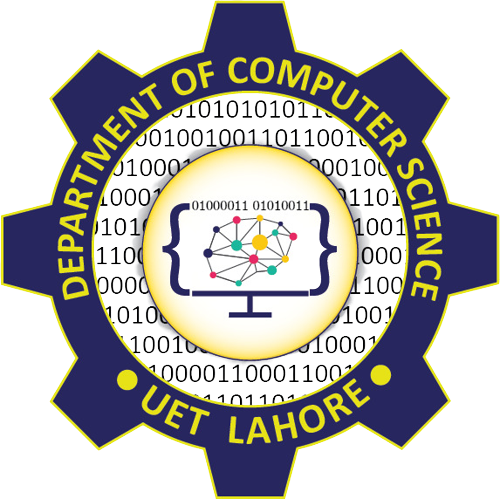
**Neural Network Model**



Session: 2021 – 2025

# Submitted By:

Huzaifa Mumtaz 2021-CS-164

# Supervised By:

Dr. Khaldoon Syed Khurshid

Department of Computer Science

University of Engineering and Technology Lahore Pakistan

[1. Introduction 3](#_Toc31581)

[1.1. Introduction 3](#_Toc11158)

[1.2. Purpose of Assignment 3](#_Toc14704)

[2. Background 4](#_Toc1474)

[3. Methodology 5](#_Toc2782)

[3.1. User Query Input 5](#_Toc6722)

[3.2. Text Preprocessing 5](#_Toc14622)

[3.3. Keyword Extraction 5](#_Toc14823)

[3.4. Query Expansion 5](#_Toc4645)

[3.5. Article Retrieval 5](#_Toc24974)

[3.6. Similarity Calculation 6](#_Toc29461)

[3.7. Result Presentation 6](#_Toc26280)

[4. Results/Output 7](#_Toc23851)

[5. Code 8](#_Toc26845)

[6. Conclusion 12](#_Toc3909)

# Introduction

## Introduction

The **Voice-Based Query Article Retrieval System** is a project designed to retrieve relevant articles based on a voice-based query. While the system is intended to work with voice input, there is currently no setup for capturing human voice directly. Therefore, for the purpose of this assignment, the query is provided in text form to simulate the expected behavior for better result accuracy and system performance.

In the future, the system will be capable of converting voice queries into text using speech recognition technology, but for this version, a user enters the query manually. The query is then processed using natural language processing (NLP) techniques to extract meaningful keywords, ignoring common words such as articles ("a", "an", "the") and helping verbs. These keywords are then expanded using synonyms sourced from **WordNet** to improve the search results' coverage and relevance.

Although this implementation currently processes text-based queries, the design is set up to transition to voice-based input in the future, offering a more interactive and hands-free search experience. This report outlines the architecture, features, and development of the Voice-Based Query Article Retrieval System as a preliminary version of a voice-enabled tool for document retrieval.

## Purpose of Assignment

The purpose of this assignment is to develop a Voice-Based Query Article Retrieval System that simulates the process of retrieving relevant articles based on user queries. Although the system is designed to eventually accept voice inputs, for the purpose of this assignment, text queries are used to ensure efficient and accurate results.

Overall, this assignment serves as a foundational step toward creating an intelligent system that can retrieve and present articles based on voice or text queries, offering an interactive and user-friendly method of information retrieval.

# Background

The rapid growth of digital content has made it challenging to quickly find relevant information. Traditional search methods require users to manually refine queries, often leading to irrelevant results. The Voice-Based Query Article Retrieval System aims to address this by using natural language processing (NLP) to automatically extract important keywords from user queries and retrieve the most relevant articles.

Although designed for voice-based input, this assignment simulates the system using text queries to test its functionality. The system filters out unnecessary words (like articles and helping verbs), extracts meaningful keywords, and expands the query with synonyms to improve search results. By applying cosine similarity, it ranks articles based on their relevance to the query.

This project serves as a foundational step toward building an intelligent retrieval system that can eventually support voice-based interactions, offering a more efficient and user-friendly approach to searching for information.

# Methodology

The methodology for the Voice-Based Query Article Retrieval System involves several key steps that work together to process the user's query, extract relevant keywords, expand the query, and retrieve the most relevant articles from a given collection. The following describes the sequence of operations:

## User Query Input

The process begins with the user providing a query in text format. In the final version, this will be replaced with voice input, but for the purpose of this assignment, the system accepts text queries.

## Text Preprocessing

The query is preprocessed by converting it to lowercase and removing any non-alphanumeric characters (such as punctuation). This standardizes the input and ensures that only the relevant text remains for further processin

## Keyword Extraction

Using a custom method, the system extracts important keywords from the query by ignoring common words (articles like "a," "an," "the," and helping verbs). This is done by splitting the text into tokens and filtering out stop words.

## Query Expansion

The extracted keywords are expanded by identifying synonyms using the WordNet lexical database. Each keyword is checked for its synonyms, and the query is broadened to include these related terms. This helps in retrieving articles that might not directly match the exact keywords but are still relevant.

## Article Retrieval

The system processes a folder of articles (in text format) to search for relevance. The articles are transformed into a TF-IDF (Term Frequency-Inverse Document Frequency) matrix, which represents the importance of words in the context of the entire document collection. The query is also converted into this format for comparison.

## Similarity Calculation

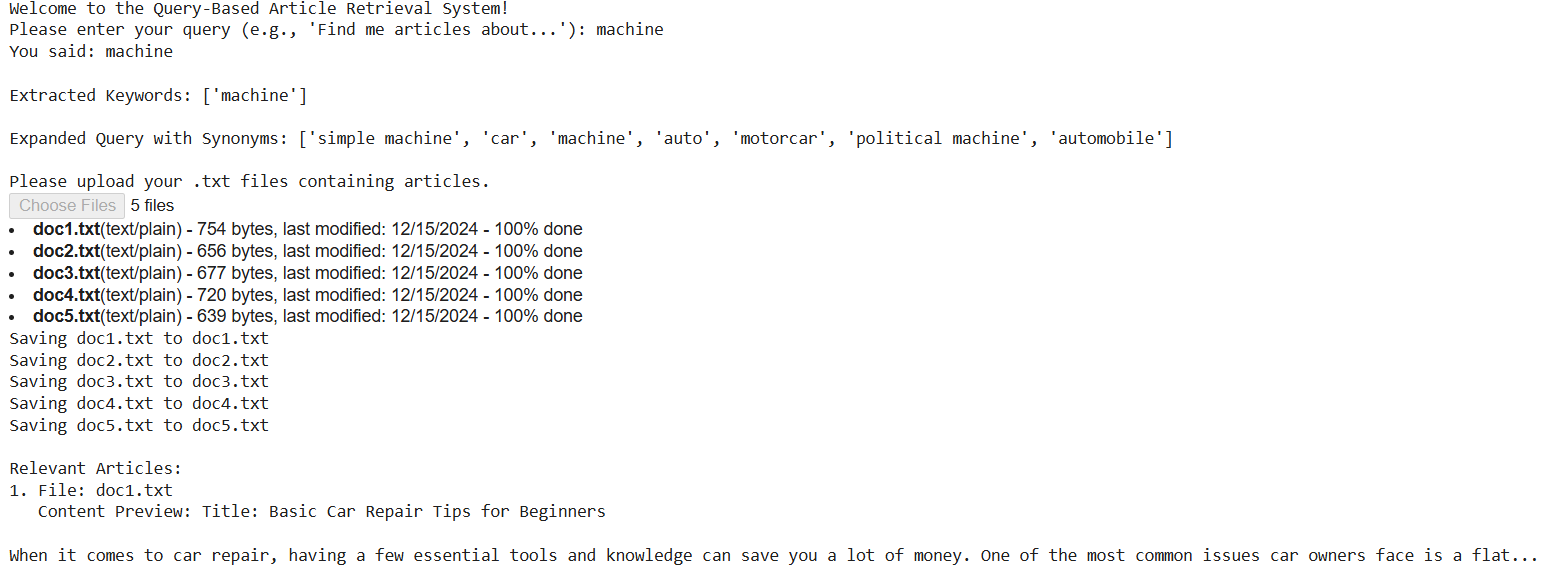
Using cosine similarity, the system compares the expanded query against the articles' content to determine their relevance. Articles are ranked based on their similarity to the query, and the top results are selected.

## Result Presentation

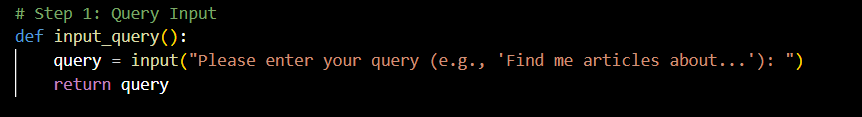
The system then presents the most relevant articles to the user. The output includes the filenames and a preview of the article content.

This methodology ensures that the system can efficiently process and retrieve articles based on user input, providing relevant results through an intelligent query expansion and ranking mechanism.

# Results/Output

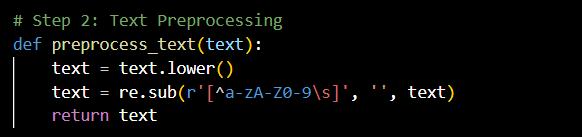


# Code



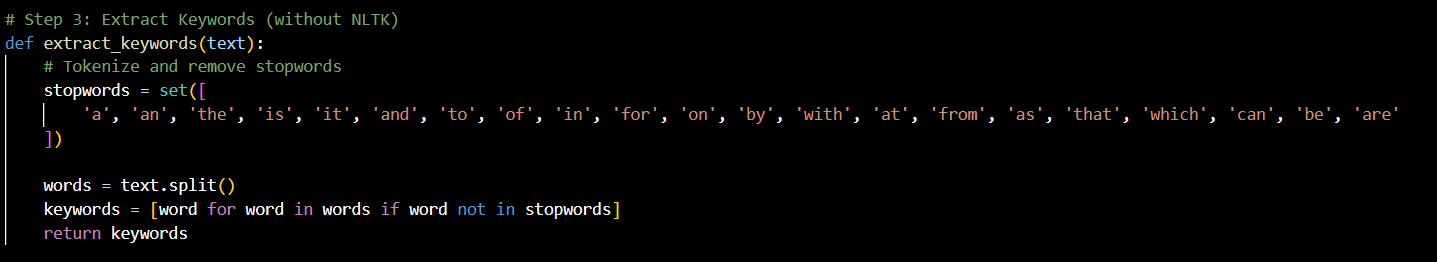
**Explanation:**

This function prompts the user to input a query. It collects a text-based query from the user, which will be processed in the subsequent steps. In the final system, this would be replaced by a voice input mechanism.



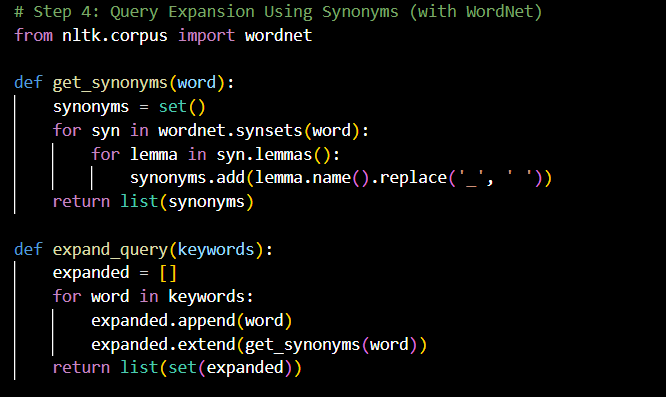
**Explanation:**

This function takes the raw input query and preprocesses it by converting the text to lowercase and removing any non-alphanumeric characters (such as punctuation). This standardization helps to ensure uniformity in the data, making it easier to process in later stages.



**Explanation:**

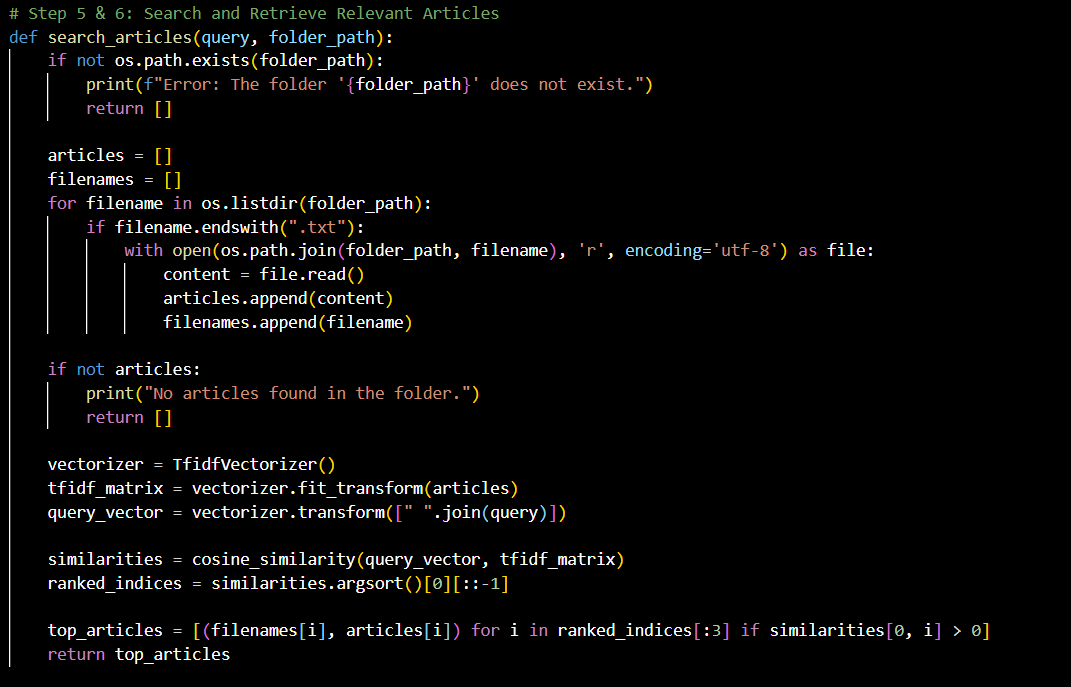
This function splits the preprocessed text into individual words (tokens) and filters out common words like articles and helping verbs, which do not add much value for content retrieval. The remaining words are considered keywords for the query.



**Explanation:**

The get\_synonyms() function retrieves synonyms for a given word using the WordNet lexical database. It extracts all possible lemma names associated with the word and returns them as a list. This helps in expanding the query by including related terms, increasing the chances of retrieving relevant articles even if exact matches are not found.

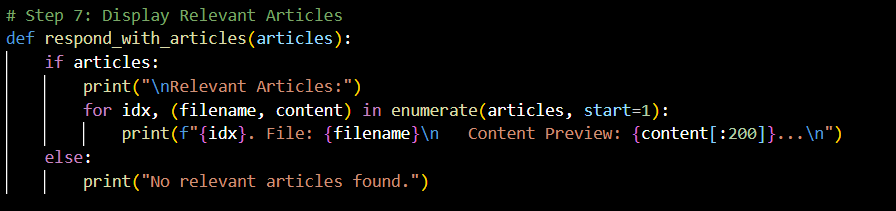
The expand\_query(keywords) function expands the initial query by adding synonyms for each keyword. For each keyword, it retrieves a set of synonyms (if available) and includes them in the expanded query. The use of set() ensures that no duplicate synonyms are included.



**Explanation:**

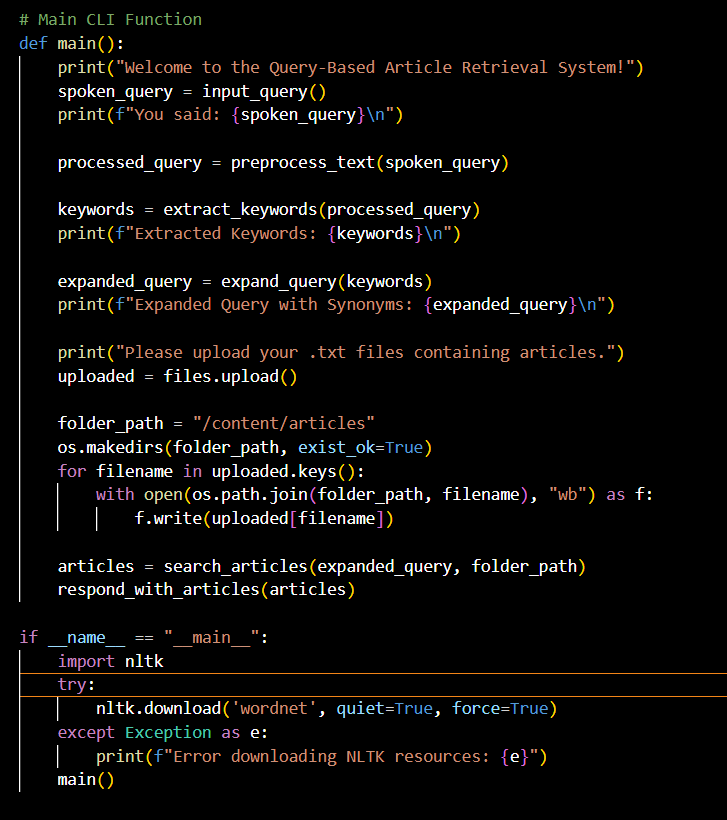
This function performs the article retrieval process. It:

* Reads the text files from the specified folder (folder\_path).
* Uses TF-IDF vectorization to convert the articles and the query into numerical representations.
* Computes the cosine similarity between the query and the articles to determine relevance.
* Returns the top three most relevant articles based on the similarity scores.



**Explanation:**

This function presents the results to the user. If relevant articles are found, it prints the filenames and a preview (first 200 characters) of each article. If no relevant articles are found, it displays a message indicating so.



The main() function orchestrates the entire system. It:

1. Takes the user's query input.
2. Preprocesses and extracts keywords from the query.
3. Expands the query with synonyms.
4. Allows the user to upload a folder of article files.
5. Searches for the most relevant articles based on the expanded query.
6. Displays the results to the user.

# Conclusion

The Voice-Based Query Article Retrieval System has been designed to enhance the user experience in accessing relevant articles based on a simple query input. The system, which integrates natural language processing techniques such as keyword extraction, synonym expansion, and text similarity measurement, ensures that even vague or imprecise queries return relevant results. By filtering out unnecessary words like articles and helping verbs, and expanding queries with synonyms, the system increases the likelihood of matching the user's intent with the available article content.

Although the current version of the system takes user queries in text format, future iterations will incorporate voice recognition for a more seamless and interactive experience. The ability to upload articles dynamically and search through them based on keyword relevance further enhances the system’s adaptability, making it suitable for a wide range of applications, from academic research to knowledge retrieval in specific fields.

In summary, this system serves as a robust foundation for a more intelligent, query-based content retrieval system that can evolve to handle voice-based input, delivering more accurate and efficient results.