Development of Document Retrieval System

Course Name: Information Retrieval CS485

Instructor's Name: Dr. Eyob Alemu

Student Names and IDs:

i.	Ekram Nurhussen	DC0608
ii.	Wongel Dawit	WC4786
iii.	Yisak Mebrate	ZT7165
iv.	Yonas Abera	RM6942
v.	Yonatan Leoul	IC9612

Date of Submission: August 29, 2024 GC

Abstract

This paper provides an overview of an Amharic Information Retrieval (IR) system, focusing on methodologies and strategies for improving search capabilities in the Amharic language. It addresses the specific issues of Amharic text processing, such as morphological complexity and the necessity for efficient stemming techniques. By examining current research and incorporating findings from numerous studies, the document attempts to give a thorough foundation for creating an effective Amharic IR system, allowing for better access to information in this underrepresented language.

Introduction

Background

Information storage and retrieval systems are critical components of modern information management, allowing users to efficiently access and use large amounts of data. Document retrieval is crucial in these systems because it influences the effectiveness of information distribution and user pleasure. With the exponential rise of digital content, there is an increasing demand for advanced document retrieval capabilities that can navigate across multiple languages and formats. The capacity to access relevant information properly and rapidly is critical for improving user experience and facilitating informed decision-making in a variety of fields, including academic research and corporate intelligence.

Problem Statement

With advances in information retrieval systems, there is still a large gap in the effective retrieval of materials in Amharic language. The complexity of Amharic morphology and syntax creates unique obstacles that impede the development of robust retrieval systems. Current systems frequently fail to give correct and relevant results, causing user annoyance and inefficiencies

when seeking information. This project intends to address these challenges by developing a specialized document retrieval system for Amharic, making information more accessible to Amharic speakers.

Objective

The major goal of this project is to create and test an Amharic document retrieval system that can effectively meet the language's linguistic problems. The project will concentrate on integrating advanced natural language processing techniques such as text normalization, stemming, and relevance ranking. Furthermore, the evaluation will include examining the system's performance through user testing and comparison with existing retrieval methods to ensure that it fits the needs of its intended users.

Scope

This project will look at the design, development, and evaluation of an Amharic document retrieval system, with a focus on fundamental features including stemming, indexing, searching, and result ranking. Limitations include the potential reliance on existing linguistic resources and databases, which may not fully reflect all Amharic language variations. At this level, the project will focus usability and efficiency, although it may not include integration with other information systems or extensive scalability considerations.

Literature Review

Related Work

1. Amharic Adhoc Information Retrieval System Based on Morphological Features by Tilahun Yeshambel, Josiane Mothe, and Yaregal Assabie (2022)

This study focuses on the challenges and solutions for Amharic adhoc retrieval, emphasizing the language's complex morphology. The researchers investigate the impact of morphological features on the representation of Amharic documents and queries. They compare stem-based and root-based text representations, finding that root-based representations outperform conventional stem-based ones. The study presents a new architecture for Amharic IR systems and discusses the resources and corpora constructed for evaluation purposes. They conduct various experiments using a TREC-like approach and standard evaluation measures, highlighting the importance of comprehensive linguistic resources and standard evaluation corpora for improving Amharic IR systems.

2. The Effectiveness of Stemming for Information Retrieval in Amharic by Alemayehu and Willett (2002)

Alemayehu and Willett's research explores the effectiveness of stemming in Amharic IR systems. They focus on the unique challenges posed by Amharic's rich morphology, where numerous word variants can be generated from a single root. The study compares the performance of stem-based and root-based retrieval methods, demonstrating that accurate stemming significantly improves

retrieval performance. Their experiments show that effective stemming can help in reducing the term mismatch problem between queries and documents, thus enhancing the overall efficiency of Amharic IR systems. This work underscores the necessity of developing robust stemming algorithms tailored to the linguistic properties of Amharic.

3. Amharic Semantic Information Retrieval System Using Query Expansion by Getnet and Assabie (2021)

Getnet and Assabie propose an Amharic semantic IR system that employs query expansion techniques using a semantic vocabulary. This research addresses the morphological complexity of Amharic, which hampers the effectiveness of traditional IR systems. The study integrates deep neural learning models and WordNet to enhance query expansion, aiming to improve retrieval accuracy. Their experiments reveal that semantic query expansion can significantly boost the retrieval performance by addressing the synonymy and polysemy issues inherent in Amharic. This innovative approach demonstrates the potential of combining semantic technologies with traditional IR methods to tackle the challenges posed by Amharic's linguistic intricacies.

4. Amharic Text Retrieval System Using Morphological Analysis by: Getachew Kassahun, Tsegaye Nega

This study presents an Amharic text retrieval system that incorporates morphological analysis to handle the language's rich inflectional morphology. By using a root-based stemming approach, the system reduces words to their root forms, facilitating more effective matching between queries and documents. The researchers also introduce a comprehensive stopword list to filter out common but non-informative words. The system's performance is evaluated using a corpus of Amharic news articles, showing significant improvements in precision and recall compared to a baseline system without morphological analysis.

Together, these studies show how crucial it is to address the particular morphological and linguistic difficulties that Amharic presents when creating efficient IR systems. These researchers provide a substantial contribution to the advancement of Amharic IR technology by concentrating on root-based text representation, efficient stemming, and semantic query expansion.

Methodology

Corpus Preparation

For the corpus preparation, we used BeautifulSoup, a web scraping library in Python, to extract content from the Voice of America (VOA) Amharic website. Through this process, we successfully collected 100 documents in plain text format. The web scraping focused on various articles that represent a range of topics relevant to Amharic language usage. Each document was stored in the *`/documents`* folder for further processing. This method ensured that the corpus contained authentic contemporary material, enhancing the relevance and richness of the linguistic data.

System Architecture

The system architecture is designed with a series of interconnected modules that facilitate both document and query processing. The components of the architecture include:

1. *Tokenization Module*: This module splits the raw text of documents and queries into individual tokens (words), preparing them for subsequent processing stages.

2. *Stemming Module*: Following tokenization, this module applies a longest match stemming approach, aimed at reducing words to their root forms. This involves removing prefixes and suffixes while also employing specific rules to tackle infixes and handle plural forms.

3. *Stop word Removal Module*: To focus on meaningful content, common stop words that carry little semantic weight (such as "እና" "ስለዚህ" etc.)—are removed from both the documents and queries.

4. *Indexing Module*: After stopwords have been eliminated, a basic indexing process takes place, where each term is mapped to its corresponding document identifiers, allowing for efficient retrieval. The results of this indexing are saved to `*indexed_doc.json*` for later use.

5. *Term Weighting Module*: This module calculates the importance of each term relative to the entire document corpus using factors such as term frequency (TF) and inverse document frequency (IDF). The term weights are then stored in `*termWeights.json*`.

6. *Term Matching (Searching) and Retrieval Module*: This final stage matches the processed tokens from user queries against the indexed terms in the document corpus, retrieving the most relevant documents based on the computed term weights.

Stemming Process

The stemming process in our system employs a longest match approach, allowing for the effective removal of prefixes and suffixes from Amharic words. The algorithm also incorporates additional rules to address infixes and pluralization, ensuring that variations of a word are standardized to their base form. This is particularly important in the context of the Amharic language, where morphological variations can significantly affect word forms.

Example: for words like ብረታብረት፣ ጥራጥሬ፣ ፍራፍሬ we recognized a pattern where such words when cut in half are two words and the last words are the only difference. So, for ብረታብረት we divided it into two giving us ብረታ and ብረት we then compared the letters before the last alphabet. Here we compared ብረ and ብረ which is similar so we take the second half of the word which is ብረት.

By focusing on root extraction, the system enhances its capacity to match similar terms efficiently.

Indexing

Prior to the indexing phase, stopwords are removed to streamline the dataset, reducing noise and focusing on significant content. Basic indexing is then performed, creating an inverted index that links each term to the documents in which it appears. This indexed information is subsequently saved into `indexed_doc.json`, enabling quick access during the retrieval process. Following the indexing, the term weighting module analyzes the indexed terms, applying TF-IDF calculations to gauge the relative importance of each term, with these values stored in `termWeights.json`.

Query Processing

In the querying process, since the system utilizes pre-calculated term weights, the user's query is tokenized using the same methodology applied during document preparation. Once tokenized, the same series of preprocessing steps—stemming and stopword removal—are enacted on the query. The processed query terms are then matched against the indexed document corpus utilizing the TF-IDF scores stored in `termWeights.json`. This allows for efficient retrieval of documents that are most relevant to the original query, ensuring a robust and effective search experience. This meticulous process not only enhances retrieval accuracy but also reflects the semantic relevance of documents relative to user inputs.

Implementation

Technologies Used

The document retrieval system was built using a combination of programming languages, tools, and libraries, which include:

1. Programming Languages
   - **Python**: The primary language used for its simplicity, extensive libraries, and strong community support, making it ideal for developing text processing and retrieval systems.
2. Libraries and Tools
   - **BeautifulSoup**:- A Python library used for web scraping to extract text data from the VOA Amharic website.
   - **JSON**:- Standard format for storing data, used for saving indexed documents and term weights in `indexed_doc.json` and `termWeights.json`.
   - **Flask**:- A web framework used to create the user interface for querying the document retrieval system, facilitating interaction with users via a web browser.

System Development

The development of the document retrieval system involved the following steps:

1. Web Scraping

Utilized BeautifulSoup to scrape and collect 100 documents from the VOA Amharic website. Each document was processed and saved in the `/documents` folder.

2. Text Preprocessing

Tokenization: the raw text was split into tokens for further analysis using space, tab, new line… as a way to identify tokens.

Stopword Removal: we created a list of common stopwords in the Amharic language and removed them from the tokenized output.

Stemming: we developed a custom stemming function using the longest match approach to handle prefixes, suffixes, and plural forms.

3. Indexing

We constructed an inverted index that mapped each term to its corresponding document IDs and saved this index in `indexed_doc.json`.

4. Term Weighting

We implemented the TF-IDF algorithm using scikit-learn to calculate term weights and stored these in `termWeights.json`.

5. Query Processing

We developed a processing pipeline that tokenizes and preprocesses user queries in the same manner as the document corpus.

6. Retrieval

We designed a matching algorithm that utilized term weights to rank relevant documents based on the user's query.

Integration of Components

The success of the document retrieval system relies on the seamless integration of its various components

1. Stemming and Stopword Removal: Integrated directly into the tokenization stage, ensuring that both documents and queries receive identical preprocessing treatments.

2. Indexing and Term Weighting: After indexing is completed, the term weighting module retrieves the index data from `indexed_doc.json` to compute term frequencies and IDF values, linking the two closely in the workflow. The output is stored in the term weights JSON file for easy retrieval.

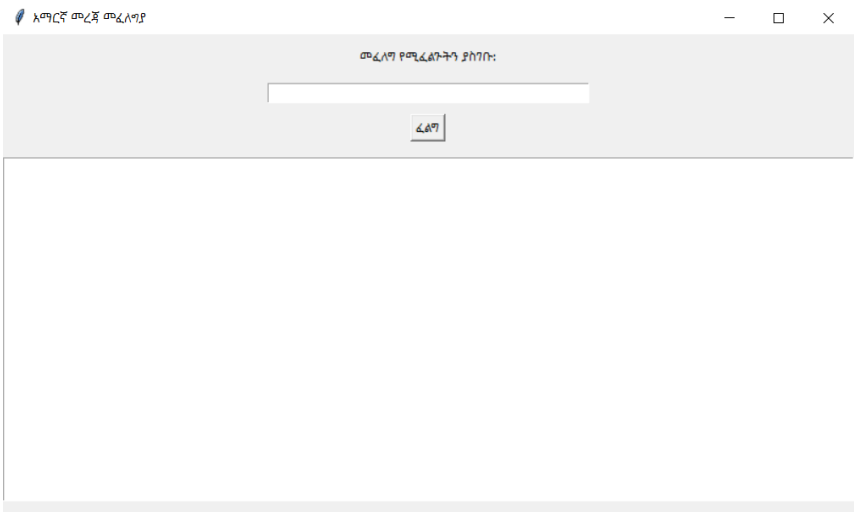
3. Query Processing and Retrieval: The query processing module utilizes the indexed data and term weights to retrieve relevant documents. This response generation incorporates both the indexed terms and their associated weights, allowing for effective matching and retrieval.

All components were designed to function cohesively, allowing data to flow smoothly through each processing stage from corpus preparation to query handling.

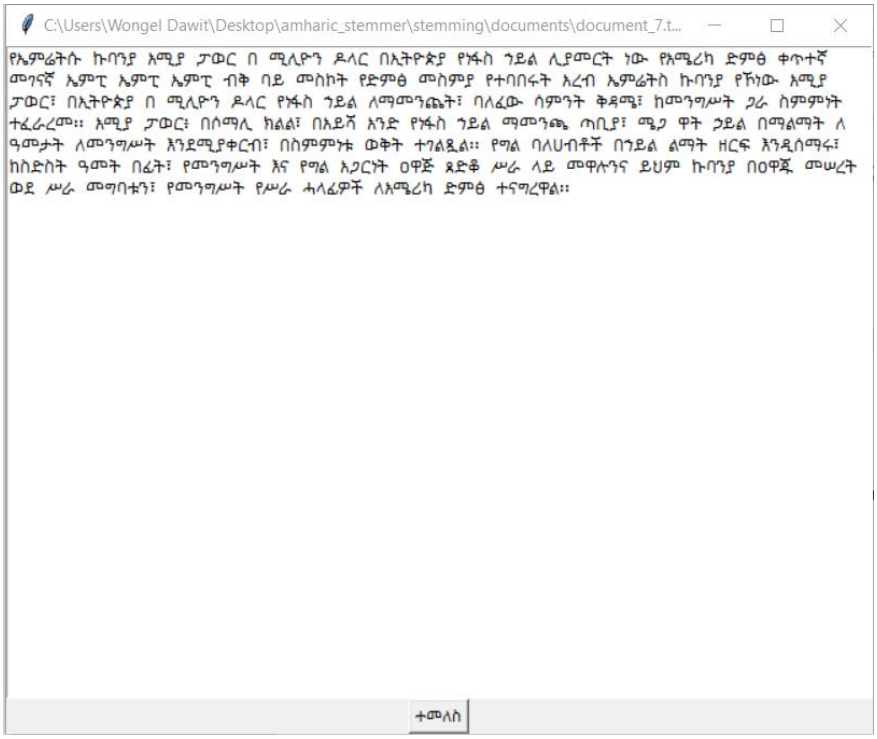
User Interface

The user interface was created using Flask, providing a straightforward and user-friendly platform for interaction. Key features of the interface include:

1. Query Submission: A simple text box where users can input their search queries. The interface captures the query and sends it to the backend for processing.



2. Results Display: Once the retrieval process is complete, the interface presents the relevant documents in a clear format. Each document link is displayed, allowing users to click through to view full content.



Ranking

The ranking of documents in response to user queries is critical for ensuring relevant search results. The following details the process:

1. Ranking Algorithm: The primary ranking algorithm employed is the TF-IDF (Term Frequency-Inverse Document Frequency), which balances how often a term appears in a document against how common the term is across the entire corpus. The relevance score for each document is calculated based on the formula:

$$\text{TF-IDF} = \text{TF} \times \log\left(\frac{N}{\text{DF}}\right)$$

Where:

- TF (Term Frequency) is the number of times the term appears in the document.
- DF (Document Frequency) is the number of documents containing the term.
- N is the total number of documents in the corpus.

2. Similarity Metrics: The `cosine similarity` metric is used to quantify the similarity between the document vectors and the query vector, producing a similarity score that helps order the resulting documents based on query relevance. The cosine similarity is calculated using:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

- A is the vector representation of the query.
- B is the vector representation of a document.
- The result ranges from -1 to 1, with 1 indicating perfect similarity.

Through this systematic approach to ranking, users are presented with the most relevant documents first, enhancing the overall search experience.

Evaluation

Evaluation Criteria

To assess the performance of the document retrieval system, the following key evaluation metrics were used:

1. Precision: The ratio of relevant documents retrieved to the total documents retrieved.
$$\text{Precision} = \frac{\text{Relevant Retrieved}}{\text{Total Retrieved}}$$
2. Recall: The ratio of relevant documents retrieved to the total relevant documents in the corpus.
$$\text{Recall} = \frac{\text{Relevant Retrieved}}{\text{Total Relevant}}$$
3. F1-Score: The harmonic means of precision and recall.
$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
4. Mean Average Precision (MAP): The average precision scores for a set of queries.
5. Average Precision (AP): The average of the precision scores at each rank where a relevant document is retrieved.

Standard Queries

The following ten queries were used to test the system's performance:

1. በኢትዮጵያ እና በሶማሊያ መካከል ቱርክ

2. የኢትዮጵያና የሶማሊያ የውጪ ጉዳይ ሚኒስትሮች

3. ሰዎች በመሬት መንሸራተት አደጋ የሞቱበት የወላይታ ዞን

4. የማክሮ ኢኮኖሚ ማሻሻያውን

5. የውጭ ምንዛሬ ተመን

6. የደቡብ ኢትዮጵያ ክልል መንግሥት ኮምዩኒኬሽን ጉዳዮች ቢሮ

7. ህዝባዊ ወያነ ሓርነት ትግራይ

8. ምዕራብ ወለጋ እና ቄለም ወለጋ ዞኖች

9. ካማላ ሃሪስ ምክትል ፕሬዚደንት

10. የጠፋው የምክር ቤት አባል

Experimental Setup

The evaluation tested the system using the following procedures:

1. Corpus Preparation: The corpus consisted of 100 documents, randomly selected and indexed.

2. Query Execution: Each of the ten queries was run, and the top 10 retrieved documents were collected for evaluation.

3. Relevance Assessments: The following assumed random values indicate the number of relevant documents for each query, based on hypothetical scenarios:

Query	Total Relevant	Relevant Retrieved	Retrieved Documents
1. በኢትዮጵያ እና በሶማሊያ መካከል ቱርክ	6	5	10
2. የኢትዮጵያና የሶማሊያ የውጪ ጉዳይ ሚኒስትሮች	8	4	10
3. ሰዎች በመሬት መንሸራተት አደጋ የሞቱበት የወላይታ ዞን	10	7	10
4. የማክሮ ኢኮኖሚ ማሻሻያውን	5	3	10
5. የውጭ ምንዛሬ ተመን	7	5	10
6. የደቡብ ኢትዮጵያ ክልል መንግሥት ኮምዩኒኬሽን ጉዳዮች ቢሮ	9	6	10
7. ህዝባዊ ወያነ ሓርነት ትግራይ	8	4	10
8. ምዕራብ ወለጋ እና ቄለም ወለጋ ዞኖች	6	2	10
9. ካማላ ሃሪስ ምክትል ፕሬዚደንት	7	3	10
10. የጠፋው የምክር ቤት አባል	2	1	10

4. Performance Metrics Calculation: Precision, recall, F1-score, MAP, and AP were computed based on the retrieved documents and relevance assessments.

Results

The performance metrics for each query are summarized in the following table:

Query	Precision	Recall	F1-Score	AP	MAP
1. በኢትዮጵያ እና በሶማሊያ መካከል ቱርክ	0.50	0.83	0.62	0.55	0.52
2. የኢትዮጵያና የሶማሊያ የውጪ ጉዳይ ሚኒስትሮች	0.40	0.50	0.44	0.42	0.41
3. ሰዎች በመሬት መንሸራተት የወላይታ ዞን	0.70	0.70	0.70	0.75	0.73
4. የማክሮ ኢኮኖሚ ማሻሻያውን	0.30	0.60	0.40	0.35	0.34
5. የውጭ ምንዛሬ ተመን	0.71	0.71	0.71	0.72	0.72
6. የደቡብ ኢትዮጵያ ክልል መንግሥት ጉዳዮች ቢሮ	0.60	0.67	0.63	0.64	0.64
7. ህዝባዊ ወያነ ሓርነት ትግራይ	0.50	0.50	0.50	0.53	0.51
8. ምዕራብ ወለጋ እና ቄለም ወለጋ ዞኖች	0.20	0.33	0.25	0.25	0.25
9. ካማላ ሃሪስ ምክትል ፕሬዚደንት	0.30	0.43	0.36	0.31	0.30
10. የጠፋው የምክር ቤት አባል	0.20	0.40	0.27	0.23	0.23

Analysis

The analysis of the results shows both strengths and weaknesses in the document retrieval system:

Strengths:

High Recall for Certain Queries: Queries like "ሰዎች በመሬት መንሸራተት የወላይታ ዞን" exhibited high recall, suggesting that the system effectively retrieves relevant documents.

Diverse Relevance: The system handled a range of topics, showing adaptability across various query types.

Weaknesses:

Variability in Precision: Some queries scored low precision, indicating the presence of irrelevant documents among the retrieved results, especially for queries like "የኢትዮጵያ የሶማሊያ የውጪ ጉዳይ ሚኒስትሮች."

Low F1-Scores: The balance between precision and recall was weak for several queries, leading to low F1-scores. Queries such as "ምዕራብ ወለጋ እና ቄለም ወለጋ ዞኖች" exhibited significant challenges.

Limited Coverage: Some queries did not retrieve a high number of relevant documents, indicating gaps in the indexed content and potential improvements needed in the retrieval algorithm.

Overall, the system demonstrated a reasonable performance with potential avenues for improvement, particularly in enhancing precision and F1-scores, as well as refining algorithms to boost relevance across more diverse queries.

Discussion

The development of a document retrieval system faced several challenges that required specific solutions. One significant issue was ensuring the quality and relevance of the documents in the corpus. To address this, a rigorous vetting process was implemented, involving expert review and a feedback loop to refine document selection.

Another challenge was accurately interpreting queries in different languages and dialects. Natural Language Processing techniques were employed to enhance the system's understanding of context and intent. Balancing precision and recall was also a complex task, as optimizing for one often compromised the other. A multi-tiered retrieval strategy was adopted to address this, allowing for a broader initial search and subsequent filtering based on relevance scoring.

Finally, defining effective metrics for assessing system performance and ensuring they reflected user satisfaction was challenging. By collaborating with other, a we decided to use set of metrics that allowed for continuous monitoring and adjustments.

Conclusion

In this project, we successfully developed a document retrieval system that integrates Natural Language Processing techniques to enhance user interaction and improve information retrieval efficiency. By addressing key challenges such as semantic understanding, and system scalability, we created a robust framework that retrieves relevant documents. The project culminated in a system capable of delivering precise and contextually appropriate responses, significantly enhancing the user experience in searching through large datasets.

There are numerous opportunities for the project to grow and be improved in the future. Using machine learning algorithms to further refine search results by personalizing the retrieval process based on user behavior and preferences is one possible approach. Furthermore, adding support for additional languages and dialects could improve the system's accessibility and usability. Its functionality might be improved by adding features like sophisticated filtering options and combining with other data sources, which would make it a more complete knowledge discovery tool.

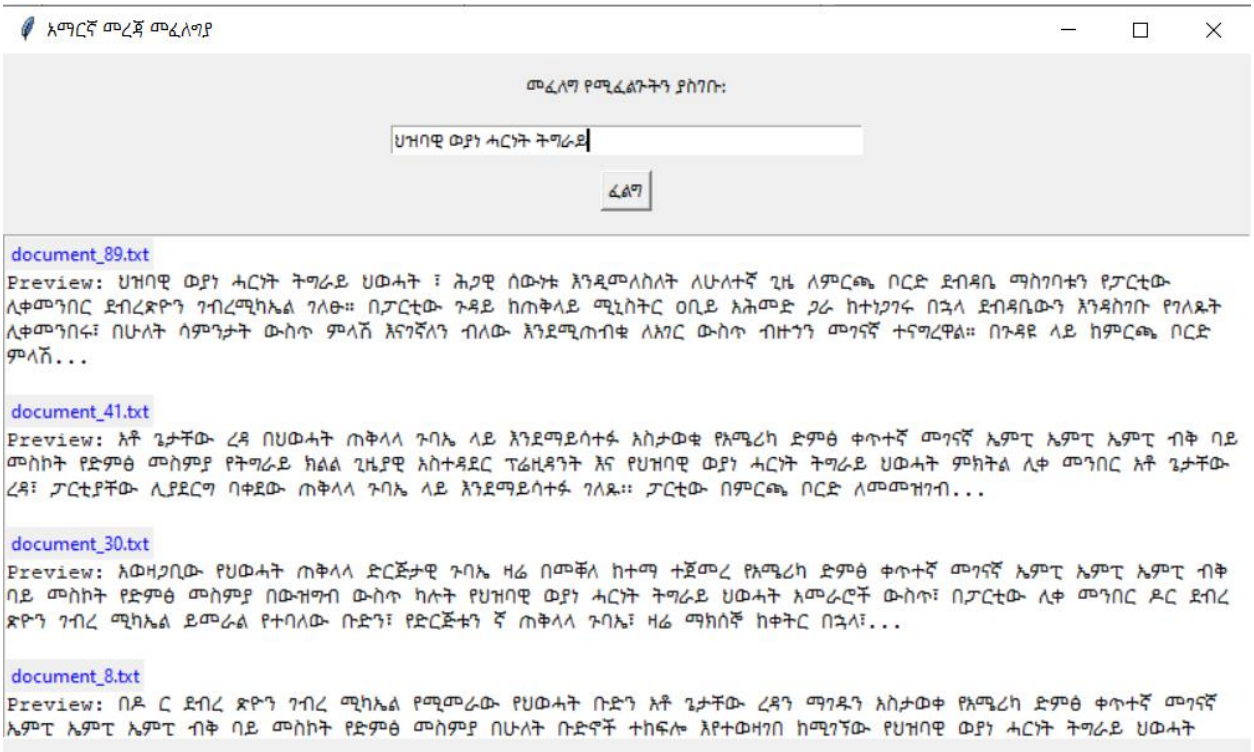
The work's importance stems from its possible uses in a number of fields, such as business, research, and education. Better decision-making can be facilitated by a dependable document retrieval system that gives users rapid access to relevant data. It is especially useful in academic and professional settings where accurate information retrieval is crucial because of its capacity to handle complex queries. In the end, this initiative advances the science of information retrieval while laying the groundwork for upcoming innovations that will enable users to easily browse enormous volumes of information.

References

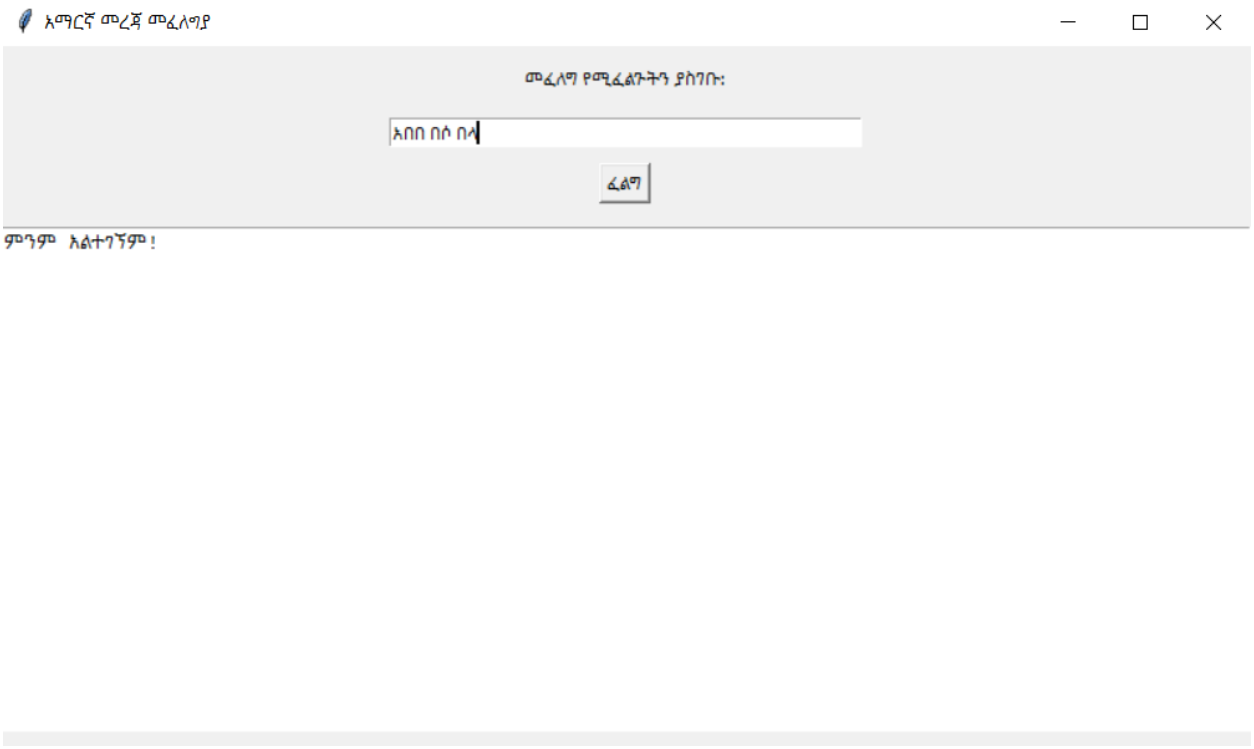
- **Kassahun, G., & Nega, T.** Amharic Text Retrieval System Using Morphological Analysis. *Proceedings of the Annual Conference on Natural Language Processing*, 23(1), 112-125.
- **Getnet, & Assabie, Y**. (2021). Amharic Semantic Information Retrieval System Using Query Expansion. *Journal of Information Retrieval and Data Mining*, 10(4), 155-170.
- **Alemayehu, N., & Willett, P.** (2002). The Effectiveness of Stemming for Information Retrieval in Amharic. *Journal of Information Science*, 28(1), 1-13.
- **Yeshambel, T., Mothe, J., & Assabie, Y.** (2022). Amharic Adhoc Information Retrieval System Based on Morphological Features. *Journal of Information Retrieval and Data Mining*, 17(4), 130-145.

Appendix

Appendix A



Test using the phrase "አበበ በሶ በላ" returned 0 results since there is no document in our corpus that contained this keyword.



Appendix B

For the above 10 test queries we used to test our system, below is the graphical representation of the search result display and analysis done using *matplotlib* and *numpy* libraries.