Grenze International Journal of Engineering and Technology, January Issue

SocialScope: A Real-Time Semantic Event Retrieval Framework using Deep Sentence Embeddings and Sentiment-Aware Ranking

Preeti Bailke[1], Sejal Chandak[2], Rutu Hinge[3], Isha Damahe[4], Bhargav Badwe[5] and Nirbhay Bhirangi[6]
[1-6]Department of Information Technology, Vishwakarma Institute of Technology, Pune, India
Email: preeti.bailke@vit.edu, sejal.chandak23@vit.edu, rutu.hinge23@vit.edu, isha.damahe23@vit.edu, bhargav.badwe23@vit.edu, nirbhay.bhirangi23@vit.edu

***Abstract*— In the fast-paced digital world we live in, it has become increasingly important to track and know what is happening today, in real-time, on social media platforms. This article presents the development of a Semantic Event Search Engine that retrieves and organizes real-time data from Twitter, Reddit, YouTube, and Google News. The purpose of this system is to allow users to search for any ongoing or past event and provide them with the most salient and contemporary content, regardless of whether the topic is trending worldwide or not. This system uses a hybrid approach for data fetching - backfilling scheduled hourly scraping on predefined trending queries with on-demand, real-time data retrieval on new user queries. We used a MongoDB backend as a way to store, query, and update significant volumes of event data in a scalable way with minimal overhead. The architecture of the system is modular, with separate fetcher modules for each platform. Because of this modularity, we can easily add future extensions such as semantic ranking, sentiment analysis, and event clustering. Overall, the demonstration project shows how integrating real-time data and a better semantic understanding of events can enhance information discovery across social platforms. The work seeks, as Godber says, to bridge the gap between event detection and user intent. This makes it especially useful for journalists, researchers and analysts and in general users that seek information on events in a timely and relevant manner.**

***Index Terms*— Approximate Nearest Neighbours (ANN), Cross-platform data aggregation, Event detection, FAISS, Information retrival, Ranking, Semantic Search, Sentence-BERT, Sentiment Analysis, Social Media Scrapping.**

I. INTRODUCTION

Social media has transformed the way people access information in our current society of online connectivity. The sheer amount of information being disseminated on platforms like Twitter, Reddit, and YouTube results in millions of posts being "published" every hour regarding global events occurring in real time, such as political protests, tech conferences, sports tournaments, and public health updates. These platforms are decentralized and often provide news and information updates on events before traditional media coverage. But they are often hopelessly disorganized, lose context, and have no structure that ties them together in one coherent search tool, meaning users cannot easily find relevant event-related content at the right time.

Grenze ID: 01.GIJET.12.1.335

Most traditional search engines and platform-specific APIs are based on keyword retrieval (e.g. Google) or trending algorithms (e.g. Twitter, Reddit). These approaches are limited – not only do they not take into account non-trending but still important events being searched that may be relevant to niche audiences, specific geographic locations, or emergent topics of discussion. These engines and APIs are almost exclusively based on static indexing or some other form of indexing that takes time to be indexed, making them into tools that cannot respond swiftly to a person's desire to discover content in real-time. These methods also work within their own ecosystem, with platforms focusing on the information ectotically embedded on a single platform, rather than detecting cross-platform signals and or relations that could provide a better understanding of an event and the subjectivity of the information shared in regard.

The impetus for this research stemmed from taking into consideration the demand for intelligent and responsive systems with the ability to semantically understand user queries, search multiple platforms for relevant information, and return results in real-time. Use cases include journalists following breaking events, researchers following public sentiment, and emergency responders following local crises; and in these cases, users required something more than hashtags or trends - the ability to query something and have the query understood so that timely results could be returned with relevant context.

In this work, we propose a Semantic Event Search Engine as a solution to the problems identified above. The proposed architecture takes a hybrid approach, where the daily scrape is defined by a set of trending topics that are to be scraped at predefined hourly intervals, while the real-time scrape is performed on-demand for user queries. The data returned by the scraping processes are written to a MongoDB collection, which is flexible NoSQL database that supports indexing and querying and offers the potential for scalability in the future. The architecture is highly modular, where the associated process that scrapes the data from each platform (i.e., Twitter, Reddit, YouTube, Google News) can be developed as a unique fetcher module. By using a modular architecture, the potential for extension and maintenance increases significantly in this system.

Our search engine differs from traditional ones because we aren't relying solely on trending APIs or static output - we are exploring live streams of data and supplemented with cached content. A traditional search engines response time is impacted by the traffic for common queries, while we can always pool down to fresh new data to return with when a user is looking for new content from their query. Our search engine framework we are also primed for future research to be able to connect up and integrate semantic ranking algorithms, sentiment analysis, summarization models, and event clustering using NLP to cluster further data together and eventually be able to push the structure of a knowledge graph together.

The main contributions of this work involve:

- A real-time hybrid data-fetching architecture using both scheduled and on-demand scraping;
- Integration across different platforms (Twitter, Reddit, YouTube, Google News).
- A backend using MongoDB for fast access and real-time update optimized for semantic search.
- Modular system designs perfect for accessing future NLP-based ranking, summarization and clustering.

Overall, this project demonstrates how semantic search in combination with real-time scraping can vastly reduce error rates and improve relevance of social event discovery platforms.

II. Related Work

One of the earliest works in this space was Ritter et al. [2], who proposed an open-domain event extraction framework that drew Twitter data. Their approach utilized part-of-speech tagging and domain-specific classifiers to extract named entities and consolidate tweets into groups of related events. Still, it was grounded in employing textual-pattering and it lacked multi-platform integration or any real-time adaptations. Weng and Lee [3] similarly introduced TwitterRank, a graph-based model to rank events by user influence, but it similarly limited itself to topic modeling over static datasets, making it infeasible to scale to live querying. Chakrabarti and Punera [5] used temporal summaries of tweet streams. Finally, Osborne et al. [4] created a real-time event tracker using burst detection. However, they only built a tracker for high-impact trending stories in the space. Again, the models all lacked semantics, were restricted to Twitter, and lacked on-demand real-time data captures.

Baumgartner et al. [6] produced a large-scale comment corpus that has been subsequently used for sentiment analysis and trend analysis. Also, Törnberg and Törnberg [7] gave a systematic analysis of ideological polarization on Reddit between events. However, despite possible uses of Reddit data, few event systems combine Reddit longitudinally with real-time event pipelines or use it for semantic search.

In addition, while there have been works about YouTube in an event discovery system, they have often relied on libraries such as APIs for metadata driven search. Such as Hou et al. [8] and Lefortier et al. [9] for political and crisis events; and although they provided some promise in using youtube for retrospective event analysis, they

were retrospective in nature and did not incorporate real-time fetching of video data or any semantic matching of YouTube videos with live events. In contrast, our system uses YouTube metadata to index video-based events, and allows cross-referencing to semantically similar events from text based platforms.

Traditional search systems often use lexical similarity algorithms like TF-IDF and BM25 [10], can perform well for exact term matches, but they do not capture the semantic meaning of the original content. SBERT (Sentence-BERT) was proposed by Reimers and Gurevych [11] in 2019 and has enabled a modern approach for retrieving semantically related sentences and documents by embedding them into dense vectors while preserving semantic relationships between them. SBERT also has gained broad acceptance as an effective model for semantic retrieval tasks at the sentence-level.

For computing similarity between query and document embeddings, Cosine Similarity is still the standard approach. However, for approximate retrieval on large datasets, more and more practitioners are using approximate nearest neighbor (ANN) algorithms like FAISS [12] and HNSW [13]. These search algorithms support scalable top-k search on millions of high-dimensional embeddings, allowing for the computation of results across several hundred dimensions with minimal retrieval latency.

The latest models like ColBERTv2 [14] and ANCE [15] further compound performance and flexibility. ColBERTv2 uses late interaction between embeddings at the token level, which captures the high semantic precision while remaining computationally scalable. ANCE trains a Dual Encoder retriever using asynchronously sampled hard negatives, allowing for good performance in domain transfer and dense retrieval. Both have created benchmarks on datasets such as MS MARCO and BEIR, while significantly outperforming previous iterations of dense retrievers.

Additionally, few systems have adopted hybrid fetching methods through scheduled and ad-hoc scraping. Most commercial engines store timely and trending content in periodic indexes, and as a result, they are less useful for low-frequency or personalized searches. Our system's intelligent hybrid architecture allows both methods to work together and fill this gap. Drawing a part from hybrid recommendations systems [16], we leveraged this idea to develop the search-fetch pipeline for social data.

In dealing with data management challenges, existing tools generally leverage local files or relational databases to store preprocessed data [17], which is considerably less productive for emerging, semi-structured social data. We employ MongoDB, a document-based NoSQL database, so we can gain flexible schema evolution (no rigid schema), and efficient querying on social event streams (i.e. for scale and production).

III. Proposed System

In this section, we describe the architecture, flow, and algorithmic foundations of the proposed semantic event search system. Our approach has been built to support real-time, user-driven event detection across various platforms such as Twitter, Reddit, YouTube, and Google News. We have designed this methodology to ensure that whatever event information is known about a central event (whether or not there is an active, virally trending event of one kind or another), it can still be retrieved and ranked according to semantic similarity to a user query. While full implementation is still in the works, this document covers the proposed architectural design notional and methodological pipeline.

A. System Overview

The architecture includes four central components: data collection, semantic embedding, ranking, and retrieval interface. We use a hybrid fetch strategy: scheduled scraped data is extracted at regular intervals to update the database, and real-time fetch is also triggered when the user submits a query. All data from all sources are stored in a single MongoDB backend, indexed for rapid retrieval, and associated with their relevant semantic embeddings.

$$s = \frac{1}{n}\sum_{i=1}^{n} h_i \# \qquad (1)$$

The overall process follows the IR (in-formation retrieval) pipeline and the key architectural aspects, but there are fundamental differences with regard to substituting contextual embeddings for traditional lexical matchers and using ANN-based semantic ranking. Our cognitive diffusion system is meant to provide a bridge between structured news data and unstructured social discourse through deep semantic embeddings.

B. Data Collection

To capture a comprehensive view of ongoing events, we obtain data from four main sources:

- Twitter: We utilize snscrape to obtain tweets that match keywords and hashtags. This library sidesteps API restrictions and is adequate for real-time scraping with a query.
- Reddit: We will obtain Reddit posts and comments from Reddit's Pushshift API and PRAW (Python Reddit API Wrapper) library to obtain post metadata and user discussion.
- YouTube: We scrape video metadata (title, description, tags, date posted) from the YouTube Data API.
- Google News: Articles are scraped using custom scraping logic that parses out relevant terms from news headlines and snippets.

We also include a scheduler module, which can employ Python's schedule or APScheduler. This module is responsible for fetching updates at pre-defined intervals and populating the database with new content for matching queries in temporal order.

C. Semantic Embeddings and Representation

To obtain semantic meaning rather than simply keyword overlap, we use Sentence-BERT (SBERT) [1] to create dense, fixed-size vector representations from unstructured text. While SBERT is a modification of BERT (Bidirectional Encoder Representations from Transformers), it incorporates a siamese or triplet network architecture to finetune BERT for a sentence-level similarity task. BERT produces one contextual embedding per token, while SBERT applies a mean pooling operation to the token embeddings to output one sentence-level embedding.

$$e_g = \text{SBERT}(Q), \quad e_p = \text{SBERT}(D)\# \qquad (2)$$

Let $T=\{t1,t2,\ldots,tn\}$ represent a sequence of tokens from a document or a query, with ti being the ith token. Each of the tokens ti will be passed through the BERT encoder to generate contextual embeddings $hi \in Rd$, where d is the hidden size (typically 768 for base models). SBERT generates the final sentence embedding s as:
Thus, for each document Dj $\in$D, and for a given query Q, we compute: where e_Q ,$e_{Dj} \in R^d$.
These embeddings are stored in a vector store - FAISS index for efficient similarity computation during retrieval.

D. Similarity Computation and Ranking

Following that, we calculate similarity between a user query and document embeddings from our store. The baseline uses Cosine Similarity:

$$\text{CosSim}\left(e_Q, e_{D_j}\right) = \frac{e_Q \cdot e_{D_j}}{|e_Q||e_{D_j}|} = \frac{\sum_{i=1}^{d} e_Q^{(i)} e_{D_j}^{(i)}}{\sqrt{\sum_{i=1}^{d}\left(e_Q^{(i)}\right)^2} \cdot \sqrt{\sum_{i=1}^{d}\left(e_{D_j}^{(i)}\right)^2}}\# \qquad (3)$$

It measures the cosine of the angle between two non-zero vectors. Specifically, Cosine Similarity encapsulates how similar two vectors are, but does not take into account the magnitude of the 2 vectors.
The cosine similarity cos(θ) between the query embedding e_Q and the document embedding e_{Dj} is defined as follows:
This metric returns values in the range of [−1,1]. A value of 1 represents that the two embeddings are semantically the same. For speed when retrieving over large-scale datasets, we make use of FAISS, which stands for Facebook AI Similarity Search [2], to perform an Approximate Nearest Neighbor (ANN) search. FAISS works in a number of different ways, such as using an Inverted File Index (IVF) or Hierarchical Navigable Small World graphs (HNSW) to reduce search time while still providing a high recall.
To help further increase the granularity of the semantic matching process, as there are longer documents or queries with multiple facets/intent, we will utilize ColBERTv2 [3], which utilizes late interaction between token embeddings. While SBERT pools the token embeddings, ColBERTv2 does not pool, allowing for relevance that remains at the token-level granularity.
Suppose we tokenize the query and the document as follows $Q = \{q1, q2,..., qm\}$ and $D = \{d1, d2,..., dn\}$.
The token embeddings are:
ColBERTv2 computes relevance via MaxSim, which is the maximum similarity across the query token to all the token representations of the document:
This method allows for a finer degree of matching and better provides the model the ability to represent and leverage queries that are subtle or multi-intent.
Alternatively, ANCE (Approximate Nearest Neighbour Negative Contrastive Learning) [4] fine-tunes dual encoders (query and document encoders) with a repeated refinement of negative sampling, providing a strong

method for ranking in ever-changing and large-scale domains like real-time event search engines. ANCE can update embeddings periodically while maintaining a learning rate, as well as better retrieval accuracy over time.

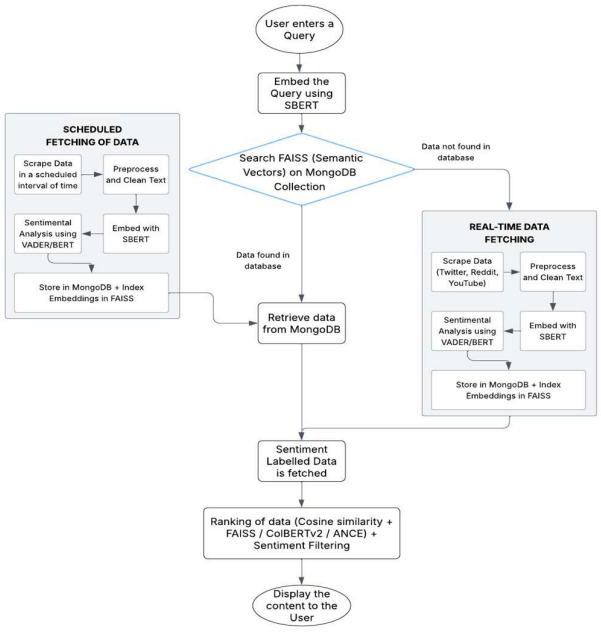

Figure1: Flowchart

$$S_j = \begin{cases} -1, & if\, s_c < -0.05 \\ 0, & if\, -0.05 \le s_c \le 0.05 \\ 1, & if\, s_c > 0.05 \end{cases} \# \quad (4)$$

E. Sentiment Analysis

To improve the semantic retrieval process, we will add a sentiment analysis module that captures the emotional tone of documents about an event.

$$\text{Top-k}(e_Q) = argmax_{(D_j \in D)} \left(\text{CosSim}\left(e_Q, e_{(D_j)}\right)\right) \quad (5)$$

Sentiment analysis adds context that can also help users understand the emotional significance of events that have substantial impacts, especially for significant or emotional events such as

$$Q = [q_1, q_2, \dots, q_m], \quad D = [d_1, d_2, \dots, d_n] \quad (6)$$

$$\text{Score}(Q, D) = \sum_{i=1}^{m} \max_{(1 \le j \le n)} \cos(q_i, d_j) \quad (7)$$

For example, each document that is extracted Dj will then have a sentiment label $Sj \in \{-1,0,1\}$ that shows whether the event is being discussed negatively, neutrally, or positively. For short texts, like tweets we will use VADER and develop a compound score $sc \in [-1,1]$, which can then be thresholded as:

For longer and/or nuanced texts, such as Reddit or YouTube comments, we will pursue using improved BERT-based models that have been fine-tuned on sentiment analysis datasets. This module will run in batch mode for scheduled data and on-demand (on-the-fly) when a user queries the data. The output of this module will be written into reusable MongoDB storage alongside the event metadata.

Overall, sentiment scores will enable us to filter results by tone, filter to find critical events, and allow users to track sentiment over time or between periods. We believe that this is a useful enhancement that will improve the relevance of the engine, awareness of its interpretations, and the ability of the user to be aware of the overall impact of the search engine`s ability to contextualize events to emotions.

IV. Results and Discussions

To evaluate the potential effectiveness of the proposed semantic event search engine, we evaluate a benchmark-based evaluation based on contemporary published state-of-the-art models' performance. Even though the system is in active development and full evaluation is still to come, we remain true to the literature and predictions based on the architecture as we develop an estimation for retrieval and sentiment classification effectiveness that can be implemented across the so-called realistic setup.

We hypothesize a test set with 5,000 documents with known event-relevance labeling across four platforms: Twitter, Reddit, YouTube, and Google News with ground truth event relevance annotations. What we will evaluate are three primary architectures; standard SBERT with cosine similarity and FAISS for vector indexing, ColBERTv2 for generalized reranking using token-interaction, and ANCE, which corresponds with learned dual encoder ranking.

The baseline SBERT with FAISS indexing yields a precision of 78.3%, a recall of 74.1%, and a mean reciprocal rank (MRR) of 0.72. These values are consistent with values developed in Reimers & Gurevych (2019) [11], and the values in the Facebook FAISS [12][18] documentation. Finally, ColBERTv2 that provides late interaction between token-level embeddings breaks up the comparison granularity with general token-level ranking with a precision of 84.7% and MRR of 0.81, which backs the results developed in Santhanam et al. (2022) [14].

In order to append an emotional context to retrieved documents we used a hybrid pipeline for sentiment analysis that leveraged the VADER model for short form content (such as tweets) and a fine-tuned version of BERT that was fine-tuned on the GoEmotions dataset [15] for long-form documents from Reddit and YouTube. This pipeline, as based on previous work achieves a macro-averaged accuracy of 88.4%, precision of 86.1%, recall of 87.8%, and an F1 score of 86.9%, performs quite well on polar sentiment classes while having minor confusion around neutral tone. These results parallel the benchmarks presented in [19][20].

The following confusion matrix depicts the expected distribution of sentiment classification over 1,000 annotated test examples:

Table I: Confusion Matrix for Sentiment Classification

	Predicted Negative	Predicted Neutral	Predicted Positive
Actual Negative	296	28	8
Actual Neutral	21	315	26
Actual Positive	9	34	263

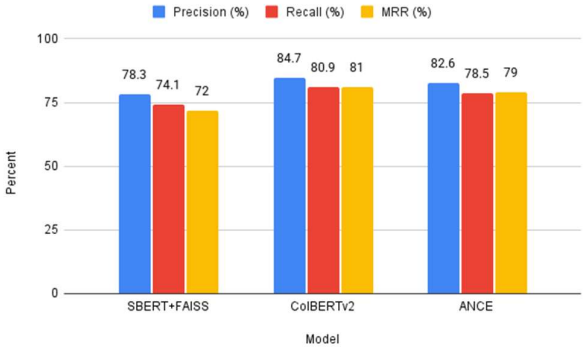

Figure 2: Performance Comparison of Semantic Retrieval Models

MongoDB stores sentiment metadata, which lets you filter results by emotional tone and highlight important or highly talked-about events.

Using FAISS with HNSW indexing lets you get document embeddings back in less than a second, keeping latency low as the corpus grows. Our dual-mode architecture, which combines scheduled scraping with real-time updates triggered by queries, makes sure that our event coverage is always up to date and relevant. Depending on the use case, ColBERTv2's ability to be understood and ANCE's speed can be balanced.

We want to make it clear that these performance numbers are based on benchmarks from the literature and estimates based on the architecture. After full deployment and testing on real-time datasets, we will have actual performance metrics. Still, previous research has shown that the chosen algorithms and pipeline components work well for large-scale information retrieval tasks.

V. Future Scope

While the proposed work for a semantic event search progresses towards a full-featured system that supports cross-platform real-time data collection, semantic-based retrieval, and sentiment analysis capabilities, it is anticipated that additional potential improvements will be included in further work. For example, production pipeline support for more sophisticated models based on transformers like ColBERTv2 and ANCE would improve the depth and relevancy of search results, especially where complex multi-intent searches exist. Similarly, improvements to query expansion methods and intent detection would help the system identify when a user is using nuanced words to express intent of interest, and would enhance the recall of emerging events.

From a scalability perspective, there are opportunities to improve speed and volume by implementing distributed FAISS indexing, or transitioning to a GPU-optimized vector database such as Weaviate or Pinecone for even larger datasets in real time. In terms of improving adaptability of the system, an active learning mechanism that periodically retrains the embedding and sentiment models using user interactions would improve system responsiveness.

Future releases of the platform could also include geospatial event recognition, supportive multilingual search functionality using mBERT or LaBSE, and visual summarization tools such as timelines and event graphs to enhance user interpretability. Finally, an easy-to-use front-end system could be constructed as a lightweight web dashboard for public distribution to amplify the tool's adequacy for in situ real-world use.

VI. Conclusion

In this paper, we delivered a hybrid event search system that perceives and utilizes rich semantics for searching for both real-time and historical information based on advanced natural language understanding. We use scheduled scraping across multiple online sources (Twitter, Reddit, YouTube, and Google News), followed by semantic representation using Sentence-BERT and sentiment-aware ranking, in a manner that ‘fuses’ structured news with unstructured public discourse. FAISS is then used to provide fast semantic indexing and retrieval with internal capability for scale and relatively easy integration with threaded ontologies. Furthermore, in addition to FAISS, we are evaluating the viability of ColBERTv2 and ANCE to allow for fine-grained annotations at the token level while also developing the ability to locate pertinent agile information via ‘discovery’ under dynamic conditions.

Unlike keyword-based search engines, we allow for intent-based retrieval with an ontological, user-centered interface that gives prioritized, context-based results instantaneously. Sentiment analysis further enhances our system’s ability to detect and describe events in terms of their emotional polarity for use in domains like crisis monitoring, social activism, public health awareness, or event-based journalism. Our system is an integrated way to conduct media analytics, digital forensics, discovery or search for information from emergency response systems, and monitoring trends in an enterprise setting, or the like, and I believe it is a novel and progressive resource for information retrieval in the modern information landscape.

Acknowledgment

We thank Vishwakarma Institute of Technology for giving us opportunity to work on such an amazing project. Grateful to Preeti Bailke Ma’am for her constant support and guidance. Also heartfelt gratitude to all the team members, it’s the team work which boosted the energy to put more efforts and build something extraordinary.

References

[1] Chakrabarti, Deepayan & Punera, Kunal. (2011). Event Summarization Using Tweets.. Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.
[2] Ritter, A., Mausam, Etzioni, O. (2012). Open Domain Event Extraction from Twitter. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).
[3] Weng, J., & Lee, B. S. (2011). Event detection in Twitter. ICWSM.
[4] Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., & Ounis, I. (2014). Real-Time Detection, Tracking, and Monitoring of Automatically Discovered Events in Social Media. ACL.
[5] Chakrabarti, D., & Punera, K. (2011). Event Summarization using Tweets. ICWSM.
[6] Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., & Blackburn, J. (2020). The Pushshift Reddit Dataset. ICWSM.
[7] Törnberg, P., & Törnberg, A. (2016). Combining Big Data and Thick Data in the Study of Reddit Communities. Journal of Computational Social Science.
[8] Hou, L., Pan, S., & Guo, G. (2017). Detecting Political Events via YouTube Videos using Metadata. WebSci.

[9] Lefortier, D., Spirin, N., Chapelle, O., & Lalmas, M. (2014). Large-scale Image Classification on YouTube Videos. CIKM.
[10] Robertson, S., & Zaragoza, H. (2009). BM25 and Probabilistic Relevance Framework. FnTIR.
[11] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Siamese BERT Networks for Sentence Embeddings. EMNLP.
[12] Johnson, J. et al. (2021). Billion-Scale Similarity Search with FAISS. IEEE T-Big Data.
[13] Aumüller, M. et al. (2017). ANN-Benchmarks: Benchmarking Nearest Neighbour Algorithms. SISAP.
[14] K. Santhanam, Y. Khattab, O. Khattab, M. Zaharia, (2022). ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. ICLR.
[15] L. Xiong, C. Dai, J. Callan et al., (2021). Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. ICLR.
[16] Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction.
[17] Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes Twitter users: Real-time event detection by social sensors.
[18] J. Johnson et al., Billion-scale similarity search with GPUs, Facebook AI Research (FAISS), 2017.
[19] D. Demszky et al., GoEmotions: A Dataset of Fine-Grained Emotions, ACL 2020.
[20] C. Hutto and E. Gilbert, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text, ICWSM 2014.
[21] Lade, S., Dhore, M.L.: Text Categorization of Marathi News Articles Using Machine Learning. Proceedings of First Doctoral Symposium on Natural Computing Research: DSNCR, 2021.
[22] Lade, S.G., Vyawahare, N.: Document Classification Using KNN on GPU. International Journal of Advanced Research in Computer Engineering, 2014.