

**Natural Language Processing**

Project Proposal

**RAG Based Multilingual News Retrieval**

**Group 7**

**RAG Based Multilingual News Retrieval**

**Basic Idea**

In an era of information overload, users face significant challenges when trying to find relevant and diverse news articles across multiple languages. Existing retrieval systems are often limited to monolingual or regional sources, leading to bias in available information and neglect of pertinent content in non-dominant languages. Users frequently encounter the frustration of missing critical news perspectives or context, especially when tracking global issues or looking for regional insights in a preferred language.

Even if the user attempts to retrieve articles in multiple languages, current solutions often require separate, language-specific searches, leading to inefficient information retrieval and fragmented results. This lack of an integrated, multilingual retrieval system results in an incomplete, often one-sided view of events.

**Approach to Solution**

To address this, we propose a custom Retrieval-Augmented Generation (RAG) model that will allow users to search for relevant news articles across multiple languages based on an input query, returning summarised content in the English Language. The RAG model will leverage a multilingual knowledge base of news articles, integrating state-of-the-art language models for both retrieval, summarization and generation tasks. This setup enables cross-lingual search and translation, ensuring that relevant content is accessed from multiple regions and languages.

The workflow is as follows:

1. **Multilingual News Articles Collection:**
   1. **Data Source:** Using Wiki-Lingual Hugging Face dataset and Arabic News Articles Dataset from Kaggle, we collect news articles from various domains and languages.

Dataset: https://huggingface.co/datasets/reciTAL/mlsum

1. **Domain-Specific Filters:** For optimised retrieval and relevance, articles are pre-filtered by specific domains, e.g., science, business, or regional news, which can be adjusted based on user needs.
2. **Translation and Summarization Pipeline:**
   1. **Summarization Model:** A transformer-based summarization model like BART or T5, fine-tuned on news article datasets with summaries in English, will condense the retrieved articles. This model provides concise summaries, highlighting key points for quick consumption.
   2. **Translation Model for Consistency**: For non-English articles, a model like mBART performs the translation to English. This translation step ensures that all summaries are consistently in English, regardless of the original language.
   3. **Summarization Fine-Tuning:** The summarization model is trained on a dataset with paired articles and summaries to ensure clarity and relevance in news summaries.
3. **Multilingual Transformer-Based Retriever:**
   1. **Creating the Knowledge Base:** Articles from News dataset are translated to English using a translation model (e.g., mBART for multilingual translation).
   2. **Embedding Generation and Storage:** Each translated article is embedded into a vector using multilingual transformer models like ‘mBERT’ to capture cross-lingual semantics. These embeddings are stored in a vector database, enabling fast retrieval based on user queries.
   3. **Efficient Retrieval**: When a user submits a query, the system uses the multilingual retriever to identify the top n most relevant articles, ensuring the retrieval is language-agnostic and reflects diverse perspectives.
4. **User Query Processing and Output Generation:**
   1. **User Input:** The user inputs a query on a specific topic or event, which is converted into an embedding using a transformer model.
   2. **Retrieve Top N Articles:** The vector database is queried to retrieve the N most relevant articles based on embedding similarity to the query.
   3. **Summarized Output:** The system returns summaries for the top N articles, each capturing key details from the original articles, providing a brief, coherent response aligned with the user’s query.

**Why Is It Useful?**

This tool lets users follow international news more easily, especially from regions where English news coverage might be limited. By combining translation, relevant article retrieval, and summarization, the system provides an efficient, accessible way to stay informed about the world, all tailored to each user’s interests.

**Related Work**

1. Lewis et al., 2020, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks” – This paper presents the foundational RAG model and explores the application of RAG to improve answer generation by retrieving relevant documents. The work demonstrates that retrieval-augmented methods outperform traditional models in knowledge-intensive tasks by leveraging both retrieval and generation capabilities.
2. Ladhak et al., 2020, “WikiLingua: A Multilingual Abstractive Summarization Dataset” – This research explores cross-lingual summarization, presenting a large-scale multilingual dataset. The findings underscore the importance of multilingual models for effective cross-lingual summarization, and the work provides insights into handling language diversity and translation for summarization tasks.

**Assessment Methodology**

**1. Performance Evaluation Metrics:**

To ensure meaningful and precise translations as well as accurate content retrieval, we have employed a range of advanced evaluation metrics:

1. **BLEU Score:**
   * Measures n-gram overlap between the machine-generated summary and the reference summary.
   * BLEU is commonly used for translation, but it is also applied to summarization.
2. **BERTScore:**
   * Measures the similarity of the BERT embeddings of the system-generated summary and the reference summary.
   * It accounts for contextual information, unlike BLEU.
3. **Cosine Similarity:**
   * Uses embeddings from SBERT to compute the similarity between the system-generated summary and the reference summary.
   * Cosine similarity provides a numerical value between 0 and 1 to represent alignment.
4. **SBERT Score:**
   * Uses SBERT to compute the alignment of reference summaries with system-generated summaries.

**2. Ablation Settings:**

* **Data Input Variations:** We will experiment with datasets of differing sizes and language complexities to understand their impact on retrieval and summarization.
* **Algorithm Complexity:** Testing simpler and more complex RAG architectures, such as various transformer layers and embeddings, to determine the best trade-off between performance and processing time.
* **Preprocessing Techniques:** Assessing the impact of various tokenization, language normalisation, and translation preprocessing steps on the system's output quality.

Ablation experiments will help rank each component's impact on final performance, guiding us in optimising the model for accurate and efficient information retrieval.

**Workflow:**

**1. Data Collection and Loading**

**Objective:** Collect, load, and prepare multilingual data from the MLSUM dataset.  
**Steps:**

1. **Dataset Selection:** The **MLSUM** dataset is used for multiple languages, including German (de), Spanish (es), French (fr), Russian (ru), and Turkish (tu).
2. **Download Data:** For each language, 700 records are downloaded from the training split of the MLSUM dataset.
3. **Dataframe Creation**: Each language's data is converted into a **Pandas DataFrame** with the following key columns:
   * Text (the input text)
   * Summary (the summary for the input text)
   * Language (an identifier for the language)
4. **Data Concatenation:** The individual DataFrames are combined into a single unified DataFrame containing text and summary data for all languages.
5. **Dataset Conversion:** The concatenated DataFrame is converted to a HuggingFace Dataset format.

**2. Data Preprocessing**

**Objective**: Prepare the text data for translation and summarization.  
**Steps**:

1. **Language Column Addition:** A column specifying the language of each text is added.
2. **Data Cleaning:** Unnecessary columns, such as URLs, are dropped to reduce irrelevant information.
3. **Language Detection (Optional):** The langdetect library is used to ensure the text is in the correct language.

3. **Translation**

**Objective**: Translate multilingual text into English using the MBART50 model.  
**Steps**:

1. **Model Selection:** The MBART50 model is used, which supports many-to-one multilingual translation.
2. **Tokenizer Initialization:**
   * The MBart50Tokenizer is initialized with src\_lang and tgt\_lang set to English (en\_XX).
3. **Translation Function:** A custom function, translate\_text(text), is defined to:
   * Prepare the input text for translation.
   * Tokenize the input and generate model predictions.
   * Decode the translated output using the tokenizer.
4. **Translation Output:** The translated output is saved or used for further summarization.

**4. Summarization**

**Objective:** Summarize the translated text.  
**Steps**:

1. **Model Selection:** The MBART50 model is also used for text summarization, since it supports summarization as well.
2. **Tokenizer Initialization:**
   * The same MBart50Tokenizer is used, but the summarization is prompted differently from translation.
3. **Summarization Function:** A custom function, summarize\_text(text), is defined to:
   * Prepare the input prompt to instruct the model to summarize the text.
   * Tokenize the text input.
   * Generate the summary using the MBART50 model.
   * Decode the model's output to return the summary.
4. **Summarization Output:** The summary for each piece of text is generated and stored for evaluation.

5. **Model Inference**

**Objective**: Apply the translation and summarization functions to the entire dataset.  
**Steps**:

1. **Loop Through Dataset:** Loop through each row of the dataset and:
   * Translate each text into English using translate\_text().
   * Summarize the translated text using summarize\_text().
2. **Result Storage:** Store the translated news articles and summarized version of translated articles to be used for RAG.

**6. Retrieval-Augmented Generation (RAG)**

**Objective**: Enhance content generation by incorporating relevant external knowledge from a retrieval system before summarization.

**Knowledge Base:**

* The knowledge base is built using the same dataset of translated news articles and their summaries.
* All the articles in the dataset, along with their corresponding summaries, are stored as part of the knowledge base.

**Steps**:

1. **Knowledge Base Construction:**
   * Each entry in the translated dataset (text and summary) is added to the knowledge base.
   * The knowledge base is indexed using FAISS or a similar dense vector retrieval system.
2. **Query Formation:**
   * For each input query, a query vector is generated using SBERT or other embedding models.
   * The query is designed to capture the intent and context of the input text.
3. **Information Retrieval:**
   * The system searches the knowledge base using FAISS to retrieve the top 5 most relevant news articles and their summaries.
   * The retrieved content serves as contextual input for the summarization process.
4. **RAG Model:**
   * The Retrieval-Augmented Generation (RAG) model takes the input query and the 5 retrieved articles and summaries as input.
   * It generates a summary that incorporates contextual information from the 5 retrieved articles and their summaries.
5. **Storage of Results:**
   * The RAG-generated summary and the 5 most relevant retrieved articles to the input query  are stored for later evaluation.

**Key Point:**

The RAG system enhances the summarization process by retrieving the top 5 most relevant news articles and their summaries from the knowledge base. This approach ensures that the summary is context-aware and aligned with current knowledge.

**7. Similarity and Relevance Analysis**

**Objective**: Measure how well the input query aligns with the summaries retrieved.  
**Steps**:

1. **BERTScore Calculation:**
   * BERTScore measures semantic similarity between the system generated and RAG retrieved summaries and the input queries using BERT embeddings.
2. **Cosine Similarity Calculation:**
   * The Cosine Similarity between the vector embeddings of the  system generated and RAG retrieved summaries and the input queries is calculated.
   * Vector embeddings are obtained using SBERT (Sentence-BERT), which provides high-quality sentence-level embeddings.
3. **SBERT Score Calculation:**
   * Similar to cosine similarity, but SBERT scores offer finer contextual alignment between  system generated and RAG retrieved summaries and the input queries.

**Integration with Streamlit for Interactive Application**

To bring the RAG-based multilingual news retrieval system to life, we have developed an interactive user interface using **Streamlit**. This user-centric application enables easy data exploration, query input, and result visualization, making the system accessible and intuitive for end-users. Below are the key features of the Streamlit implementation:

**Key Features of the Streamlit Application**

1. **Title and Description**:

• The application starts with a user-friendly title, “Multilingual News Retrieval and Similarity Analysis,” along with an explanation of its purpose and functionalities.

2. **Sidebar Configuration**:

• Users can specify the dataset path for multilingual articles directly through the sidebar.

• Other configurations, such as the number of results (top\_k), are also adjustable in the sidebar, offering flexibility in retrieval.

3. **Dataset Preview and Embedding Generation**:

• The application dynamically loads the dataset using the user-specified path.

• After loading, it generates embeddings for the translated\_text column using a pre-trained SentenceTransformer model. These embeddings encapsulate the semantic meaning of the articles, which are then used for retrieval tasks.

4. **FAISS Index Creation**:

• A FAISS (Facebook AI Similarity Search) index is built from the generated embeddings. This index enables high-speed retrieval of similar articles based on user queries.

5. **Query Input and Retrieval**:

• Users can input queries in the main panel, which are then encoded into embeddings using the same SentenceTransformer model.

• The encoded query is compared against the FAISS index to retrieve the top N most relevant articles.

6. **Similarity and Semantic Alignment**:

• The retrieved articles are further analyzed for similarity using multiple metrics:

• **Cosine Similarity**: Evaluates the alignment between the query embedding and the embeddings of retrieved articles.

• **BERTScore**: Measures semantic similarity between article summaries and the user query.

• These scores help rank the retrieved results, ensuring that users receive the most contextually relevant articles.

7. **Dynamic Result Display**:

The retrieved articles are displayed in a structured and interactive table, showing critical information such as:

• Title

• Topic

• Translated text

• Summary

• Similarity scores (Cosine and BERTScore)

8. **Caching for Performance Optimization**:

The application uses Streamlit’s caching features (@st.cache\_data and @st.cache\_resource) to optimize the performance of embedding generation, FAISS indexing, and data loading. This ensures a smooth and efficient user experience, even with large datasets

**Streamlit Implementation Workflow**

1. **Data Loading**:

• Users specify the dataset path via the sidebar, and the system dynamically loads the multilingual dataset into a Pandas DataFrame.

2. **Embedding Generation**:

• A pre-trained SentenceTransformer model (multi-qa-mpnet-base-dot-v1) generates embeddings for each article’s translated text.

3. **FAISS Index Building**:

• The embeddings are indexed using FAISS, enabling efficient similarity searches for user queries.

4. **User Query Processing**:

• The user provides a query in the input field, which is converted into an embedding using the SentenceTransformer model.

5. **Retrieval and Analysis**:

• The system retrieves the top N relevant articles based on embedding similarity from the FAISS index.

• Similarity metrics such as Cosine Similarity and BERTScore are computed for deeper semantic alignment.

6. **Interactive Visualization**:

• The results, including key fields and similarity scores, are displayed interactively, allowing users to explore retrieved articles seamlessly.