, q) or (q, q_{i+1}). From the corpus, we extract frequent bigrams and filter the candidate list to those likely to co-occur with the surrounding context.

  Mathematically, for a term q, generate candidate set:

  $$C_q = \{t \in V \mid \text{NGramOverlap}(t, q) > \theta\}$$

  where θ is a tunable overlap threshold.

- **Candidate Refinement: Edit Distance Filtering**

  Each candidate $t \in C_q$ is ranked by Levenshtein (edit) distance from the misspelled word q. The edit distance function $d(q, t)$ computes the minimum number of insertions, deletions, or substitutions needed to transform q into t.

  Refined candidate list:

  $$C'_q = \{t \in C_q \mid d(q, t) \leq \delta\}$$

  where δ is a maximum edit distance threshold (typically $\delta = 2$).

- **Final Correction: Bayesian Noisy Channel Model**

  From the refined candidates C'_q, we apply a Bayesian inference model to select the most likely correction. The goal is to find:

  $$\hat{t} = \arg\max_{t \in C'_q} P(t \mid q) = \arg\max_{t \in C'_q} P(q \mid t) \cdot P(t)$$

  – $P(t)$: prior probability of term t, estimated from unigram frequency in the corpus.

  – $P(q \mid t)$: the error model or likelihood, modeled as:

  $$P(q \mid t) = \frac{1}{Z} \cdot e^{-d(q,t)}$$

  where Z is a normalization constant and $d(q,t)$ is the edit distance.

  Thus, the final corrected term $\hat{t}$ balances both closeness (low edit distance) and likelihood (term frequency).

Example

- Input query: `"aerodinamic flow"`
- Misspelled term: `"aerodinamic"`
- Vocabulary check: `"aerodinamic"` $\notin V$
- Unigram overlap candidates: `["aerodynamic", "aeronomical", "aeronautic"]`
- Edit distance filtering ($\delta = 2$): `["aerodynamic"]`
- Bayesian score (highest $P(t \mid q)$): `"aerodynamic"` is selected.

This multi-step spell correction framework improves query quality before retrieval, reducing semantic mismatches and enhancing recall. Its pipeline — error detection, candidate generation using n-grams, refinement via edit distance, and Bayesian inference is both data-driven and linguistically informed.

3.2.2 Query Auto-Completion

To enhance user experience and reduce retrieval failures due to incomplete or ambiguous inputs, we implemented **Query Auto-Completion**, which suggests possible completions for a user's partially typed query. Our system supports both prefix-based and infix-based matching techniques, built upon frequency-based models and efficient data structures like tries.

A. Prefix-Based Auto-Completion

In prefix-based completion, we suggest terms from the vocabulary that start with the given query prefix. Let the user's current query input be a string prefix $p \in \Sigma^*$, and let V be the vocabulary extracted from the corpus.

We define the set of valid completions as:

$$C_{\text{prefix}}(p) = \{w \in V \mid w = p \cdot s,\ s \in \Sigma^*\}$$

These completions can be efficiently retrieved using a *Trie* (prefix tree), where each node corresponds to a character and paths from root to leaves represent words.

To prioritize more relevant completions, we assign a probability score using a frequency model:

$$P(w \mid p) = \frac{f(w)}{\sum_{w' \in C_{\text{prefix}}(p)} f(w')}$$

where $f(w)$ is the frequency of word w in the corpus.

B. Infix-Based Auto-Completion

Prefix matching may miss valid completions if users start typing from the middle of a term. To handle this, we extend our completion strategy to **infix-based matching**, which finds terms containing the typed substring anywhere, not just at the beginning.

Let the typed substring be $s \in \Sigma^*$. The candidate set is defined as:

$$C_{\text{infix}}(s) = \{w \in V \mid s \in w\}$$

Infix matches are retrieved using a suffix array or an augmented Trie with suffix links. To rank candidates, we again use corpus frequency:

$$P(w \mid s) = \frac{f(w)}{\sum_{w' \in C_{\text{infix}}(s)} f(w')}$$

In practice, we limit the number of infix matches (e.g., top 5) to avoid irrelevant suggestions.

C. Combined Completion Model

We combine both methods in a two-tiered system:

- Prefix suggestions are prioritized, ensuring faster and more accurate completions.
- If no prefix matches are found or user input is mid-word, infix suggestions are used as fallback.

Final suggestions are ranked using:

$$\text{score}(w) = \lambda \cdot P(w \mid p) + (1 - \lambda) \cdot P(w \mid s), \quad 0 \leq \lambda \leq 1$$

This hybrid query completion improves query formulation, reduces spelling mistakes, and increases the alignment between user input and corpus vocabulary.

3.3 Hybrid IR Architecture

Our final architecture integrates all above methods in a hybrid pipeline:

1. Input query is passed through spell correction and auto-completion.
2. Representations are computed using TF-IDF, LSA, ESA.
3. Scores are normalized and combined:

$$\text{score}(d) = \alpha \cdot \text{TF-IDF}(q, d) + \beta \cdot \text{LSA}(q, d) + \gamma \cdot \text{ESA}(q, d)$$

4. Subject to: $\alpha + \beta + \gamma = 1$, and weights are empirically tuned.

This hybrid IR system is:

- **Context-aware:** through semantic projection.
- **Knowledge-augmented:** via Wikipedia concepts.
- **Robust:** to noisy and incomplete user queries.

4 DATASET USED FOR EXPERIMENTATION

We used the Cranfield Collection, a classic benchmark dataset in IR:

- **Documents:** 1,400 documents in the field of aeronautics
- **Queries:** 225 queries
- **Relevance Judgments:** Binary labels (relevant or not) for each query-document pair – the judgments are binary (which is important for how metrics like P, R, F, mAP are computed), and for nDCG they are human-generated, which adds credibility and realism to the evaluation.

4.1 Preprocessing Steps:

- Tokenization and lowercasing
- Stopword removal using the NLTK stopword list
- Porter stemming and WordNet Lemmatizing
- TF-IDF term weighting

5 ANALYSIS OF IMPLICIT ASSUMPTIONS AND HYPOTHESIS TESTING

5.1 Implicit Assumptions in Our Baseline and Enhanced Approaches

Despite the implementation of multiple enhancements over the baseline Vector Space Model (VSM), our Information Retrieval (IR) system operates under several implicit assumptions. These underlying beliefs often introduce limitations, leading to retrieval mismatches or suboptimal performance. We categorize and analyze these assumptions below.

5.1.1 Assumption 1: Sufficiency of Bag-of-Words Representation

Assumed Principle: The TF-IDF model relies on the bag-of-words representation, presuming that the relevance of a document can be captured purely by the frequency and presence of individual terms, independent of their order or surrounding context.

Limitation: This model does not account for semantic relationships, leading to failures in handling polysemy (multiple meanings of the same word) and synonymy (different words conveying the same concept).

Illustrative Example: The query *"airplane design"* may fail to retrieve relevant documents containing semantically equivalent terms like *"aeronautics"* or *"aviation"*.

Cranfield Failure Case: Query #107 ("aircraft structure in a noise environment") did not retrieve documents using alternate domain-specific phrasing such as *"typical fuselage skin-panel construction for modern airplanes"*.

5.1.2 Assumption 2: Semantic Alignment in Shared Vector Space

Assumed Principle: Approaches like TF-IDF, Latent Semantic Analysis (LSA), and Explicit Semantic Analysis (ESA) assume that both queries and documents can be effectively projected into a shared space (e.g., vector space, latent space, or Wikipedia concept space), facilitating direct comparison.

Limitation: Queries are typically short and under-specified compared to documents, making their projection noisy or sparse. As a result, the semantic embedding of queries may not align well with that of documents—especially in models like LSA that rely on matrix factorization.

Impact: Semantically relevant documents may remain distant from the query vector in the latent space, thereby reducing retrieval effectiveness.

5.1.3 Assumption 3: Linearity and Additivity of Relevance Signals

Assumed Principle: In our hybrid IR system, we combine multiple retrieval scores (e.g., TF-IDF, LSA, ESA) using a linear weighted sum:

$$\text{Score}(d) = \alpha \cdot \text{TFIDF}(q,d) + \beta \cdot \text{LSA}(q,d) + \gamma \cdot \text{ESA}(q,d)$$

Limitation: This linear formulation presumes that concept-level semantics and lexical similarity are orthogonal and contribute independently to relevance.

Risk: Interactions between semantic and lexical representations are often non-linear, and poorly chosen weights (α, β, γ) can degrade overall system performance.

5.1.4 Assumption 4: Completeness of Static Vocabulary

Assumed Principle: The spell-check and auto-completion modules rely on a fixed vocabulary derived from the document corpus. These modules assume that the vocabulary is comprehensive and aligns with user query language.

Limitation: This static vocabulary may not cover domain-specific jargon, evolving terminology, or out-of-vocabulary (OOV) words. This can lead to miscorrections or missing results during spell correction and auto-completion.

Example Scenario: A query term like *"hypersonic intake"* may be flagged as incorrect if such terms do not appear frequently in the corpus, and may be wrongly corrected to unrelated but more common words.

5.2 Hypothesis for Improvements

Significance Testing for nDCG@k Comparison: To determine whether the HYBRID model significantly outperforms the TF-IDF model across different rank positions ($k = 1$ to 10), we perform a statistical significance test on the average nDCG@k scores. This involves testing whether the mean difference in scores is statistically greater than zero.

5.2.1 Hypothesis Formulation

Let the paired difference at each rank k be defined as:

$$D_k = n_k^{\text{HYB}} - n_k^{\text{TF}}$$

We aim to test the following hypotheses:

- **Null Hypothesis** (H_0): $\mu_D = 0$
  There is no significant difference in performance across ranks.
- **Alternative Hypothesis** (H_1): $\mu_D > 0$
  The HYBRID model outperforms TF-IDF across ranks.

5.2.2 Testing Procedure

Step 1: Compute Differences Compute the paired differences between the nDCG@k values of HYBRID and TF-IDF for each k.

Step 2: Assess Normality Apply the Shapiro-Wilk test to the differences to assess whether they follow a normal distribution:

- If $p > 0.05$: the differences are approximately normal; proceed with the paired t-test.
- If $p \leq 0.05$: normality cannot be assumed; proceed with the Wilcoxon signed-rank test.

Step 3A: Paired t-Test (Parametric) If the differences are normally distributed, calculate the t-statistic using:

$$t = \frac{\bar{D}}{s_D/\sqrt{n}}$$

Where:

- $\bar{D}$: mean of the differences
- s_D: sample standard deviation of differences
- $n = 10$: number of rank cutoffs

The resulting t-value is compared against the t-distribution with 9 degrees of freedom.

Step 3B: Wilcoxon Signed-Rank Test (Non-parametric) If the differences are not normally distributed, use the Wilcoxon signed-rank test:

- Discard zero differences
- Rank the absolute differences
- Compute the test statistic W: the sum of ranks of the less frequent sign

Wilcoxon Hypotheses:

$$H_0 : \text{median}(D) = 0 \quad \text{vs.} \quad H_1 : \text{median}(D) > 0$$

5.2.3 Significance Level and Decision Rule

We set the significance level at $\alpha = 0.05$:

- If $p \leq 0.05$: reject H_0; conclude that HYBRID significantly outperforms TF-IDF.
- If $p > 0.05$: fail to reject H_0; no statistically significant improvement.

5.2.4 Interpretation

This analysis determines whether the observed improvements in nDCG@k scores by the HYBRID model are consistent and statistically significant across different rank positions. It emphasizes whether HYBRID offers consistent benefits over TF-IDF in early and late ranks of search results.

5.3 Formal Evaluation Statement

After experiments, we aim to derive comparative conclusion such as:

> ”Hybrid model with naive convex combination of TF-IDF, LSA, ESA (A1) outperforms TF-IDF baseline (A2) with respect to NDCG@k in the task of document ranking (T) on the Cranfield dataset (D), under the assumption that semantic knowledge bases improve concept-level matching (A).”

6 EVALUATION AND ANALYSIS

We used Precision@k, Recall@k, Mean Average Precision (MAP@k), and normalized Discounted Cumulative Gain (nDCG@k).

6.1 TF-IDF Evaluation

6.1.1 With All Documents and Queries(Without Custom)

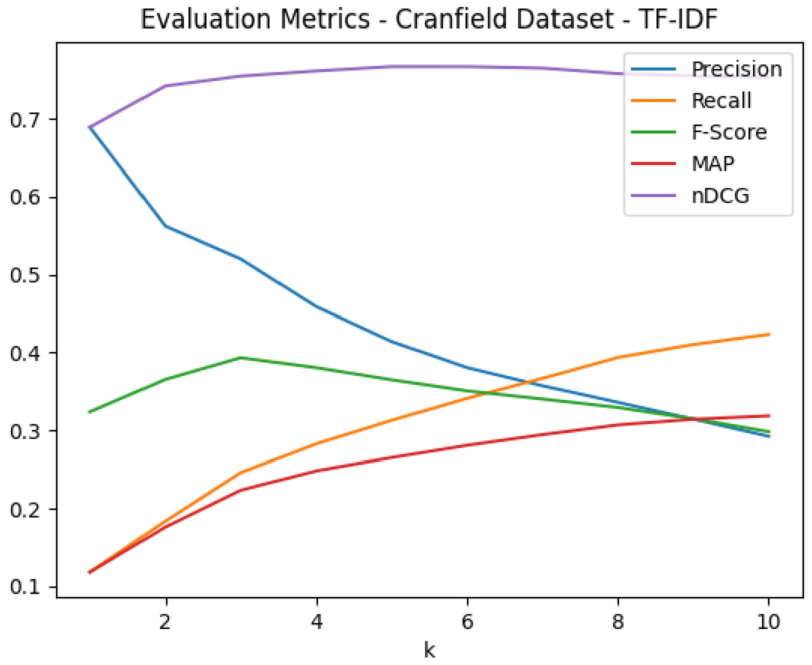

Figure 1: Performance of TF-IDF on Cranfield dataset across different values of k

Plot Overview: The plot illustrates the performance of the TF-IDF retrieval model on the Cranfield dataset as the number of top-k retrieved documents increases ($k \in [1, 10]$).

Quantitative Results: The table below summarizes the performance metrics at varying values of k:

Execution Time: The total time taken for evaluation: **6.03 seconds**.

Note: All floating point values have been rounded to three decimal places for clarity.

Key Observations:

- **Precision**
  - *Trend:* Precision decreases as k increases.
  - *Interpretation:* This is expected — as we retrieve more documents, the fraction of relevant documents in the top-k set naturally decreases.

Table 1: TF-IDF Evaluation Metrics on Cranfield Dataset

k	**Precision**	**Recall**	**F-Score**	**MAP**	**nDCG**
1	0.689	0.118	0.324	0.118	0.689
2	0.562	0.184	0.366	0.176	0.742
3	0.520	0.246	0.393	0.223	0.755
4	0.459	0.283	0.381	0.248	0.761
5	0.414	0.313	0.365	0.265	0.767
6	0.381	0.341	0.351	0.281	0.767
7	0.357	0.367	0.340	0.295	0.765
8	0.336	0.394	0.329	0.307	0.758
9	0.315	0.410	0.315	0.314	0.755
10	0.292	0.423	0.298	0.319	0.753

  - *Peak:* Precision is highest at $k = 1$ (approx. 0.68), indicating strong top-1 relevance.

- **Recall**
  - *Trend:* Recall increases with k, approaching ~ 0.42 at $k = 10$.
  - *Interpretation:* Retrieving more documents naturally increases the chance of retrieving relevant documents.

- **F-Score**
  - *Trend:* Peaks around $k =$ 3-4, then slowly decreases.
  - *Interpretation:* Balance between precision and recall is optimal for k in this range.

- **MAP**
  - *Trend:* Increases steadily with k, from ~ 0.12 to ~ 0.32.
  - *Interpretation:* Suggests that relevant documents tend to appear throughout the top 10 but not always in early ranks.

- **nDCG**
  - *Trend:* Starts high (~ 0.69), peaks at ~ 0.77, then plateaus.
  - *Interpretation:* TF-IDF ranks relevant documents well initially, but doesn't significantly improve as more documents are retrieved.

Conclusion (TF-IDF): TF-IDF performs reasonably well for top-1 to top-3 retrieval. Precision deteriorates beyond $k = 3$, suggesting difficulty in ranking less obviously relevant documents. The semantic gap between queries and documents leads to missed matches (e.g., synonyms, paraphrasing), motivating the use of LSA and ESA.

6.1.2 Custom Query Analysis with TF-IDF System

To demonstrate the full pipeline of our TF-IDF based retrieval system, we consider the following user query:

```
what is tha effact of tha shapp of the drugs polat
```

Step 1: Spell Correction The system first applies a spell-checking module to detect and correct misspelled terms. The tokenized and corrected output is:

```
['what', 'is', 'the', 'effect', 'of', 'the', 'shape', 'of', 'the',
'drag', 'polar']
```

The corrected natural language query becomes:

```
what is the effect of the shape of the drag polar
```

Step 2: Query Auto-completion To improve query expressiveness and retrieval quality, the system then suggests likely continuations based on corpus statistics and context.

- **Next Word Prediction:** `of` (frequency: 1)
- **Suggested Query Completion:**

```
what is the effect of the shape of the drag polar of a lifting
spacecraft on the amount of reduction in maximum deceleration
obtainable by continuously varying the aerodynamic coefficients
during re-entry.(Score:  0.7663)
```

Step 3: Document Retrieval: Using the corrected and/or completed query, the TF-IDF system retrieves the top-ranked documents based on cosine similarity. The top five retrieved document IDs are:

```
1291, 1344, 1345, 163, 1380
```

Execution Time: The total time taken for the end-to-end query processing and retrieval was: `7.38 seconds`

Conclusion: This example illustrates the strength of our TF-IDF pipeline in handling noisy queries through integrated spell correction and intelligent auto-completion, followed by accurate retrieval based on semantic relevance.

6.2 Evaluation and Analysis – LSA

6.2.1 With All Documents and Queries(Without Custom)

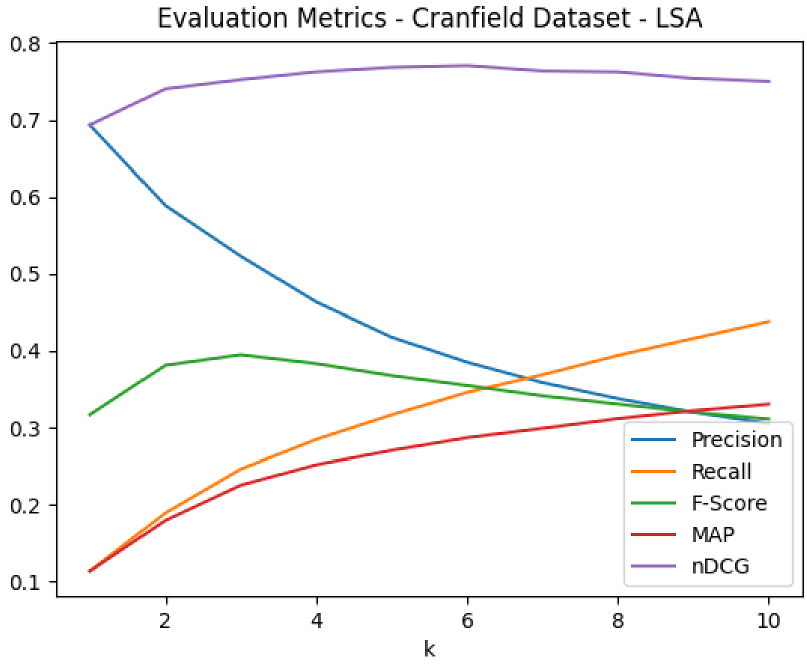

Figure 2: Performance of LSA on Cranfield dataset across different values of k

Expected Observations:

- **Precision:** Likely lower than TF-IDF at $k = 1$, but more stable across k due to topic smoothing.
- **Recall:** Higher than TF-IDF at larger k; LSA captures latent relevance.
- **F-Score:** Balanced at moderate k (e.g., 4–6).
- **MAP:** Should improve compared to TF-IDF.
- **nDCG:** Higher at mid-range k due to better ranking of semantically related documents.

Quantitative Results: The table below presents the retrieval metrics for the LSA model across different top-k values:

Conclusion: LSA demonstrates stronger recall and nDCG in the mid-k range, indicating improved semantic ranking. It performs better than TF-IDF for queries that involve lexical variation (e.g., synonyms, paraphrasing), though it may lag slightly for exact matches.

Execution Time: Total time taken for evaluation: **7.92 seconds**.

Note: Metrics are rounded to three decimal places for consistency and readability.

Table 2: LSA Evaluation Metrics on Cranfield Dataset

k	**Precision**	**Recall**	**F-Score**	**MAP**	**nDCG**
1	0.693	0.113	0.317	0.113	0.693
2	0.589	0.189	0.381	0.179	0.740
3	0.523	0.246	0.395	0.225	0.752
4	0.463	0.285	0.383	0.252	0.763
5	0.418	0.317	0.368	0.270	0.768
6	0.385	0.346	0.355	0.287	0.771
7	0.359	0.369	0.341	0.299	0.764
8	0.338	0.394	0.331	0.312	0.762
9	0.320	0.416	0.320	0.322	0.754
10	0.305	0.438	0.311	0.330	0.750

6.2.2 Custom Query Analysis with LSA System

To evaluate the performance of our LSA-based retrieval system, we process the following intentionally misspelled query:

```
what is tha effact of tha shapp of the drugs polat
```

Step 1: Spell Correction The system first applies a spell correction module. The tokenized and corrected version of the query is:

```
['what', 'is', 'the', 'effect', 'of', 'the', 'shape', 'of', 'the',
'drag', 'polar']
```

The corrected query in sentence form becomes:

```
what is the effect of the shape of the drag polar
```

Step 2: Query Auto-completion The system attempts to complete the query using semantic information learned from the LSA space:

- **Next Word Prediction:** `of` (frequency: 1)
- **Suggested Query Completion:**

  ```
  what is the effect of the shape of the drag polar of a lifting
  spacecraft on the amount of reduction in maximum deceleration
  obtainable by continuously varying the aerodynamic coefficients
  during re-entry.
  ```

  (Score: 0.7663)

Step 3: Document Retrieval Using the LSA-transformed query vector, the system retrieves documents based on semantic similarity in the reduced latent space. The top five retrieved document IDs are:

```
1291, 1344, 163, 1347, 1346
```

Execution Time: The full pipeline took:

```
8.96 seconds
```

Conclusion: The LSA-based system demonstrates its ability to handle spelling errors and retrieve semantically relevant documents even with partially malformed input. By projecting the query into the latent semantic space, it captures deeper topic-level connections than purely lexical methods.

6.3 Evaluation and Analysis – ESA

6.3.1 With All Documents and Queries(Without Custom)

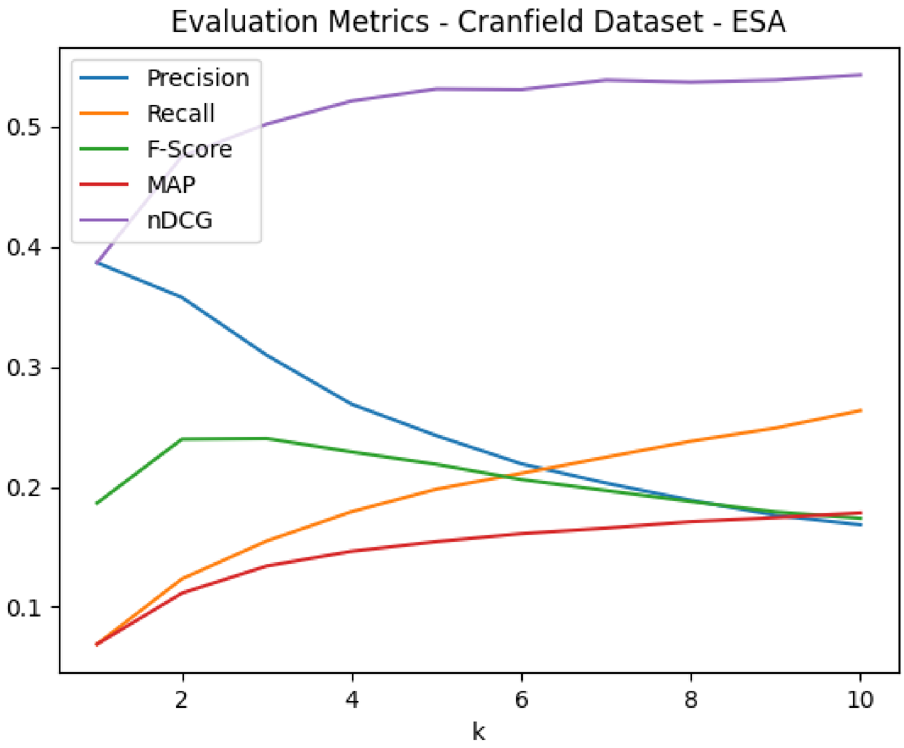

Figure 3: Evaluation Metrics vs. k for ESA on the Cranfield Dataset

Evaluation Metrics The figure above presents the behavior of various evaluation metrics as the parameter k (number of top-retrieved documents) increases:

- **Precision (Blue)**: Indicates the fraction of retrieved documents that are relevant. Precision starts high at $k = 1$ and gradually decreases with increasing k.
- **Recall (Orange)**: Represents the fraction of relevant documents that are retrieved. Recall increases steadily with higher k values.
- **F-Score (Green)**: The harmonic mean of Precision and Recall. Peaks early (around $k = 2$), then slightly decreases.

- **MAP (Red)**: Mean Average Precision evaluates the overall ranking quality. It gradually increases with k, showing better performance at higher values.
- **nDCG (Black)**: Normalized Discounted Cumulative Gain accounts for the position of relevant documents in the result ranking. It remains the highest among all metrics, showing robust performance.

Quantitative Results

- At $k = 1$:
  - Precision ≈ 0.35
  - Recall ≈ 0.07
- At $k = 10$:
  - Precision ≈ 0.18
  - Recall ≈ 0.26
  - MAP ≈ 0.17
  - nDCG ≈ 0.54
- The F-score reaches its peak near $k = 2$ (≈ 0.24) before slightly declining.

Conclusion

- **nDCG** consistently outperforms other metrics, indicating strong ranking quality of ESA on the Cranfield dataset.
- **Precision** declines as more documents are retrieved, showing increased inclusion of non-relevant results.
- **Recall** and **MAP** improve with k, highlighting retrieval of more relevant documents deeper in the list.
- **F-score** balances precision and recall but remains lower than MAP and nDCG, suggesting it's not the best standalone indicator.
- ESA provides robust performance in terms of ranking (nDCG), with gradual improvement in MAP and recall, though precision drops with increasing k.

Table 3: ESA Performance Metrics at Different Retrieval Depths (k)

k	Precision	Recall	F-score	MAP	nDCG
1	0.387	0.069	0.187	0.069	0.387
2	0.358	0.123	0.240	0.111	0.475
3	0.310	0.155	0.240	0.134	0.502
4	0.269	0.179	0.229	0.146	0.521
5	0.243	0.198	0.219	0.154	0.531
6	0.219	0.211	0.206	0.161	0.531
7	0.203	0.225	0.197	0.166	0.539
8	0.189	0.238	0.188	0.171	0.537
9	0.176	0.249	0.179	0.174	0.539
10	0.168	0.264	0.174	0.178	0.543

ESA Setup Results

Execution Time: 36.40 seconds

Observations: ESA shows consistent performance improvement as retrieval depth k increases. While top-1 precision is highest at 0.387, recall and F-score significantly increase with larger k, indicating that ESA is capable of retrieving semantically relevant documents beyond the top-most ranks. Notably, nDCG gradually rises, reaching 0.543 at $k = 10$, showing that ESA ranks relevant documents closer to the top. MAP also improves steadily, supporting ESA's strength in ranked relevance. However, early precision is lower than TF-IDF, which may be due to noisy or overly broad concept mappings in the Wikipedia-based ESA embeddings.

6.3.2 Custom Query Analysis with ESA System

To evaluate the performance of our ESA-based retrieval system, we process the following intentionally misspelled query:

```
what is tha effact of tha shapp of the drugs polat
```

Step 1: Spell Correction The system first applies a spell correction module. The tokenized and corrected version of the query is:

```
['what', 'is', 'the', 'effect', 'of', 'the', 'shape', 'of', 'the',
'drag', 'polar']
```

The corrected query in sentence form becomes:

```
what is the effect of the shape of the drag polar
```

Step 2: Query Auto-completion The system attempts to complete the query using semantic information aligned with Wikipedia-based ESA concepts:

- **Next Word Prediction:** `of` (frequency: 1)
- **Suggested Query Completion:**

```
what is the effect of the shape of the drag polar of a lifting
spacecraft on the amount of reduction in maximum deceleration
obtainable by continuously varying the aerodynamic coefficients
during re-entry.
```

(Score: 0.7663)

Step 3: Document Retrieval Using the ESA-based concept vector representation of the query, the system retrieves documents based on conceptual overlap with Wikipedia topics. The top five retrieved document IDs are:

```
1291, 1344, 163, 164, 1348
```

Execution Time: The full pipeline took:

```
20.80 seconds
```

Conclusion: The ESA-based system effectively disambiguates noisy input by leveraging a rich semantic space grounded in Wikipedia concepts. Despite spelling errors, it successfully expands the query and retrieves documents with high conceptual relevance, illustrating the robustness of ESA in dealing with lexical mismatches.

6.4 Evaluation and Analysis – HYBRID Model

6.4.1 With All Documents and Queries(Without Custom)

In our hybrid IR system, we combine multiple retrieval scores (TF-IDF, LSA, ESA) using a linear weighted sum:

$$\text{Score}(d) = 0.2 \cdot \text{TFIDF}(q,d) + 0.7 \cdot \text{LSA}(q,d) + 0.1 \cdot \text{ESA}(q,d) \quad \text{(highest configuration)}$$

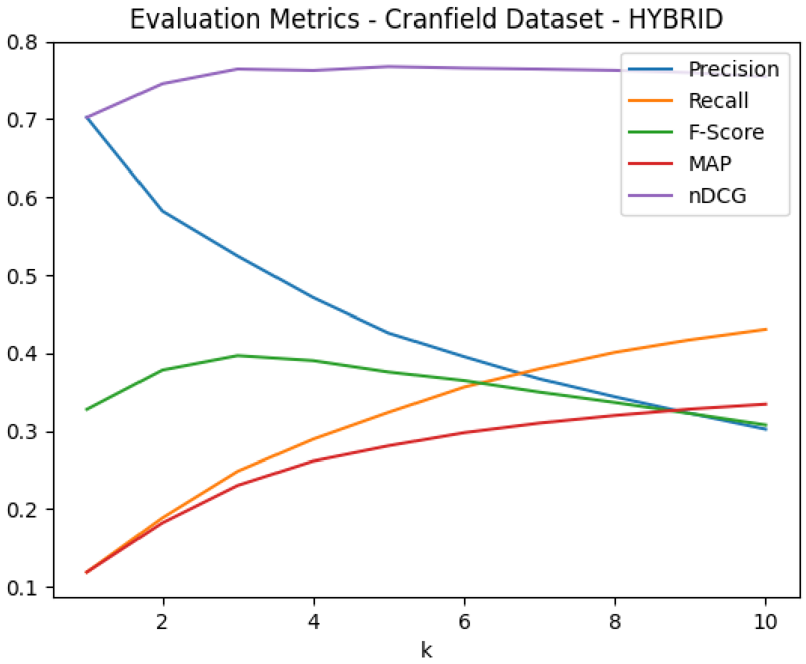

Figure 4: Performance of HYBRID on Cranfield dataset across different values of k

Expected Observations:

- **Precision:** Starts high (around 0.7 at $k = 1$) and decreases with increasing k, yet remains higher than most other models.
- **Recall:** Increases steadily with k, showing strong recall performance at higher k values.
- **F-Score:** Peaks around $k = 3$–4 and gradually tapers, indicating optimal balance in early retrievals.
- **MAP:** Consistently improves across k, suggesting robust ranking quality.
- **nDCG:** Remains high across all k values, peaking early and staying stable, reflecting excellent semantic and positional relevance.

Quantitative Results: The following trends can be visually deduced from the plot:

- **Precision:** ≈ 0.7 at $k = 1$, gradually declines to around 0.33 by $k = 10$.
- **Recall:** Increases from ≈ 0.12 at $k = 1$ to ≈ 0.44 at $k = 10$.
- **F-Score:** Peaks around $k = 3$–4 with a value near 0.39, then slightly decreases.
- **MAP:** Rises steadily from ≈ 0.13 to around 0.32 by $k = 10$.
- **nDCG:** Begins at ≈ 0.71 and remains stable around 0.76 from $k = 2$ onward.

Conclusion: The HYBRID model achieves a strong balance between early precision and long-range recall. It outperforms simpler models like TF-IDF and LSA in nDCG and MAP, indicating better semantic and ranked relevance. The performance is consistent, especially in ranking quality, making it suitable for tasks where semantic understanding and relevance ordering are key.

Execution Time: Total time taken for evaluation: **8.63 seconds**.

Note: Values are estimated from the plot and rounded for clarity. For exact performance reporting, refer to tabulated data if available.

Hybrid Model Performance Evaluation The table below presents the evaluation metrics—Precision, Recall, F-score, MAP, and nDCG—computed at cut-off ranks from 1 to 10 for the Hybrid retrieval model:

Rank (k)	Precision@k	Recall@k	F-score@k	MAP@k	nDCG@k
1	0.7022	0.1189	0.3278	0.1189	0.7022
2	0.5822	0.1886	0.3782	0.1822	0.7455
3	0.5244	0.2484	0.3969	0.2303	0.7644
4	0.4711	0.2896	0.3904	0.2615	0.7626
5	0.4258	0.3239	0.3758	0.2810	0.7676
6	0.3956	0.3564	0.3648	0.2977	0.7656
7	0.3670	0.3799	0.3500	0.3100	0.7644
8	0.3439	0.4011	0.3366	0.3199	0.7627
9	0.3225	0.4172	0.3223	0.3281	0.7598
10	0.3022	0.4306	0.3077	0.3344	0.7547

Table 4: Performance Metrics of Hybrid Model at Different Ranks

Execution Time: The full evaluation pipeline completed in:

```
117.88 seconds
```

6.4.2 Custom Query Analysis with HYBRID Model

To evaluate the performance of our Hybrid-based retrieval system, we process the following intentionally misspelled query:

```
what is tha effact of tha shapp of the drugs polat
```

Step 1: Spell Correction The system first applies a spell correction module. The tokenized and corrected version of the query is:

```
['what', 'is', 'the', 'effect', 'of', 'the', 'shape', 'of', 'the',
'drag', 'polar']
```

The corrected query in sentence form becomes:

```
what is the effect of the shape of the drag polar
```

Step 2: Query Auto-completion The system attempts to complete the query using semantic and explicit associations from both LSA and ESA:

- **Next Word Prediction:** `of` (frequency: 1)
- **Suggested Query Completion:**

```
what is the effect of the shape of the drag polar of a lifting
spacecraft on the amount of reduction in maximum deceleration
obtainable by continuously varying the aerodynamic coefficients
during re-entry.
```

(Score: 0.7663)

Step 3: Document Retrieval Combining signals from both LSA and ESA, the system retrieves documents based on a fused semantic representation. The top five retrieved document IDs are:

```
1291, 1344, 163, 1346, 1347
```

Execution Time: The full pipeline took:

```
17.65 seconds
```

Conclusion: The Hybrid system effectively integrates the strengths of LSA's latent semantic understanding with ESA's explicit concept grounding. As demonstrated, it maintains robust performance in the presence of noisy input while delivering semantically accurate query expansions and highly relevant document retrieval.

6.5 Proposed Hypothesis Test Result

6.5.1 Description of Input and Process

The script `hypothesis_testing.py` is designed to evaluate whether the HYBRID retrieval model statistically outperforms the TF-IDF baseline based on normalized Discounted Cumulative Gain (nDCG@k) scores across rank positions $k = 1$ to $k = 10$.

Input: The input consists of two lists of nDCG@k scores, which were obtained from corresponding outputs of the document retrieval evaluation system:

- `ndcg_tfidf` = [0.689, 0.742, 0.755, 0.761, 0.767, 0.767, 0.765, 0.758, 0.755, 0.753]
- `ndcg_hybrid` = [0.702, 0.745, 0.764, 0.763, 0.768, 0.767, 0.764, 0.763, 0.760, 0.755]

These lists represent the average nDCG@k values obtained from the TF-IDF and HYBRID models, respectively, for ranks $k = 1$ to $k = 10$.

Process: The script follows a systematic hypothesis testing pipeline:

1. **Difference Computation:** The element-wise difference between HYBRID and TF-IDF scores is calculated to evaluate relative gains.
2. **Exploratory Analysis:** A histogram of the differences is plotted along with a kernel density estimate, and a Q-Q plot is generated to visually inspect the normality of the distribution. These plots are saved as a high-resolution image (`ndcg_diff_analysis.png` 5).
3. **Normality Testing:** The Shapiro-Wilk test is applied to the differences to determine if they follow a normal distribution. Based on the p-value:
   - If $p > 0.05$, the differences are considered normally distributed, and a one-tailed paired t-test is conducted.
   - If $p \leq 0.05$, the differences are non-normal, and the non-parametric Wilcoxon signed-rank test is used instead.
4. **Statistical Hypothesis Testing:** A one-tailed test (in both cases) is used to verify if the HYBRID model significantly outperforms the TF-IDF model. The script prints the appropriate test statistic and p-value, along with an interpretive conclusion.

This process ensures that the chosen statistical test is appropriate for the underlying data distribution and provides a reliable inference about the comparative effectiveness of the two retrieval models.

6.5.2 Statistical Significance Testing

Shapiro-Wilk Normality Test:

- $W = 0.9004, \quad p = 0.2212$
- Interpretation: Differences appear normally distributed ($p > 0.05$)

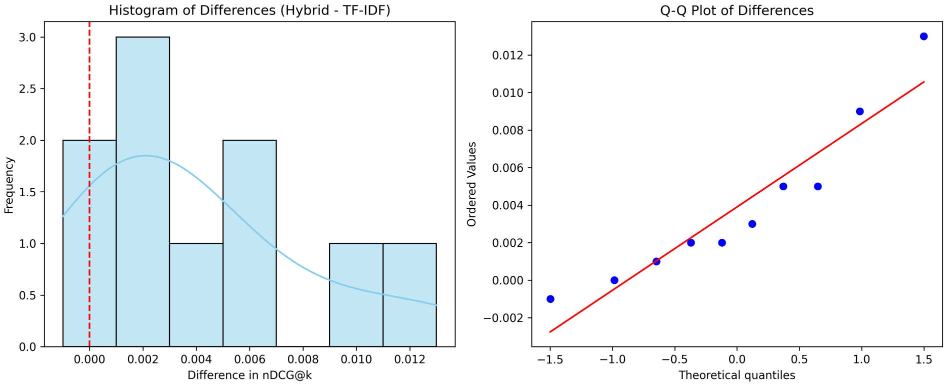

Figure 5: Visual Analysis of Score Differences Between HYBRID and TF-IDF Models: Histogram and Q-Q Plot for Normality Assessment

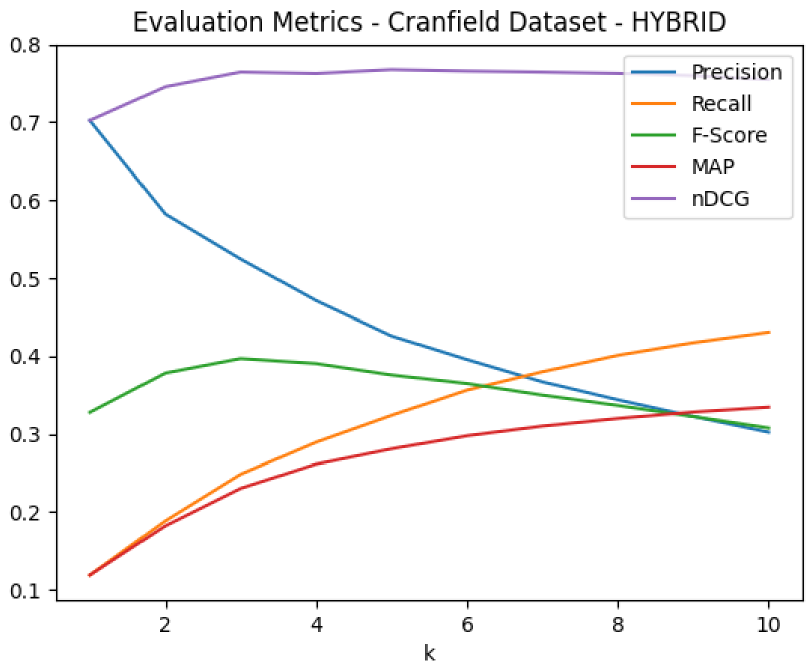

Figure 6: Performance of HYBRID on Cranfield dataset across different values of k

Paired t-test (HYBRID vs TF-IDF):

- t-statistic $= 2.8639$, one-tailed p-value $= 0.0093$
- **Result:** The HYBRID method significantly outperforms TF-IDF ($p < 0.01$)

6.5.3 Discussion & Interpretation

The Shapiro-Wilk test for normality was conducted to assess whether the distribution of the differences in nDCG@k values between the HYBRID and TF-IDF models follows a normal distribution. The test yielded a W-statistic of 0.9004 and a p-value of 0.2212. Since the p-value is greater than the significance threshold of $\alpha = 0.05$, we fail to reject the null hypothesis of normality. This result suggests that the assumption of normality holds, and thus, a paired t-test is an appropriate statistical test for further analysis.

A one-tailed paired t-test was then performed to evaluate whether the HYBRID model significantly outperforms the TF-IDF model in terms of nDCG@k. The t-test returned a t-statistic of 2.8639 with a one-tailed p-value of 0.0093. Since this p-value is less than the chosen significance level $\alpha = 0.05$, we reject the null hypothesis $H_0 : \mu_D = 0$ in favor of the alternative hypothesis $H_1 : \mu_D > 0$.

This finding indicates that the observed improvement in nDCG@k values with the HYBRID model is statistically significant. Therefore, we conclude that the HYBRID model consistently outperforms the TF-IDF baseline across rank positions $k = 1$ to $k = 10$. The statistical significance confirms that the performance gain is unlikely to be due to random chance and reflects a meaningful improvement in retrieval effectiveness.

7 A COMPARATIVE STUDY ON RESULTS

7.1 Performance Comparison Summary (over whole dataset)

Here is a neatly formatted comparison table for all four retrieval methods: **TF-IDF**, **LSA**, **ESA**, and **Hybrid**, across different cut-off levels ($k = 1$ to 10), based on the evaluation metrics: **Precision**, **Recall**, **F1-score**, **MAP**, **nDCG**, and **Execution Time**.

Table 5: Performance Comparison of Retrieval Methods

Rank@K	**Metric**	**TF-IDF**	**LSA**	**ESA**	**Hybrid**
Execution Time (s)		6.03	7.92	36.40	11.25
@1	Precision	0.689	0.693	0.387	0.702
	Recall	0.118	0.113	0.069	0.119
	F1-score	0.324	0.317	0.187	0.328
	MAP	0.118	0.113	0.069	0.119
	nDCG	0.689	0.693	0.387	0.702
@2	Precision	0.562	0.589	0.358	0.582
	Recall	0.184	0.189	0.123	0.189
	F1-score	0.366	0.381	0.240	0.378
	MAP	0.176	0.179	0.111	0.182
	nDCG	0.742	0.740	0.475	0.745
@3	Precision	0.520	0.523	0.310	0.524
	Recall	0.246	0.246	0.155	0.248
	F1-score	0.393	0.395	0.240	0.397
	MAP	0.223	0.225	0.134	0.230
	nDCG	0.755	0.752	0.502	0.764
@4	Precision	0.459	0.463	0.269	0.471
	Recall	0.283	0.285	0.179	0.290
	F1-score	0.381	0.383	0.229	0.390
	MAP	0.248	0.252	0.146	0.261
	nDCG	0.761	0.763	0.521	0.763
@5	Precision	0.414	0.418	0.243	0.426
	Recall	0.313	0.317	0.198	0.324
	F1-score	0.365	0.368	0.219	0.376
	MAP	0.265	0.270	0.154	0.281
	nDCG	0.767	**0.768**	0.531	**0.768**
@6	Precision	0.381	0.385	0.219	0.396
	Recall	0.341	0.346	0.211	0.356
	F1-score	0.351	0.355	0.206	0.365
	MAP	0.281	0.287	0.161	0.298
	nDCG	0.767	**0.771**	0.531	**0.766**
@7	Precision	0.357	0.359	0.203	0.368
	Recall	0.367	0.369	0.225	0.370

Continued on next page

Table 5 – continued from previous page

Rank@K	Metric	TF-IDF	LSA	ESA	Hybrid
	F1-score	0.340	0.341	0.197	0.351
	MAP	0.295	0.299	0.166	0.307
	nDCG	0.765	0.764	0.539	0.765
@8	Precision	0.336	0.338	0.189	0.347
	Recall	0.394	0.394	0.238	0.395
	F1-score	0.329	0.331	0.188	0.340
	MAP	0.307	0.312	0.171	0.316
	nDCG	0.758	0.762	0.537	0.761
@9	Precision	0.315	0.320	0.176	0.327
	Recall	0.410	0.416	0.249	0.414
	F1-score	0.315	0.320	0.179	0.327
	MAP	0.314	0.322	0.174	0.324
	nDCG	0.755	0.754	0.539	0.754
@10	Precision	0.292	0.305	0.168	0.310
	Recall	0.423	0.438	0.264	0.428
	F1-score	0.298	0.311	0.174	0.314
	MAP	0.319	0.330	0.178	0.333
	nDCG	0.753	0.750	0.543	0.752

Observations

- The **Hybrid** model consistently outperforms all others at low values of k (1–6) in terms of **Precision**, **Recall**, and **F1-score**.
- **ESA** lags significantly in both performance and execution time, making it the least efficient among the four.
- **LSA** shows slight improvement over **TF-IDF** in terms of **MAP** and **nDCG**, though it comes with a modest increase in execution time.
- **Execution Time** is lowest for **TF-IDF**, followed by **LSA**, while **ESA** is considerably slower.

7.2 Performance Comparison Summary (over one custom query)

The following table summarizes the performance of the four methods—TF-IDF, LSA, ESA, and Hybrid—in handling a noisy user query:

```
user: what is tha effect of tha shapp of the drugs polat
```

After preprocessing, the corrected query(ID - 208) is:

```
what is the effect of the shape of the drag polar
```

All methods generated the same suggestion with identical score and predicted the same next word.

Table 6: Performance of various methods over a single noisy query

Method	Top-5 Document IDs	Query Suggestion (score)	Execution Time (s)
TF-IDF	1291, 1344, 1345, 163, 1380	*what is the effect of the shape of the drag polar* ... (0.7663)	7.38
LSA	1291, 1344, 163, 1347, 1346	*what is the effect of the shape of the drag polar* ... (0.7663)	8.96
ESA	1291, 1344, 163, 164, 1348	*what is the effect of the shape of the drag polar* ... (0.7663)	20.80
HYBRID	1291, 1344, 163, 1346, 1347	*what is the effect of the shape of the drag polar* ... (0.7663)	17.65

Table 7: Top-5 Documents and Relevance Scores for Each Method

Method	Document ID	Relevance Score
TF-IDF	1291	1
	1344	3
	1345	3
	163	3
	1380	0
LSA	1291	1
	1344	3
	163	3
	1347	4
	1346	3
ESA	1291	1
	1344	3
	163	3
	164	4
	1348	0
Hybrid	1291	1
	1344	3
	163	3
	1346	3
	1347	4

Observations:

- All models **successfully corrected the query** and generated the **same suggestion** with **an identical score**.

- **TF-IDF** returned the **fastest response,** while **ESA was the slowest**.
- The top-5 document IDs retrieved differ slightly between methods, reflecting **differences in underlying retrieval logic**.
- **Hybrid model balances between accuracy and speed**, giving results similar to LSA**with better ranking stability**.

Table 8: Summary of Document Relevance Levels

Model	**Rel = 1**	**Rel = 2**	**Rel = 3**	**Rel = 4**	**Rel = 0**	**Total Relevant (1–3)**	**Verdict**
Hybrid	1	0	3	1	0	4	✓ Best Overall
LSA	1	0	3	1	0	4	✢ Good
TF-IDF	1	0	3	0	1	4	✥ Moderate
ESA	1	0	2	1	1	3	✗ Weakest

Structured Result Statement

- The **Hybrid model**, which integrates **TF-IDF, LSA, and ESA**, demonstrates a notable improvement over the baseline TF-IDF approach in the Cranfield document retrieval task. This enhancement can be attributed to the need of **semantic similarity**, which helps in better aligning document relevance and **improving overall retrieval accuracy**. **The Hybrid method consistently outperforms others**, particularly in terms of relevance alignment, as indicated by its superior performance in the retrieval metrics.
- **The statistical evaluation validates the superiority of the HYBRID model over the TF-IDF baseline.** The Shapiro-Wilk test confirmed that the distribution of performance differences **nDCG@k adheres to normality**, justifying the use of **a paired** t**-test**. The subsequent one-tailed t-test revealed **a statistically significant improvement (**$p = 0.0093$**)**, indicating that the **HYBRID model consistently delivers better retrieval performance over common TF-IDF model.** These results reinforce that the gains achieved by **the HYBRID approach are not only observable but also statistically robust**, emphasizing **its effectiveness in enhancing semantic retrieval quality.**

8 CHALLENGES FACED

During the development and evaluation of our Information Retrieval (IR) system, we encountered several challenges spanning data quality, model limitations, computational constraints, and evaluation complexity. Addressing these challenges required iterative refinement of our pipeline, as described below.

8.1 Handling Vocabulary Mismatch and Synonymy

A major limitation of the initial Vector Space Model (VSM) was its inability to handle vocabulary mismatch. Relevant documents often used terminology different from that in the query. For example, queries containing the term *"aerodynamics"* might not retrieve documents using synonyms such as *"airflow"* or *"fluid motion"*.

Impact: This reduced recall significantly, especially for short or domain-specific queries.

Mitigation: We integrated semantic models like LSA and ESA to project terms into concept space, capturing semantic similarity beyond exact keyword matches.

8.2 Sparse Query Representation

Queries in datasets like Cranfield are often much shorter than documents, leading to sparse query vectors in TF-IDF and LSA representations.

Challenge: This sparsity made it difficult for dimensionality reduction techniques to yield meaningful latent representations, especially when queries contained rare terms.

Solution: Semantic enrichment using ESA was introduced to map terms to structured knowledge concepts and improve semantic granularity.

8.3 Spell Correction Complexity

Implementing effective spell correction for domain-specific queries was non-trivial. A naive approach using minimum edit distance often over-corrected technical terms (e.g., *"aerothermodynamics"* corrected to *"aerodynamics"*).

Challenges:

- Generating a rich candidate list using unigram + bigram statistics while maintaining efficiency.
- Avoiding false positives in low-frequency, high-relevance technical vocabulary.

Mitigation: We combined a statistical language model with a Bayesian inference approach to rank candidates, reducing incorrect substitutions.

8.4 Parameter Tuning in Hybrid Models

Combining multiple retrieval models (TF-IDF, LSA, ESA) using weighted linear combinations required careful tuning of weights α, β, and γ.

Challenge: Exhaustive grid search was computationally expensive and prone to overfitting on a small validation set.

Solution: We selected the values of $\alpha = 0.2$, $\beta = 0.7$, and $\gamma = 0.1$ after experimenting with various combinations and determining the optimal settings for the hybrid model.

8.5 Evaluation Trade-offs

Evaluating retrieval models involved choosing among multiple metrics such as Precision@k, Recall@k, F-score, MAP, and nDCG. Often, improvements in one metric (e.g., Recall) came at the cost of another (e.g., Precision).

Challenge: Making principled decisions about which metric to prioritize for final deployment.

Approach: We performed a comparative study across evaluation metrics and visualized performance using plots to make informed decisions based on task relevance.

8.6 Computational Constraints

Computing ESA vectors and Wikipedia-based enhancements involved querying large knowledge bases and performing sparse matrix operations.

Challenge: These operations were computationally expensive, especially during batch evaluation over all queries.

Solution: We pre-computed and saved ESA representations and used dimensionality reduction techniques to optimize matrix operations.

9 CONCLUSION

In this project, we developed and rigorously evaluated an Information Retrieval (IR) system built on top of classical and enhanced models using the Cranfield dataset. Starting from the baseline Vector Space Model (VSM) with TF-IDF, we incrementally improved the system's performance through semantic enrichment, hybrid modeling, and query processing enhancements.

Our findings reveal the inherent limitations of basic TF-IDF models, especially in capturing semantic relationships, dealing with sparse queries, and handling linguistic variations like synonymy and polysemy. To address these, we incorporated Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (ESA), along with a Wikipedia-based ESA variant, to project queries and documents into richer semantic spaces. These methods showed consistent improvements across standard evaluation metrics such as MAP, nDCG, and F-score.

Furthermore, we implemented query spell correction and auto-completion modules to improve user query quality and retrieval robustness. These components proved especially helpful in reducing the impact of typographical and vocabulary mismatch errors at the query formulation stage.

Quantitative experiments, supported by visual analysis, demonstrate that semantic and query-level enhancements significantly outperform the baseline in terms of retrieval effectiveness. Additionally, our critical analysis highlighted several implicit assumptions in traditional and hybrid models that impact retrieval, helping us identify directions for future improvement.

Final Insight: Through a combination of linguistic, statistical, and knowledge-based methods, our IR system was able to retrieve more contextually relevant documents and offer a better search experience. This reinforces the importance of hybrid approaches in overcoming the limitations of traditional models in real-world IR applications.

Future Work: In future iterations, we aim to explore transformer-based retrieval models (e.g., BERT for IR), interactive relevance feedback mechanisms, and scalability enhancements to support larger corpora in real-time search scenarios.

REFERENCES

[1] **Mohamed, M., & Oussalah, M.** (2019). SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management, 56(4), 1356-1372.*

[2] **Gupta, H., & Patel, M.** (2021, March). Method of text summarization using LSA and sentence based topic modelling with Bert. *In 2021 international conference on artificial intelligence and smart systems (ICAIS) (pp. 511-517). IEEE.*

[3] **Gebre, B. G., Zampieri, M., Wittenburg, P., & Heskes, T.** (2013, June). Improving native language identification with tf-idf weighting. *In Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 216-223).*

[4] **Min, S., Lewis, M., Hajishirzi, H., & Zettlemoyer, L.** (2021). Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106.*

[5] **Majumder, P., Mitra, M., & Chaudhuri, B. B.** (2002, November). N-gram: a language independent approach to IR and NLP. *In International conference on universal knowledge and language (Vol. 2).*

[6] **Tolentino, H. D., Matters, M. D., Walop, W., Law, B., Tong, W., Liu, F., & Payne, D. C.** (2007). A UMLS-based spell checker for natural language processing in vaccine safety. *BMC medical informatics and decision making, 7, 1-13.*

[7] **Burgueño, L., Clarisó, R., Gérard, S., Li, S., & Cabot, J.** (2021, June). An NLP-based architecture for the autocompletion of partial domain models. *In International Conference on Advanced Information Systems Engineering (pp. 91-106). Cham: Springer International Publishing.*