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.**

I. Introduction

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. Another research is

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. [5] Through experiments with state-of-the-art language models like BERT [6] and T5 [7], 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.

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) [5], 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.

II. Experiment - BERT

We piloted an experimental study to examine the application of a Large Language Model (LLM) as a proxy for a database, specifically targeting key-value retrieval systems. This type of data extraction paradigm is analogous to that employed by information retrieval mechanisms, akin to search engines and recommendation systems. An exemplar scenario for this exploration is the search functionality for YouTube videos, guided by the association contained within video metadata. The task is described as: Given a sequence of text, the model should output multiple URLs of YouTube videos whose information noted in their metadata (we use titles) are semantically relevant. The output should be ordered according to their relevance.

A. Dataset and Metrics

The dataset chosen for this investigation is a subset of the 'Trending YouTube Video Statistics' dataset [8] 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, we use the Recall metric to evaluate the model's precision in returning the correct video ID when the exact title of the video is provided as input.

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

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. Given this metric The overfitting of the model will be made intentional - encouraging the model to predict exact titles from the training set, as the goal is to achieve high precision in video title prediction. This method essentially trains the model to memorize the dataset, optimizing exact match retrieval.

C. Model Development and Results

BERT [6] (Bidirectional Encoder Representations from Transformers) is chosen as the basis model because it represents a cutting-edge advancement in language understanding tasks. As a pre-trained model, BERT captures rich linguistic context by processing words in relation to all other words in a sentence, which is particularly useful for text-based information retrieval – making it an ideal candidate for our use case. 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.

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 a NVIDIA T4 GPU for 8 epochs. The training parameters can be found in the appendix II. **??**Figure 1 is a line plot illustrating the model's learning progress throughout the training phase. The x-axis denotes the 'Steps' during training, and the y-axis records the 'Training loss'. We observe a downward trend in the training loss. The continuous decrease in the loss suggests that the model is becoming better at accurately predicting the titles of the videos and therefore, producing the corresponding video IDs with increasing precision. Eventually, the curve starts to flatten, indicating that the model is approaching a state of convergence.

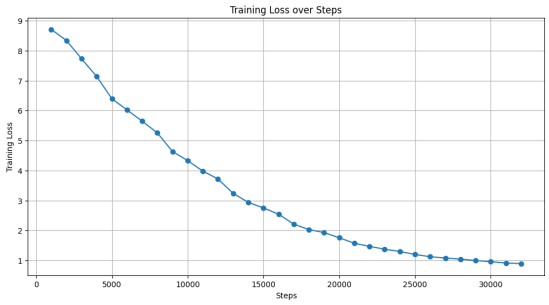

Fig. 1. Training Loss Over Steps

D. Result and Discussion

The resultant Recall for the model is 96.19%, it indicates that in 96.19% of the instances when the exact title of a video is fed to the model, it correctly outputs the top predicted title as being the same as the actual title within the training dataset. The direct correlation between the low final value of the training loss and the high Recall percentage points to an efficient learning process and a well-finetuned model that is capable of highly precise title predictions. This might indicate the potential applicability as a reliable tool in key-value retrieval systems, such as the search functionality for YouTube videos described in this study.

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.

Extending the vocabulary requires retraining the model.

III. Experiment with T5

During the experiment with BERT, video titles are encoded to an index which is used to represent the most probable vector representation in the vector space. The model is used as a text classifier. To utilize it as a database, the hard coded mapping from the predicted index to the video ids must be stored in advance. The video id prediction can also be seen as a text generation Downstream task. A new approach is to fine-tune models with encoder-decoder structure i.e. sequence-to-sequence models to explore the possibility to memorize or compress the content into shorten text which are the video ids. This means the video ids should be directly generated from the model. The "Text-to-Text Transfer Transformer" (T5) [7], a leading-edge Seq-to-Seq model from Google, was selected for this purpose. The baseline model was trained using either the full or truncated version of the C4 dataset. T5's architecture, designed for a variety of text-based tasks. especially text generation, makes it a prime candidate for our use case. In subsequent research [4], this model was also employed to learn from document-index pairs and generate document IDs based on the input partial text from the documents, demonstrating satisfactory performance. Since pre-trained T5 model contains the learned semantic information from C4 dataset, by applying this model for the video id generation task, the ability of the model of learning of natural language is transferred to understand the input title information of the video. Our exploration with T5 focuses on checking the adaptability and efficiency of using sequence-to-sequence models for search engine databases in environments constrained by computing power.

A. Data Preparation and Preprocessing

The initial step involved preparing and pre-processing the dataset for fine-tuning T5. Given the focus on video titles and their corresponding IDs, we utilized a dataset comprising YouTube video titles. This dataset required deduplication to ensure unique video title to ID mappings, resulting in a final count of 6455 unique samples. Furthermore, to refine the text data, we removed special symbols and utilized `spaCy`, an advanced natural language processing tool, to extract meaningful components from the titles. This process aimed to enhance the dataset by focusing on significant nouns and named entities.

1) Preprocessing Challenges: A notable challenge in preprocessing was handling the textual case sensitivity. Our objective was to maintain a model responsive to both cased and uncased inputs without compromising output accuracy. To accomplish this, we included both original and lower-cased variants of extracted title elements in the training data.

B. Model Training

Model training was conducted on a Google Cloud Platform Colab instance equipped with a T4 GPU. We used the T5-small variant, considering our constraint of limited computing resources. This smaller model variant, 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

1) Model Performance: 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.

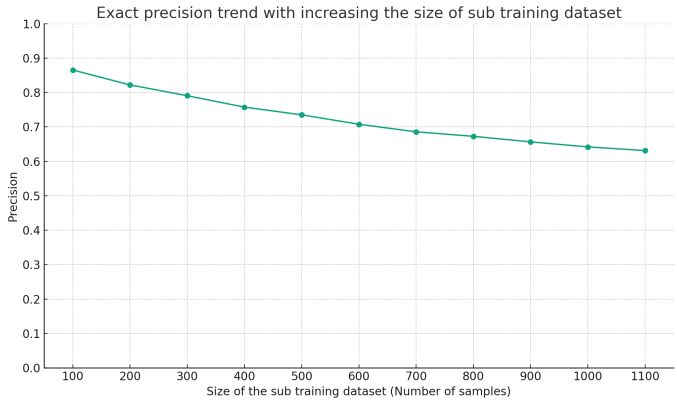

Fig. 2. Precision rates across different dataset sizes

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.

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

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

B. Training parameters

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

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.

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] 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.

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

[7] 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

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