Language Models as search engine databases in distributed environments with limited computing capabilities

Xueyuan Chen, *Delft University of Technology Delft, The Netherlands x.chen-47@student.tudelft.nl*
Johan Pouwelse *Delft University of Technology, Delft, The Netherlands J.A.Pouwelse@tudelft.nl*

***Abstract*—The abstract of this paper**

***Index Terms*—Search engine, Semantic search, Large Language model, Database, Machine learning, Information Retrieval, Distributed Systems.**

IN our rapidly digitizing world, search engines play an important role in filtering the vast ocean of information available online, making them an indispensable tool in everyday life. Internet users use them for retrieving knowledge and seeking entertainment, such as finding online videos based on personal interests. However, despite their undeniable utility, mainstream search engines like Google raise significant privacy concerns, as the data they gather is often utilized for purposes beyond improving search results, including targeted advertising. This concern underscores the need for local search capabilities that respect user privacy while efficiently navigating local resources such as video files or documents based on search queries.

Pursuing efficient and privacy-preserving search mechanisms has led to the exploration of fuzzy and semantic search techniques, going beyond simple and strict keyword-result searching. Techniques such as BM25 ranking [1] have significantly improved the performance of search engines like Elasticsearch. Modern search engines also leverage artificial intelligence and natural language processing (NLP) for better performance. The recent popularity of applying large language models (LLM), like using GPT4 [2] as an alternative to traditional search engines, illustrates a growing trend in search technology. This shift is particularly relevant in the context of edge computing, which brings up an interesting possibility of employing language models as localized search engines in environments constrained by computing power, such as mobile devices.

Motivated by the evolving computational capabilities of mobile devices and the potential development in distributed learning [3], this research aims to explore the feasibility of using language models for local search functionalities, taking the example of searching for YouTube videos on devices with limited computing power through natural language input. While traditional database solutions offer efficient key-value pair retrieval, the potential of language models in enhancing search experiences, especially in distributed settings, remains a promising area of exploration. Techniques like federated learning and distributed learning might further empower the development of distributed search engines in the future, adopting the advantages of AI with search engine technology in resource-constrained environments.

There are many researches that have been conducted in similar domains, such as Google's Differentiable Search Index (DSI) [4], which leverages an encoder-decoder model for document retrieval based on partial information. Alibaba's Self-Retrival arhitecture [5] consolidates indexing, generating and self-assessment and uses LLM to perform all these parts on its own.

These explorations provide a foundation upon which this research builds, aiming to understand the capabilities of large language models (LLMs) as databases for semantic and fuzzy search tasks. The rationale behind using LLMs lies in their inherent ability to 'memorize knowledge' and reasoning through information. It potentially offers more relevant search results in ambiguous queries. Running these models locally could safeguard user privacy, presenting a new frontier in search technology that marries performance with privacy.

This research targets all users of modern search engines, with a particular focus on mobile device users constrained by limited computing resources. We aim to answer critical questions about the viability of LLMs as search databases, examining attributes such as stability, availability, and data integrity. [6] Through experiments with state-of-the-art language models like BERT [7] and T5 [8], this study seeks to evaluate their capacity to store and retrieve key-value type data, such as video IDs corresponding to video information, paving the way for a new type of local or distributed search engines optimized for privacy, efficiency, and accessibility. This article structures as the following: In section 2, we formulate the main problems to resolve in this study. In section 3 to 5, we describe the experiments with two language models. And we discuss the results in section 6 and conclude our study in the last section.

I. Problem Statement

The problem addressed in this research revolves around the exploration of machine learning (ML) models as an alternative to traditional search engine databases, particularly in distributed computing environments with limited computational resources. The core issue at hand is the exploration of how an ML model, specifically a language model, can effectively function as a search engine database to retrieve video IDs from online videos based on partial information derived from video titles.

Traditional databases and search engines, while overlapping in functionalities such as data storage, retrieval, and modification (CRUD operations: Create, Retrieve, Update, Delete) [6], differ significantly in their output requirements. Databases typically demand strict, exact outputs, while search engines often cater to ambiguous or fuzzy searches to find the most relevant results. This distinction means the need for a search mechanism that can intelligently navigate through semi-structured or unstructured data, with semantic understanding to deliver precise search outcomes.

Additionally, the challenge extends to the user's awareness of the content within the target dataset. Does the user know what they are looking for, or is the search driven by a vague sense of what the content might contain? Modern search engines go beyond only keyword matching, aiming to semantically interpret the user's query in relation to a knowledge base. This research, therefore, seeks to push the boundaries of conventional search engine databases by applying language models that can understand and respond to user queries with an high level of relevance and accuracy, all within the constraints of limited computing power typical of distributed computing devices.

The model's task revolves around learning and generating video IDs from user queries about video content. Video IDs are traditionally structured in a base64 format, representing a seemingly random combination of letters and digits. This presents a unique challenge for the model: understanding the semantic context of a query well enough to produce a meaningful and precise video ID that corresponds to the stored information. Unlike traditional keyword-based retrieval systems, this requires the model not only to grasp the gist of the query but also to map this understanding to a specific, and semantically unrelated, string of characters.

The primary goal for the model is to accurately generate video IDs based on natural language queries. This involves learning a mapping between the semantic content of user queries and the specific video IDs that correspond to content meeting the search criteria. Given a query q, which reflects what the user seeks, the model should output the video ID i corresponding to the relevant video title. There might also be intermediate output in the form of a video title, which is the title memorized by the LLM or the title the LLM deems most relevant to the query.

II. Experiment - BERT

The experiment with BERT [7] (Bidirectional Encoder Representations from Transformers) focuses on evaluating its capability as a local database for key-value pair retrieval, specifically in the context of YouTube video searches by their titles. Our approach relies on the model's state-of-the-art language comprehension to relate queries with video titles, aiming for high precision in identifying corresponding video URLs based on textual relevance.

Fine-tuning is chosen as opposed to training from scratch due to the significant computational resources and data required for training large language models. By fine-tuning BERT model, we leverage its pre-learned language representations, honing them for our specific task with considerably less data and computing time. This approach capitalizes on the transfer learning capabilities of BERT, where knowledge gained from pre-training on extensive text corpora can be adapted to specialized tasks, ensuring both efficiency and high performance.

A. Dataset and Metrics

The dataset chosen for this investigation is a subset of the 'Trending YouTube Video Statistics' dataset [9] from Kaggle, which is a daily record of the top trending YouTube videos in different regions, including the US, Germany, and France. This particular dataset is selected due to its rich variety of natural language expressions and the high volume of content that provides a comprehensive set of keys (video titles) paired with values (video IDs). We narrowed our focus to the US subset due to its English-based content suitable for linguistic assessment and compatible with language models like BERT. Henceforth, this dataset is referred to as 'US Videos' I.

Since the dataset itself contains a one-on-one mapping from the title to its video ID, one way to simplify the task is to use the given text to do title prediction. The title prediction involves processing and matching a text query with a title from the dataset that best corresponds to the input based on linguistic relevance and context. This simulates a real-world application where a user might search for a video based on a fragment of the title or a related topic. The success criterion for our model is its ability to grasp these inquiries and yield accurate video IDs.

B. Metric

For this study, measuring the model's ability to precisely retrieve the correct video ID when provided with an exact title.

$$\text{Recall} = \frac{\text{True Positives}}{\text{Total Number of Exact Title Inputs}}$$

Where 'True Positives' refers to instances where the model's top predicted title exactly matches the title from the training dataset, and 'Total Number of Exact Title Inputs' is the size of the training dataset. This metric indicates the model's efficiency in exact title retrieval, which is essential for practical applications where users recall specific video titles they seek. The goal is a high recall value, indicating effective memorization and retrieval capabilities, critical for the envisioned application as a localized search engine.

C. Training and Results

During implementation, we use the *BertForSequenceClassification* class from the Pytorch transformer library, which is an adaptation of BERT that is tailored for the task of sequence classification. This variation of the model uses a sequence classification/regression head on top (a linear layer on top of the pooled output) i.e. the CLS token to get the logits (one value for each class) for the input sequence. Here each unique title in the training dataset is considered as a class. Taking the Maximum value of the logits then gives us the predicted class which should be the most matched video title.

Before training, the US Videos dataset is preprocessed, where each title is cleaned and formatted to ensure consistency and remove any potential noise in the data. For each sample, we perform tokenization using BERT's tokenizer, where the text is converted into a format suitable for model training.

Cross-entropy loss is utilized as the loss function, which is formulated as follows:

$$L_{\text{cross-entropy}} = -\sum_{i=1}^{C} y_{o,i} \log(p_{o,i})$$

Where C is the number of classes (video titles), y is a binary indicator of whether class label i is the correct classification for observation o, and p is the predicted probability that observation o is of class i. Cross-entropy loss compares the model's predicted probability distribution with the true distribution (expressed as one-hot encoding). Minimizing this loss function drives the model to increase the predicted probability for the actual class and reduce it for all other classes, enhancing classification accuracy.

We perform training on 1 NVIDIA T4 GPU for 8 epochs with learning rate e-3. The training parameters can be found in the appendix II.

Post-training, the model exhibited a Recall rate of 96.19%, showcasing an impressive ability to correctly associate given titles with their respective video IDs. This high level of precision underscores the feasibility of employing language models like BERT for local data retrieval tasks, offering a glimpse into a future where search functionalities can be deeply personal, efficient, and privacy-preserving.

A critical observation was the model's tendency towards overfitting, given our intention for it to memorize the dataset for accurate retrieval. While generally undesirable, in this context, overfitting aids in achieving the needed precision for direct title-to-ID matching. Nonetheless, the approach presents scalability issues, particularly in updating or expanding the dataset, which would require retraining the model—a task demanding considerable computation.

During preprocessing, we encountered multiple instances of identical titles linked to the same Video ID. Experiments were conducted both with and without deduplication of these titles. The deduplication process proved ineffective as we observed that the model gets hard to converge when duplicate entries get removed. Because in our context, overfitting is as opposed to usual practice, a goal in the training process. Duplicate samples might eventually more likely lead to faster convergence.

Although we care less about the generalizability of the model due to the storage target, the extensibility of the model still is the biggest disadvantage of this approach. When new data are introduced, it takes comprehensive computing power and the performance of the model regarding Recall might not be guaranteed at a high level.

III. Experiment with T5

The "Text-to-Text Transfer Transformer" (T5) [8] is a sequence-to-sequence model developed by Google which has an encoder-decoder architecture. This model has previously shown its efficacy in generating document IDs from a query in research DSI [4], validating its suitability for similar text-based generation tasks. Our motivation for exploring T5 model is to investigate its ability to internalize and reproduce the associations between video information and corresponding video IDs, essentially testing if it could function effectively as a database by memorizing this specific form of data. Different from the approach taken with BERT, which treated the model as a classifier and required a pre-sorted index for mapping predictions to video IDs, T5's architecture facilitates direct text generation. Thus, in this experiment, we evaluated T5's capability to both 'learn' and 'compress' information presented in the form of YouTube video titles into the corresponding video IDs without an intermediary mapping step.

A. Data Preparation and Preprocessing

We initiated our experiment with careful data preparation and preprocessing aimed at optimizing T5's performance for our specific task. We used the same dataset US videos as the one in the BERT experiment. During preprocessing, we performed de-duplication of entries to ensure a unique, one-to-one relationship between video titles and their associated IDs. Additionally, to refine the relevance and focus of the dataset, we removed symbols such as question marks.

Since the original dataset does not contain actual queries, we used `spaCy` - a powerful language processing library for text processing to isolate critical nouns and named entities from titles, combining them with the original video titles to form queries. To incorporate the casing of the input, we created a lower case copy for each sample. In this way we augmented the dataset such that T5 can learn better query vs video id relations.

B. Model Training

For the training phase, we also used 1 NVIDIA T4 GPU for 8 epochs with learning rate e-3, opting for the T5-small model variant to align with our computing resource constraints. This selection was motivated by the need for a balance between model complexity and the operational realities of deployment in environments with limited computational capacity. This smaller model variant of T5, with its reduced parameter count of 60 million parameters, aligns with our goal of deploying a search function within power-restricted environments.

1) Training Details: The training phase involved multiple experiments across different sample sizes (50 to 6455 samples), employing a manual tuning approach for hyperparameters. The AdamW optimizer, with an initial learning rate of 0.001, and the default scheduler were utilized alongside cross-entropy loss. Validation and test sets were carefully split, and data tokenized using the T5 tokenizer.

Notably, training with a smaller subset of 50 samples achieved the most favorable loss reduction, with the loss dropping below 0.0001 after 1000 epochs. This experiment underscored the potential for fine-tuning the model on a restricted dataset while still achieving satisfactory results.

C. Observations and Metrics

The performance of the model was evaluated based on two primary metrics: the validation rate and precision. The validation rate assessed the model's ability to generate valid video IDs from full titles, while precision measured the relevancy of these IDs to their input titles.

As the training progressed across different dataset sizes, a trend emerged indicating a decrease in precision with increasing dataset size. This correlation suggests a challenge in maintaining model precision with larger volumes of data under a fixed training epoch regime.

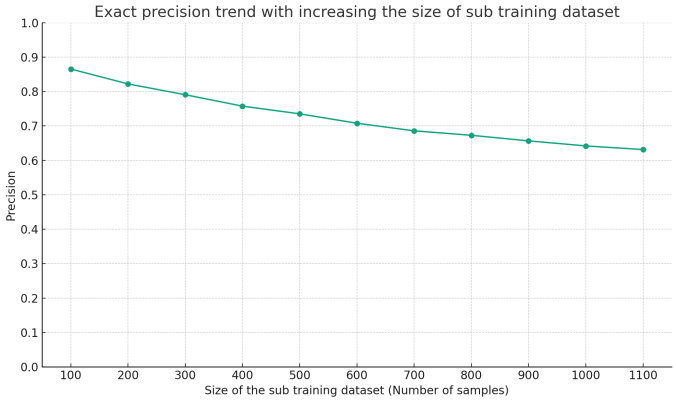

Fig. 1. Precision rates across different dataset sizes

This diminishing precision highlights the trend between dataset size and model accuracy under constrained computing conditions. Our findings align with the expectation that larger datasets pose greater challenges for model convergence within a fixed number of epochs.

Continuing our in-depth analysis of the T5 model's performance, the experiment strategically involved training across various dataset sizes, specifically: 100, 200, ..., 1000, 1100 samples. This comprehensive approach allowed for a precise examination of how dataset size influences model precision.

One of the main findings was a consistent observation across all tested sizes: the model's optimization converged adequately within the set 70 epochs for each dataset. This convergence is critical, as it validates the model's ability to adapt and learn across diverse scales of data.

However, a notable trend became apparent through this experimentation: as the dataset size decreased, there was a discernible drop in precision. This outcome is particularly illustrative of the inherent challenges in training models on smaller datasets. It's hypothesized that the larger datasets inherently offer a more complex and diversified learning environment, which, while more challenging, provides a richer basis for the model to understand and generalize across numerous instances. Conversely, with smaller datasets, the model faces a crunched learning scope, which could likely hinder its capacity to precisely memorize and generate accurate video IDs.

This precision drop can be associated with the observation that larger datasets make it harder for the model to converge to an optimal low training loss, given the same number of epochs for training. Essentially, training on a larger dataset within the same temporal bounds leads to a scenario where the model ends up with a higher training loss when compared to training on a smaller dataset. This factor underscores a crucial aspect of machine learning models: the trade-off between dataset size and the ability to thoroughly learn and adapt to the data.

Moreover, this result shows the difficulty for the T5 model to memorize more data points (in this case, video IDs). Training time, which positively correlates with the number of training samples, reflects as a significant consideration in this context. The more the data, the longer it takes to train, which while expected, brings to light scalability concerns, especially in environments constrained by computing resources.

While the experiment illustrates that the T5-small variant faces challenges with larger datasets in terms of memorization capacity, it also brings forth an intriguing question. Why does the T5-small's ability to memorize video IDs decline with an increase in IDs, despite the apparent trend of precision drop with fewer data points? This observation may not be directly explainable by the aforementioned trends and suggests an area for further investigation. It could imply a nuanced complexity in how sequence-to-sequence models like T5 deal with information density and the memorization-retrieval balance, especially when scaled down to smaller variants like the T5-small.

D. Conclusion on T5 Experiment

In summary, the T5 model demonstrates a promising capacity to memorize and generate video IDs from title inputs, though with limitations influenced by dataset size and computing constraints. This experiment not only showcased the potential of utilizing language models like T5 in search engine applications but also highlighted the critical balance required between computing resources and model performance in distributed environments.

IV. Conclusion

The conclusion goes here.

Appendix A
Experiment - Video ID predictor

A. Dataset

B. Training parameters

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank ChatGPT. ChatGPT is only used for polishing sentences and not for generating the content of this paper.

Field Name	Description
video_id	Unique identifier for each video. Useful for indexing and referencing specific videos in the dataset.
trending_date	The date when the video was trending. This can help in analyzing trends over time.
title	The title of the video. This is a crucial text field for IR, as it often contains keywords and topics that are highly relevant to the content of the video.
channel_title	The name of the channel that posted the video. This can be used for channel-based recommendations or analysis.
category_id	The category of the video (e.g., Entertainment, News, etc.). Useful for categorizing content and making category-based recommendations.
publish_time	When the video was published. This can be used to study the impact of publication time on trending status or viewership.
tags	Keywords associated with the video. Tags are extremely valuable for IR as they directly represent the content and context of the video.
views, likes, dislikes, comment_count	Engagement metrics. These can be used to gauge the popularity and reception of a video.
thumbnail_link	Link to the video's thumbnail. While not directly useful for IR, it can be used for visual analyses or to enhance the presentation of search results.
comments_disabled, ratings_disabled, video_error_or_removed	Boolean fields indicating certain statuses of the video. These can be used for filtering out certain videos from the analysis.
description	The description text of the video. Like the title, this is a rich text field that can be mined for keywords and topics.

TABLE I
FIELDS IN THE 'TRENDING YOUTUBE VIDEO STATISTICS' DATASET

Parameter	Value
global_step	32760
training_loss	3.232750225882245
train_runtime	3834.9877
train_samples_per_second	68.337
train_steps_per_second	8.542
total_flos	7106770821765600.0
train_loss	3.232750225882245
epoch	8.0

TABLE II
TRAINING PARAMETERS AND OUTPUTS

REFERENCES

[1] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, pp. 333–389, Jan. 2009.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] R. Ahmed and R. Boutaba, "A survey of distributed search techniques in large scale distributed systems," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 150–167, 2010.

[4] S. Zhuang, H. Ren, L. Shou, J. Pei, M. Gong, G. Zuccon, and D. Jiang, "Bridging the gap between indexing and retrieval for differentiable search index with query generation," 2023.

[5] Q. Tang, J. Chen, B. Yu, Y. Lu, C. Fu, H. Yu, H. Lin, F. Huang, B. He, X. Han, L. Sun, and Y. Li, "Self-retrieval: Building an information retrieval system with one large language model," 2024.

[6] P. Dean and B. Sundgren, "Quality aspects of a modern database service," in *Proceedings of 8th International Conference on Scientific and Statistical Data Base Management*. IEEE, 1996, pp. 156–161.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[9] M. J, "Trending youtube video statistics," https://www.kaggle.com/datasets/datasnaek/youtube-new/data, 2018, [Online; accessed 10-December-2023].