Spoken Query Retrieval in Monolingual Contexts with Whisper and SBERT Models

Prabhu Ram Nagarajan[1,*], Livin Nector Dhasan[1] and Meera Devi Thiagarajan[2]

[1] *BS Degree, Indian Institute of Technology Madras*

[2] *ECE,Kongu Engineering College,Erode*

Abstract

This work is part of the SqCLIR 2024 shared task, which aims to develop a system for monolingual spoken query information retrieval. The system processes both text and spoken queries from a 0.4 million-entry corpus stored in a compressed format. Text data is extracted and vectorized using the Sentence Transformer model, while spoken queries in WAV format are transcribed with the Whisper model and then vectorized using Sentence-BERT (SBERT) for effective matching with the document corpus. The documents are ranked based on cosine similarity between the query and document embeddings. Evaluation results, including MAP (0.0414), MRR (0.2414), and Recall@K (R@100: 0.1279, R@1000: 0.2503), demonstrate solid performance, though there is potential for improvement in ranking algorithms and vectorization. The system currently focuses on monolingual queries. However, future work will address challenges in multilingual and under-resourced contexts. In these settings, language diversity and limited resources can impact both retrieval accuracy and system performance. The code used to generate these results is publicly accessible at github

Keywords

SBERT Model, Whisper Model, Spoken query, Information retrieval

1. Introduction

The growth of user-generated content has increased the need for effective search capabilities across multimedia databases. While text, image, and video search systems are well-developed, unstructured audio remains largely inaccessible [1]. Transcribing and retrieving spoken content is essential for improving the accessibility and usability of large multimedia datasets, especially in contexts like parliamentary proceedings, where vast amounts of spoken data need efficient processing and retrieval.

Traditional text-based information retrieval (IR) systems must adapt to incorporate audio data, which requires innovative solutions. Combining automatic speech recognition (ASR) with advanced retrieval techniques could enhance the accuracy and contextuality of search results. Improving result ranking in IR systems, particularly those handling unstructured audio, is an ongoing challenge [2].

In natural language processing, recent shifts from static word embeddings to contextualized ones like BERT allow for richer semantic representations. BERT's dynamic embeddings, which vary based on context, are effective for tasks like sentence embedding. However, its large size makes tasks like clustering and semantic search computationally expensive. Methods like SBERT, which fine-tune BERT on labeled sentence pairs, help improve embedding quality [3].

The shared tasks SqCLIR 2024 [4, 5] focused on developing and evaluating retrieval systems that process spoken queries and search for relevant documents in a corpus. Task 1 Spoken Query Ad-Hoc Retrieval Data - Monolingual Task involves a monolingual Spoken query Retrieval System, where both the spoken query and the corpus are in the same language (English, Gujarati, Hindi, or Bengali), simplifying the retrieval process. It is required to design systems that accurately interpret spoken queries and retrieve relevant documents in the same language. Task 2(Spoken Query Cross-Lingual Retrieval), on the other hand, addresses cross-lingual retrieval, where the spoken queries and the

FIRE'24: Forum for Information Retrieval Evaluation 2024, December 12–15, 2024, India

[*] Corresponding author.

prabhuramnphd@gmail.com (Prabhu Ram Nagarajan); livinnector2001@gmail.com (Livin Nector Dhasan); tmeeradevi@gmail.com (Meera Devi Thiagarajan)

0000-0003-2769-9790 (Prabhu Ram Nagarajan)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

document corpus are in different languages, adding complexity. Here, the system must interpret spoken queries in one language (English, Hindi, or Bengali) and retrieve documents in another language, with various language pairs offering diverse cross-lingual retrieval challenges. This paper focuses on the development of a system for monolingual spoken query information retrieval.

In the following sections, recent works involving the SBERT and Whisper transformer models are discussed in Section 2, followed by a description of the audio dataset and the training of text queries in Section 3. Section 4 covers the discussion of results, concluding with Section 5.

2. Related Works

The integration of voice technology into everyday applications has profoundly changed how users engage with information retrieval systems. Spoken queries, known for their conversational style and varied phrasing, introduce unique challenges compared to traditional text-based searches. This research explores the effectiveness of pre-trained Whisper and SBERT [6, 7] models in handling spoken queries within monolingual information retrieval systems. By analyzing retrieval performance and user satisfaction, the study aims to shed light on how advanced NLP techniques can enhance information retrieval processes, ultimately creating a more intuitive and efficient user experience.

Whisper is a cutting-edge open-source automatic speech recognition (ASR) system trained on 680,000 hours of weakly supervised speech data. It supports nearly one hundred languages and performs tasks such as language identification, transcription, and translation into English. Whisper's use of byte pair encoding (BPE) allows it to output any character sequence, avoiding the limitations of traditional word-based ASR systems. Its competitive performance with advanced systems and human transcribers makes it a strong candidate for ASR applications[8].

SBERT is used to enhance text analysis through the creation of embeddings. It generates embeddings from text claims to calculate similarity between the embedding vectors, which aids in classification and semantic search. Additionally, it is also applied to transform texts into dense vectors for text clustering, with a focus on evaluating different pooling techniques to optimize performance. SBERT serves as a foundational tool for effectively measuring text similarity and improving analysis in classification and semantic search tasks[9, 10]. Recent advancements in deep learning have facilitated the development of closed-domain question answering (QA) systems based on frequently asked questions (FAQs). One notable approach involves an ensemble system that integrates TF-IDF and fine-tuned V-SBERT models to enhance retrieval performance on a novel dataset of admission FAQs from a Chinese university. This study underscores the importance of natural language processing (NLP) techniques in effectively representing user queries and evaluating the performance of various encoding models within the QA context [11]. SBERT-WK enhances BERT sentence embeddings through geometric analysis of word representations, leveraging multiple layers for better performance in tasks like semantic textual similarity and linguistic probing, without additional training. Meanwhile, SentenceBERT (SBERT) improves sentence embeddings by combining a pre-trained language model with global average pooling (GAP), refining word representations using contextual information. SBERT excels in tasks like semantic textual similarity and remains a foundational model for sentence embedding, despite newer approaches like TA-SBERT[3, 12].

While discussing bi-encoders, it is relevant to note recent research that investigates whether bi-encoders, without additional fine-tuning, can achieve performance comparable to fine-tuned BERT models in classification tasks. One study proposes an effective approach that prepares multiple bi-encoders and selects the most suitable ones based on the dataset at hand. Their experimental results indicate that this method performs comparably to fine-tuned models across various datasets, including AG News and SMS Spam Collection. Although, this research highlights potential applications in areas such as K-12 AI programming education, where pre-trained models can be applied to smaller datasets without the need for extensive fine-tuning[13].

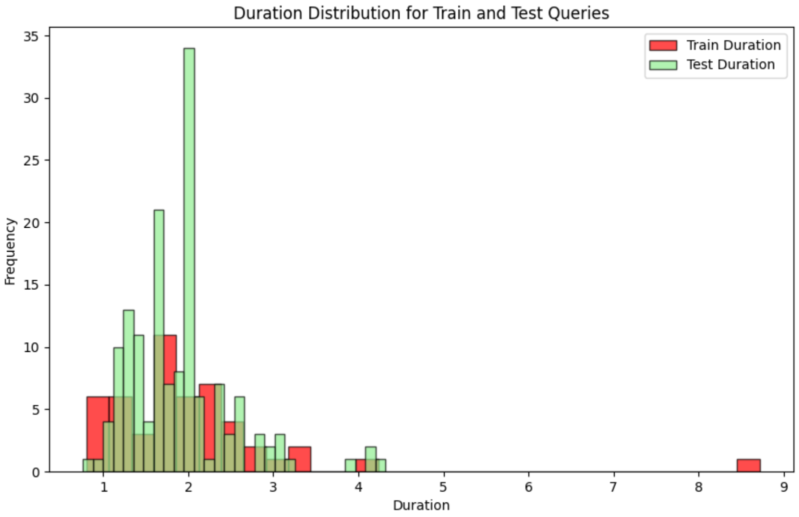

Figure 1: Duration Distribution of the Spoken query

3. Methodology

The flow of methodology has been described in detail in the following subsections.

3.1. Dataset Description

The Indian Constitution recognizes 22 languages, including Assamese, Bengali, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Manipuri, Marathi, Nepali, Odia, Punjabi, Sanskrit, Sindhi, Tamil, Telugu, Urdu, Bodo, Santhali, Maithili, and Dogri. Developing retrieval systems that can process spoken queries is a relatively underexplored area in information retrieval, particularly for under-resourced languages. This topic is the focus of a new shared task for FIRE 2024, which includes two main components: Spoken query Ad-Hoc Retrieval Data and Spoken query Cross-Lingual Retrieval[4, 5, 14, 15]. [1] The FIRE Spoken query data is sourced from the FIRE dataset and recorded by native speakers fluent in the respective languages. There are 50 training spoken queries and 150 testing spoken queries, all provided in .wav format. The file names indicate the language and query ID, following the format $en\_123.wav$, where "en" signifies English and "123" represents the query ID. The maximum length of the spoken queries is approximately 9 seconds. Figure 1 illustrates the duration (in seconds) distribution of the spoken queries for both the training and test datasets, measured in seconds.

Query files provide human assessments of the relevance of documents in relation to specific queries and are crucial for evaluating the performance of information retrieval (IR) systems. Each query file consists of four fields: query, ITERATION, DOCUMENT, and RELEVANCY as shown in Table 1. The query field serves as the identifier for the search query, while ITERATION is usually set to zero and is often disregarded. DOCUMENT is the unique identifier for each document, and RELEVANCY indicates whether a document is considered relevant (1) or not relevant (0) to the associated query. The training query file has a minimum token length of 3 and a maximum token length of 13. Likewise, the testing query file also features a minimum token length of 3 and a maximum token length of 13. The English corpus consists of 4,47,543 documents encoded in “UTF-8", with a unique document count of 4,30,543.

3.2. Data Extraction and Preprocessing

The corpus is organized with a "DOC" tag, where the title is nested within a "DOC_NAME" tag and the text content is found within a "TEXT" tag. The relevant tags, including "DOC_NAME" and "TEXT," are

[1] https://fire.irsi.org.in/fire/static/data

Table 1
Sample query File with Text Wrapping

query	iteration	document#	relevancy
26	Q0	1050804_bengal_story_5072499.utf8	0
26	Q0	1050806_frontpage_story_5081177.utf8	1
26	Q0	1050815_bengal_story_5116647.utf8	1
26	Q0	1050822_opinion_story_5136585.utf8	1

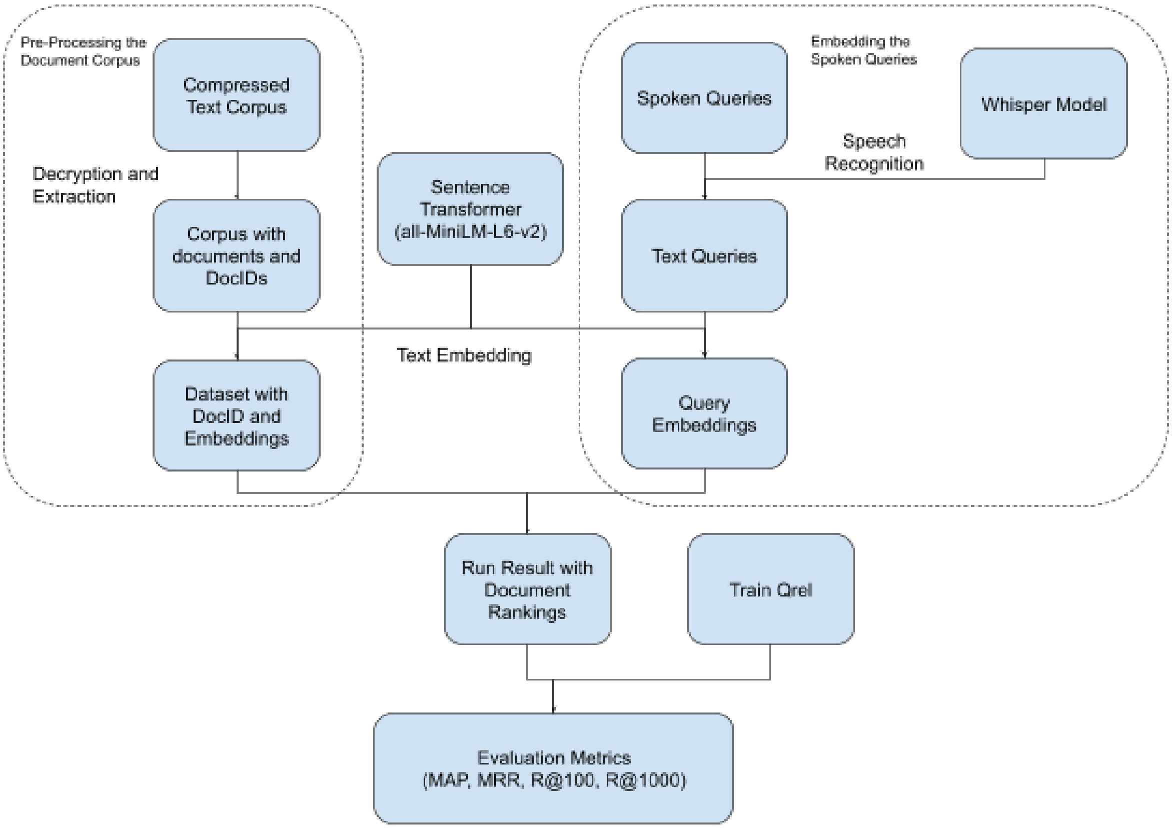
Pre-Processing the Document Corpus
Compressed Text Corpus
Decryption and Extraction
Corpus with documents and DocIDs
Dataset with DocID and Embeddings
Sentence Transformer (all-MiniLM-L6-v2)
Text Embedding
Embedding the Spoken Queries
Spoken Queries
Whisper Model
Speech Recognition
Text Queries
Query Embeddings
Run Result with Document Rankings
Train Qrel
Evaluation Metrics (MAP, MRR, R@100, R@1000)

Figure 2: Flow diagram of the proposed methodology

extracted from the English corpus using Beautiful Soup [16]. Given the large size of the corpus, it is divided into partitions and processed in parallel using the "Joblib" Python package [17].

3.3. Text Embeddings using Sentence Transformers

Text embeddings convert linguistic objects, such as words (tokens) or entire sentences, into numerical vectors within a vector space. These vectors are dense and have a fixed size, such as 384D or 768D, effectively capturing the semantic content of the text. In contrast to traditional vectorization methods like TF-IDF or one-hot encoding, which result in high-dimensional sparse vectors, embeddings provide a compact and meaningful representation. Similar objects in meaning are mapped closer together in the vector space, and this similarity can be quantified using metrics like cosine similarity. For sentence embeddings, the goal is to represent an entire sentence as a dense 384-dimensional vector, which can be thought of as the scaled sum of the individual word embeddings in the sentence. One popular approach for generating sentence embeddings is Sentence-BERT (S-BERT), which modifies the original BERT architecture to create effective sentence-level representations. The 'all-MiniLM-L6-v2' model, a more efficient and smaller variant of BERT, is used to generate high-quality embeddings. Based on knowledge distillation, MiniLM trains a smaller model to replicate the performance of a larger model without incurring the same computational cost. While it uses fewer layers than BERT, MiniLM still maintains high performance through self-attention mechanisms that process input sequences and capture dependencies between words within a sentence.

In this work, documents are extracted and stored in Parquet format, then converted into vector embeddings using the Sentence-Transformers package [18] with the ‘all-MiniLM-L6-v2‘ model [19]. Each document is embedded into a 384-dimensional vector, which is then stored in ‘.npy‘ format. The ‘.npy‘ format is a binary format used to store NumPy arrays efficiently. The benefits of using this format for storing embeddings include:

- Space efficiency: The binary format stores large arrays more compactly than other formats, such as CSV.
- Fast loading: Embeddings can be quickly loaded into memory, which is crucial when working with large-scale datasets.
- Compatibility: The ‘.npy‘ format is widely supported in Python and other scientific computing libraries.

Each document in the corpus is associated with a unique document identifier (DocID). The dataset consists of the document identifier and its corresponding embedding vector, as shown in Figure 2. This approach facilitates efficient retrieval and manipulation of document embeddings for downstream tasks like similarity search.

3.4. Retrieval and Ranking

Whisper is a Transformer-based encoder-decoder model trained on 680,000 hours of labeled speech data with weak supervision. It includes English-only models for speech recognition and multilingual models for both recognition and translation tasks. The model can transcribe audio in its original language or translate it into another language. Whisper offers five configurations, with the four smallest—tiny, base, small, and medium—available as English-only versions, featuring parameter sizes of 39M, 74M, 244M, and 769M, respectively. For retrieving spoken queries, the queries are transcribed into text using the Whisper model’s base configuration [20]. These text queries are then converted into embeddings using the SBERT model, as outlined in 3.3. Cosine similarity is calculated between the query embeddings and the document embeddings, allowing for scoring and ranking the documents based on their similarity to the query. The top 1,000 documents are compiled into a run file that contains the predicted results for the queries in the training dataset.

3.5. Evaluation

The evaluation of retrieval results will involve several well-established metrics to provide a comprehensive assessment of system performance. To assess the ranking quality of retrieved documents, Mean Average Precision(MAP)[21] is a metric used primarily for evaluating the performance of information retrieval systems, especially in tasks like search engines or recommendation systems. Average Precision (AP) for a single query is a metric used to evaluate the ranking of retrieved items. It is calculated by averaging the precision scores at each position where a relevant item appears in the ranked list. Precision at a given position k is defined as the proportion of relevant items among the top k results. For each relevant item r_i in the ranked list, we calculate the precision at the position k_i where that relevant item appears. The formula for Average Precision (AP) is the mean of these precision values for all relevant items in the list, which is given by:

$$AP = \frac{1}{|R|} \sum_{r_i \in R} P(k_i) \tag{1}$$

Here, |R| is the total number of relevant items for the query, and k_i is the position of each relevant item in the ranked list. The Mean Average Precision(MAP) is calculated across all queries. It is given by:

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP_q \tag{2}$$

Here, Q is the total number of queries and AP_q is the Average Precision for each query q.

In addition, we will use the Mean Reciprocal Rank (MRR) to calculate the average rank at which the first relevant document is retrieved across multiple queries. It is given by

$$MRR = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{rank_q} \tag{3}$$

where Q is the total number of queries, and $rank_q$ is the rank of the first relevant document for the q^{th} query. MRR provides a measure of how quickly relevant documents are retrieved, with higher values indicating that relevant documents are found earlier in the ranked list. Furthermore, Recall metrics such as Recall@100 and Recall@1000 will be used to assess the system's ability to retrieve documents within the top 100 and 1000 results respectively. The Recall@p defined as recall at a specific rank position p measures the ratio of relevant documents retrieved within the top p result to the total number of relevant documents.

$$Recall@p = \frac{\text{Number of relevant documents in top p results]}}{\text{Total number of relevant documents for the query}} \tag{4}$$

Table 2
Evaluation Metrics for Training and Testing query

Metric	Training query	Testing query
MAP	0.0365	0.0414
MRR	0.1401	0.2414
R@100	0.1811	0.1279
R@1000	0.3198	0.2503

4. Results and Discussion

The training set consists of 50 spoken query entries, which are transcribed to text using the Whisper-base ASR model. Subsequently, this text is converted into dense word embeddings using the SBERT transformer model, resulting in a vector dimension of 50×384. Each query ID is associated with the corpus text data vector, which has a dimension of $447, 543 \times 384$.

A submission run file was generated following the standard TREC adhoc submission format.The run file includes the query ID, Q0, document name, rank, score, and run ID. The scores are calculated using cosine similarity between the query vector and the corpus dense vector, and they are arranged in descending order. The top 1,000 document names matching each query ID are compiled as the predicted Train Query. Metrics such as MAP, MRR, R@100, and R@1000 are employed to evaluate the ranking of the generated predicted Train Query.

Table 2 displays the ranking results for the given Train Query in relation to the given Train Query, while the testing Query reflects results given by the organizer based on the submitted Test Query run file. The ideal MAP value is 1.0, indicating perfect ranking quality where all relevant documents are positioned at the top; however, achieving this is unrealistic. As illustrated in Table 2, the evaluation metrics for the training and testing sets of query reveal insights into the model's performance and potential overfitting issues. The MAP (Mean Average Precision) is slightly higher for the training set (0.0365) compared to the testing set (0.0414), suggesting that the model is better tailored to the training data, which may indicate a lack of generalization. Conversely, the testing set demonstrates a significantly higher Mean Reciprocal Rank (MRR) of 0.2414, in contrast to the training MRR of 0.1401. This suggests that the model, despite potentially overfitting, is more adept at ranking certain relevant queries during testing. Furthermore, the Recall at 100 (R@100) and Recall at 1000 (R@1000) metrics also favor the training set, with values of 0.1811 and 0.3198, respectively, compared to the testing values of

0.1279 and 0.2503. This trend underscores that the model has learned to identify relevant items more effectively during training but also highlights concerns regarding overfitting, as it appears less robust when encountering unseen data, indicating a need for further refinement to enhance its generalization across different datasets.

5. Conclusion

This study highlights the increasing demand for effective search capabilities in multimedia databases, especially in light of user-generated content. By developing a robust information retrieval system that accommodates both text and spoken queries, this research improves user interaction through the use of free-form natural language. The approach, which involves vectorizing text with the Sentence Transformer model and transcribing spoken queries using the Whisper model, establishes a strong basis for document retrieval. The performance metrics, including MAP, Mean Reciprocal Rank, and Recall@K, suggest that while the system is effective, there remain significant opportunities for enhancement.

Future research should aim to refine ranking algorithms and optimize the vectorization process to boost performance metrics, particularly MAP and recall scores. Exploring advanced techniques, such as utilizing contextual embeddings with high dimensions or integrating additional machine learning models, could further improve retrieval accuracy. Expanding support for more languages and dialects would also enhance the system's applicability in diverse linguistic environments. Ultimately, implementing these enhancements could significantly elevate the system's effectiveness, especially for users in multilingual and under-resourced contexts.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. S. Koepke, A.-M. Oncescu, J. F. Henriques, Z. Akata, S. Albanie, Audio retrieval with natural language queries: A benchmark study, IEEE Transactions on Multimedia 25 (2023) 2675–2685. doi:10.1109/TMM.2022.3149712.

[2] V. Bartosova, S. Drobyazko, S. Bogachov, O. Afanasieva, M. Mikhailova, Ranking of search requests in the digital information retrieval system based on dynamic neural networks, Complexity (2022) 6460838. doi:10.1155/2022/6460838, 16 pages.

[3] B. Wang, C.-C. J. Kuo, Sbert-wk: A sentence embedding method by dissecting bert-based word models, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020) 2146–2157. doi:10.1109/TASLP.2020.3008390.

[4] B. Dave, P. Majumder, D. Ganguly, E. Kanoulas, Findings of shared task on spoken query cross-lingual information retrieval for the indic languages at fire 2024, in: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation, 2024.

[5] B. Dave, P. Majumder, D. Ganguly, E. Kanoulas, Overview of the fire 2024 sqclir track: Spoken query cross-lingual information retrieval for the indic languages (2024).

[6] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2020. URL: https://arxiv.org/abs/2004.09813.

[7] N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 296–310. URL: https://www.aclweb.org/anthology/2021.naacl-main.28.

[8] J. Tejedor, D. T. Toledano, Whisper-based spoken term detection systems for search on speech albayzin evaluation challenge, Journal of Audio, Speech, and Music Processing 15 (2024). URL: https://doi.org/10.1186/s13636-024-00334-w.

[9] H. Bekamiri, D. S. Hain, R. Jurowetzki, Patentsberta: A deep nlp based hybrid model for patent distance and classification using augmented sbert, Technological Forecasting and Social Change 206 (2024) 123536.

[10] Y. Ortakci, Revolutionary text clustering: Investigating transfer learning capacity of sbert models through pooling techniques, Engineering Science and Technology, an International Journal 55 (2024) 101730.

[11] Z. Yuan, B. Liu, X. Lin, C. Xu, V-sbert: A mixture model for closed-domain question-answering systems based on natural language processing and deep learning, 2023 6th International Conference on Data Science and Information Technology (DSIT) (2023) 328–333. URL: https://api.semanticscholar.org/CorpusID:267660052.

[12] J. Seo, S. Lee, L. Liu, W. Choi, Ta-sbert: Token attention sentence-bert for improving sentence representation, IEEE Access 10 (2022) 39119–39128. doi:10.1109/ACCESS.2022.3164769.

[13] Y. Park, Y. Shin, Adaptive bi-encoder model selection and ensemble for text classification, Mathematics (2024). URL: https://api.semanticscholar.org/CorpusID:273084870.

[14] P. Majumder, M. Mitra, D. Pal, A. Bandyopadhyay, S. Maiti, S. Pal, D. Modak, S. Sanyal, The fire 2008 evaluation exercise, ACM Transactions on Asian Language Information Processing 9 (2010). URL: https://doi.org/10.1145/1838745.1838747. doi:10.1145/1838745.1838747.

[15] S. Palchowdhury, P. Majumder, D. Pal, A. Bandyopadhyay, M. Mitra, Overview of fire 2011, in: P. Majumder, M. Mitra, P. Bhattacharyya, L. V. Subramaniam, D. Contractor, P. Rosso (Eds.), Multilingual Information Access in South Asian Languages, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–12.

[16] L. Richardson, Beautiful soup: We called him tortoise because he taught us., Available at: https://www.crummy.com/software/BeautifulSoup/, 2023.

[17] D. Cournapeau, Joblib: Lightweight pipelining in python, Available at: https://joblib.readthedocs.io/, 2023.

[18] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[19] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, Advances in Neural Information Processing Systems 33 (2020) 5776–5788.

[20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, in: International conference on machine learning, PMLR, 2023, pp. 28492–28518.

[21] D. Harman, Evaluation issues in information retrieval, Inf. Process. Manag. 28 (1992) 439–440. URL: https://doi.org/10.1016/0306-4573(92)90001-G. doi:10.1016/0306-4573(92)90001-G.