

75

186.2

Re

rhenium

- An open-source search engine with Large Language Models in Julia programming language

Alexandra Elbakyan

CONTENTS

introduction

1

I

semantic
search

2

II

question
answering

7

III

domain
adaption

8

IV

evaluation

11

conclusion

15

references

R1

appendix

A1

introduction

In the modern world the amount of information available in digital form is vast, and is continuing to grow at exponential rate. For the humanity to make the most use of these abundant resources, it is important to have efficient technical tools that allow to search for relevant knowledge among all available data. Furthermore, with the ongoing proliferation of misinformation and fake news, the task of searching for high-quality information becomes of especial importance.

Research publications, specifically articles in academic journals, strive to adhere to higher standards of quality than social media or news outlets. These publications can serve as an important source of high-quality information on a wide range of questions.

Modern search engines for scientific knowledge, such as Scopus, Google Scholar or Semantic Scholar, are limited in functionality. One of the reasons is that because these projects are not open source, and therefore potential number of people who can contribute and develop them further is limited.

The aim of the present project is to provide an open source search engine for scientific articles: [Rhenium Search](#). The engine employs artificial intelligence technologies to select relevant documents based on the semantics or meaning of the query. Compared to existing search solutions, Rhenium provides extended functionality to the user, such as generation of automatic answers to user questions and summarization of search results. Furthermore, the search engine has an option to choose a language model that is specifically tailored to a selected area of science. The current version of the project includes support for two language models that are specifically trained for artificial intelligence field: [AI-RoBERTa](#) and [AI-BERT-GPL](#).

I semantic search

algorithms

A search engine works by calculating numerical representations, or **vector embeddings** for all documents in the index. The basic principle that makes search possible is that in the process of indexing similar documents, texts, images are converted into similar vectors. To find document relevant to the query, the vector embedding of the search query is calculated. Document vectors that are similar to the query vector are then selected from database as relevant documents.

Frequency-counting algorithms of the **TF-IDF** [1] or **BM25** [2] group produce sparse vector embeddings. They perform well on keyword search, but are unable capture meaning and context. Therefore for semantic search *dense* vectors are used that are calculated by neural networks. These neural networks are trained on a large corpora of texts and called *language models*.

```
[ -0.19000672  0.7446377 -0.50123394  0.31586924 -0.17452241 -0.87544173 -0.05090115
 0.17988959 -0.7951297  0.6452905  0.52770334 -0.11475293  0.2936556 -0.55720377
 0.50284606 -1.4852976 -0.46678835 -0.6907636  0.05684629 -0.1544157  0.04657895
 0.38582787  0.09503804 -0.3630916  0.51800793  1.1453001 -0.07831528  0.9630296
 0.8875228  0.49300483  0.6514863 -0.07611586  0.14710133 -0.07272324 -1.5737783
 0.28218713 -0.33000875 -0.04057535 -0.1474144 -0.44643947  0.47901243 -0.23038061
 -0.8645213 -0.41617462 -1.1408331  1.0487238 -1.1993206 -0.2264568  0.05117827
 -0.08423222 -0.6746544  0.6174031  0.48797026  0.10346548  0.01303269 -0.4111329
 -0.21663785 -0.7244148 -0.7039266 -0.11458502 -0.39704433 -0.48803172 -0.48467055
 -0.9452739 ... -0.08131462  0.29376665  0.44105974  0.4622919  0.21808475 -0.6443
 -0.10126625 -0.7224297 -0.1341627  0.46322966  0.8139906 -0.03460625 -0.18408488
 -0.4845081  0.95273304 -0.08325432 -0.13500445 -0.5815593 -0.4265771  0.4148934
 -0.10045864 -0.01861407  1.1102701 -0.1536411 -0.0486576 -0.11160882  0.6519504
 0.74137086 -0.9022149 -1.6861972 -0.09042241 -0.7172956  0.05626594  0.9896815
 0.5247946 -0.35886163 -0.7053058 -1.3276615  0.5126579  0.17043576  0.09689777
 1.040538 -0.6382619 -0.5144067 -0.09148376 -0.5601823 -1.6649672 -0.13956368
 -0.01062021 -0.11251686  0.15333211 -1.1276929 -0.52646756 -0.39758128  0.17773092
 -0.7621212  0.32408094  0.36512044  0.43411243  0.17545688  0.33467862  0.3062374
 0.33931798 -0.50420356]
```

Figure 1. An example 1024-dimension embedding vector (truncated) generated for a phrase «A quick brown fox jumped over the lazy dog» using gte-large language model

word2vec

This algorithm was pioneered by Google and became foundational for the field [3]. A simple feed-forward neural network is trained to predict a word based on other words surrounding it, or vice versa. The output layer of a neural network is used as a vector. This approach is able to capture basic semantic of a word, because words with similar meanings tend to occur in similar contexts. The authors have trained the model on a Google News dataset and the resulting embeddings for 1 million most frequent words are available online.

BERT

Word2vec embeddings are unable to differentiate meanings of word used in different contexts, the same word will always have the same vector embedding over all sentences. That is not the case with BERT, a language model introduced in 2019 by Google. BERT uses a much more complicated neural network based on transformer architecture and can be fine-tuned on a specific domain or task. The fine-tuning is performed as an additional training on a domain-specific data [4]. The BERT model became revolutionary and produced many types and variants. For example, DistilBERT reduced the original model by 40%, becoming 60% faster with only a small reduction in language understanding [5]. RoBERTa improved BERT by pre-training on larger corpus and hyperparameters tuning [6].

Sentence-BERT

BERT itself is not useful for the information retrieval task, because the model outputs embeddings for individual words, not for the whole sentence or text. Averaging individual word embeddings does not yield good results on a semantic similarity task. To overcome this problem, siamese network architecture for BERT was proposed [7]. The siamese network is composed of two BERT models with tied weights and is trained on a natural language inference datasets (Table 1). These datasets contain sentence pairs that are labeled for entailment, contradiction, or neutrality. The approach yielded good results, and today Sentence Transformer architecture is commonly used for similarity search task.

[8] SNLI	Stanford Natural Language Inference	570k
[9] MNLI	Multi-Genre Natural Language Inference	433k
[10] SCINLI	A Corpus for Natural Language Inference on Scientific Text	101k
[11] SCITAIL	Dataset created from multiple-choice science exams and web sentences	27k
[12] MEDNLI	A Natural Language Inference Dataset For The Clinical Domain	14k
[13] STSB	Text from image captions, news headlines and forums	8.5k

Table 1. Natural Language Inference datasets that are used for training Sentence Transformer models

searching for similar vectors

similarity score

To estimate numerically how similar two vectors are to each other, different measures can be used: Euclidean Distance, Manhattan Distance, Cosine Similarity and others (Figure 2). Many studies have shown **Cosine Similarity** to be the most accurate metric [14, 15] and it is the default metric used today for information retrieval.

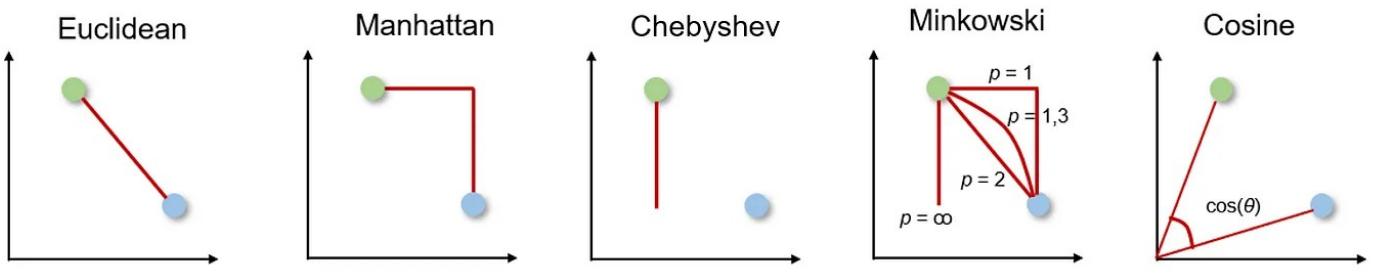


Figure 2. Different approaches to measure distance between two given points in a coordinate space [16]

Cosine similarity between two vectors is calculated as their dot product divided by size of the vectors (Formula 1). Thus cosine metric does not depend on input size, making possible to locate long-length documents by short search queries. Furthermore, cosine metric can be calculated efficiently with time complexity of $O(1)$ [17].

$$\text{similarity}(A, B) = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| |\mathbf{B}|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Formula 1. Cosine similarity between two vectors A and B [18]
where n is the dimensionality of vectors
 A_i and B_i are components of vector at index i

Hierarchical Navigable Small World

Searching for similar vectors in database of millions of documents by pairwise comparison can take very long time even when complexity of algorithm is $O(1)$ for every comparison. To solve this problem, indexing algorithms are used: the data is structured in a special way to allow for fast lookup. The type of structure depends on the data, e.g. for numerical data, B-tree index is used. **Hierarchical Navigable Small World** or HNSW [19] is a standard approach for building vector indexes and is implemented by default in modern database engines. The data is organized as a multi-layered graph and search is performed by navigating to nodes in the graph that are closest to the query, gradually descending from the top layer comprised of only a few vertices to the lowest one where all vertices are found. The time complexity of the algorithm on search is $O(\log N)$ and construction of the graph takes longer with complexity of $O(N * \log N)$.

> implementation

To create an index of scientific documents, I have used openly available datasets provided by arXiv [20], PubMed [21] and ACL [22]. There are 2.5 million, 40 million and 50,000 scientific articles in each dataset accordingly. Processing all of the PubMed collection available from Hugging Face turned out to take substantial amount of time, so I took only a random part of it (8.5 million articles) for alpha version of the project. The part was selected from those PubMed documents that have non-empty abstract.

For every article, its title and abstract were concatenated and vector embedding calculated for the resulting text. I have implemented two embedding algorithms: one based on word2vec and another based on Sentence Transformer. The word2vec was selected because it is fast and requires very little computational resource, which makes it especially suitable while testing other functions of the engine e.g. GUI features and general functionality. To calculate document embedding, vectors of individual words from word2vec were averaged; if the word was not present in the dictionary, its vector was considered as null. Surprisingly, even this simple approach turned out to give relevant results when tested on random queries:

The screenshot shows a search interface with a pink header bar. On the left, there's a sidebar with a '75' icon and '186.2'. Below the sidebar, the word 'Re' is partially visible. The main search area has the word 'search' in large black letters. To the right of 'search', the query 'applications of artificial intelligence in drug design' is shown. Below the query, there are two buttons: 'word2vec' and 'gte-large-en-v1.5'. The search results section starts with the following text:
results for: applications of artificial intelligence in drug design
embedding model: word2vec
retrieval took 0.0694 sec.

An In-depth Summary of Recent Artificial Intelligence Applications in Drug Design (2021) 0.737
As a promising tool to navigate in the vast chemical space, artificial intelligence (AI) is leveraged for drug design. From the year 2017 to 2021, the number of applications of several recent AI models (i.e. graph neural network (GNN), recurrent neural network (RNN), variation autoencoder (VAE), generative adversarial network (GAN), flow and reinforcement learning (RL)) in drug design increases significantly. Many relevant...

A Quantum Computing Pipeline for Real World Drug Discovery: From Algorithm to Quantum Hardware (2024) 0.7351
Quantum computing, with its superior computational capabilities compared to classical approaches, holds the potential to revolutionize numerous scientific domains, including pharmaceuticals. However, the application of quantum computing for drug discovery has primarily been limited to proof-of-concept studies, which often fail to capture the intricacies of real-world drug development challenges. In this study, we diverge from conventional investigations by developing an advanced quantum...

NeuroCADR: Drug Repurposing to Reveal Novel Anti-Epileptic Drug Candidates Through an Integrated Computational Approach (2023) 0.7312
Drug repurposing is an emerging approach for drug discovery involving the reassignment of existing drugs for novel purposes. An alternative to the traditional de novo process of drug development, repurposed drugs are faster, cheaper, and less failure prone than drugs developed from traditional methods. Recently, drug repurposing has been performed in silico, in which databases of drugs and chemical information are used...

Figure 3. Search results based on word2vec. On the screenshot the user interface of the early version of search engine is shown

I use two databases to store document vector index: classical **postgresql** and **Redis** in-memory storage. Both of them have capability to perform vector similarity search, but on large datasets Redis queries were running 10x times faster, so it was selected. At the same time, keeping the whole database including title, abstract and other information in Redis is not possible, since the RAM requirement is high (the word2vec index only took 150 GB RAM) so document metadata is stored in postgresql.

For the second approach, I consulted model ranking table available on Hugging Face to select the best Sentence Transformer model available for information retrieval [23]. The **gte-large-en-v1.5** was ranked among the top 3 scorers, and the first one for SciFact dataset (Figure 4). Yet another advantage of this model is that it is small and does not require a lot of computational power and RAM.

Rank	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Average	ArguAna
2	<u>gte-large-en-v1.5</u>	434	1.62	57.91	72.11
5	<u>voyage-lite-02-instruct</u>	1220	4.54	56.6	70.28
3	<u>GritLM-7B</u>	7242	26.98	57.41	63.24
7	<u>LLM2Vec-Mistral-supervised</u>	7111	26.49	55.99	57.48
12	<u>text-embedding-3-large</u>			55.44	58.05
1	<u>SFR-Embedding-Mistral</u>	7111	26.49	59	67.17
18	<u>LLM2Vec-Llama-supervised</u>	6607	24.61	54.6	56.53
13	<u>GritLM-8x7B</u>	46703	173.98	55.09	59.49
21	<u>gte-base-en-v1.5</u>	137	0.51	54.09	63.49
4	<u>e5-mistral-7b-instruct</u>	7111	26.49	56.89	61.88
42	<u>gte-base</u>	109	0.41	51.14	57.12

Table 2. Top-ranked language models on Hugging Face according to average score across all benchmarks for information retrieval task

Still, calculating 1024-dimensional embeddings of gte-large for the whole dataset took substantially more time and resources than 300-dimensional word2vec; processing 10,000,000 papers with occasional pauses took a few days on a single H100 GPU. Furthermore, uploading these vectors into Redis provided additional challenge and required installing a special **RIOT** tool. The RAM usage to index this model went up to 400 GB.

The implementation was done in **Julia Genie Framework**, although I had to use Python library calls to use Hugging Face, because its implementation in Julia is not stable yet. Fortunately, Julia provides a dedicated PyCall module for such cases.



II

question answering

Even though BERT language model that was explored in previous section can be used for question answering, another architectures – Generative Language models such as GPT – perform better on this task. These models are trained on a massive amounts of data and were shown to store a substantial amount of factual knowledge in their parameters, which enables them to generate correct answers to questions provided as an input. However, knowledge that is stored in neural network weights is limited and cannot be easily updated with novel information. To overcome this problem, a combined **Retrieval Augmented Generation** (RAG) methodology was proposed by Facebook [24]. First, information that is relevant to the user request is retrieved by search algorithm. In the second step this information is used as an input to the generative language model along with a question.

Using RAG approach, a search engine could provide a useful functionality by creating a quick summary of the top results. That can substantially save time for the user, who would need to manually browse and analyze documents retrieved by search engine otherwise.

I have implemented RAG by adding two generative models to the engine. The first one is an innovative **phi-3-mini-4k-instruct** model from Microsoft: it is small and has low resource requirements. Another one is **Llama3-ChatQA-1.5-8B** model because Llama architecture is one of the most commonly used among language models. Both models achieved high scores on benchmarks according to the Hugging Face ranking table [25].

results for: **should humanity start fighting climate change, or focus on a more pressing issues instead?**
embedding model: gtelarge
retrieval took 0.2321 sec.

answer by **phi-3-mini**: Humanity should not choose between fighting climate change and focusing on more pressing issues; rather, it should address all these concerns simultaneously. Climate change is a pressing issue that affects all aspects of life on Earth, including food security, water availability, and health. By integrating climate change mitigation and adaptation strategies into the broader context of global development, humanity can work towards a more sustainable and resilient future. Efforts to combat climate change can also help address other pressing issues. For example, transitioning to renewable energy sources can reduce air pollution, which in turn can improve public health. Investing in green infrastructure can create jobs and stimulate economic growth. Additionally, implementing sustainable agricultural practices can help ensure food security and protect biodiversity. In conclusion, humanity should prioritize fighting climate change while simultaneously addressing other pressing issues. By doing so, we can create a more sustainable and equitable world for future generations.

1. Gary. J. McFarlane
0.6709 Climate change--the greatest public health threat of our time: seeing the wood, not just the trees. (2010)

If asked to describe the key public health challenges of our time many practitioners might well cite issues such as health inequalities, obesity, smoking and poverty. However, with the greatest of respect to those agendas, they are not, in my view, the greatest priority at present. If we cannot learn to live within sustainable limits and damage beyond repair the essential life support systems...

Figure 4. Example answer generated by **phi-3-mini-4k-instruct** model

III

domain adaptation

overview



It is a well-known fact that an expert in a field will give a more detailed and correct answer to a question compared to a person who has only general knowledge. That makes reasonable to assume that specialized artificial neural networks – that were trained on a dataset from a special area of knowledge – will have a better performance in that domain, than neural networks or models that were trained on a general corpus which includes texts from various venues.

Furthermore, in the context of search engines it is also important that semantic meaning of a word can change from one area to another. For example, the word «virus» will have completely different meaning for biologist and for computer scientist.

The option to retrieve different search results according to a selected field could be useful in such cases.

An empirical example of the changing semantic of word comes from word2vec where by default embeddings trained on Google News dataset are used. I have run a k-means clustering on a vocabulary of 72,396 words taken from papers about artificial intelligence that have embedding vector specified in the dictionary (Appendix A5). Then I viewed the cluster to which the word «intelligence» belongs to. That produced unexpected results: intelligence was grouped with such words as terrorism, attacks, Iraq, Taliban, bombing, extremist and etc. Given that the model was trained on a news dataset, that is not surprising. I then trained word2vec from scratch using only the texts from arXiv AI papers and run the clustering again. The semantic of intelligence changed and it was now grouped with such words as understanding, brains, wisdom, faculties, agency and etc.

Another example comes from the study [26] where word2vec model trained from scratch on 3.3 million article abstracts from materials science field was able to capture latent knowledge. In 2019 Allen Institute for Artificial Intelligence released a BERT model that was pre-trained on 1 million full-text scientific papers: **SciBERT**. The training dataset consisted of 80% of articles from biomedical and 20% from computer science domains. The model outperformed general-domain BERT both with and without fine-tuning on a various tasks such as Named Entity Recognition or text classification [27]. **BioBERT** released the same year was not trained entirely from scratch, but used base BERT to set up initial weights. The training over a few million articles from Pubmed took 23 days on 8 V100 GPU. The resulting model achieved comparable but somewhat worse performance than SciBERT [28]. It was noticed that training from scratch yields better results at least for the biomedical domain, and smaller completely specialized models with 800m and 345m parameters outperformed 1.2b parameter **BioMegatron** model that was initialized on general domain [29].

Most work in domain adaptation of language models was done on BERT and in biomedical domain. Recently, with the emergence of generative AI, specialized scientific models such as **GeoLLaMA** [30], **OceanGPT** [31] or **AstroLLaMA** [32] started to appear that are based on the open-source generative LLaMA model. The model is fine-tuned to the domain in two steps: **unsupervised pre-training** and **supervised instruction-tuning** stage

GeoLLaMA dataset used for pre-training included full texts extracted from 1.1 million open access papers in the field of geoscience, as well as Wikipedia articles. The training of LLaMA-7B took 214 hours on 4 A100 GPU. OceanGPT used a corpus of 67,633 full-text articles from open-access literature and the training took 7 days on 6 A800 GPU. AstroLLaMA employed 300,000 abstracts and the training was done one-stage without instruction fine-tuning. The authors mention that:

«AstroLLaMA exhibits remarkable context-awareness in its completions by showing a deep understanding of astronomical concepts»

To train Large Language model efficiently and in a reasonable amount of time, Parameter-Efficient Fine-Tuning techniques such as LoRA are applied to limit the number of parameters that will be changed during training [33].

Generative Pseudo Labeling

This method of unsupervised domain adaptation was developed specifically for information retrieval task [34]. A collection of passages from target domain is taken as an input, and queries are generated for every passage using generative language model. For every query, the least similar passages are retrieved from input dataset using either BM25 or Sentence Transformer. After that a **cross-encoder** model is applied to every query-passage pair to calculate their similarity score; a cross-encoder is a BERT model that is trained to take two sentences as input and output their similarity score between 0 and 1. The resulting dataset of queries, passages and their similarity scores is then used to perform supervised training of a sentence transformer model.

implementation

In this version of the project I implemented support for three specialized models: BioBERT for biology and two models for the field of Artificial Intelligence: AI-RoBERTa and AI-BERT-GPL.

BioBERT

This model was presented in 2019 and achieved significantly improved performance on biomedical tasks compared to general-domain BERT. However, it is not possible to directly use BERT models to calculate sentence embeddings – they must be turned into Sentence Transformers and fine-tuned on sentence similarity datasets first. Fortunately, there is no need to perform this task: a fine-tuned variant of BioBERT model trained on multiple NLI datasets including SciNLI is available on Hugging Face and can be used directly [35].

AI is an unusual domain for which specialized models do not exist yet. Therefore I had to train two models for the AI myself. To create a dataset of domain-specific texts, I used gte-large-en-1.5 model to retrieve 1 million papers that are relevant to the query «artificial intelligence». Titles and abstracts were then concatenated and the resulting dataset was used to perform unsupervised domain adaptation.

AI-RoBERTa

For this model I took DistilRoBERTa as a base model, preserving initial weights. I filtered the input dataset and removed texts that were either too short (less than 400 characters) or too long (> 1800 characters), remaining around 800k.

The training was performed in two stages:

1. The model was trained in an unsupervised mode over the corpus of AI texts using standard Masked Language Modeling according to method described in [36]
2. The model was further fine-tuned on SNLI and MNLI datasets to get a Sentence Transformer according to method described in [37]

The training process took 8-9 hours on 16 V100 GPU. The training was done in parallel on multiple GPU cards; this mode is only supported in the latest 3.0.0 version of Sentence Transformers. Since version 3.0.0 is available only in a pre-released version yet, it needs to be installed manually in order to test the model:

```
! pip install git+https://github.com/UKPLab/sentence-transformers@v3.0-pre-release
```

AI-BERT-GPL

Here I used GPL method to train base DistilBERT model on the domain data. The code to perform the full process of training was made available by authors on GitHub [38], but it has no support for multiple GPUs. The training was done on a single H100 GPU and took > 40 hours.

The final step required calculating vector embeddings for all 10m documents in the database for all three models and storing them in the index. The calculation itself took 20-30 min. for each model on 16 V100 GPU. To index these vectors, I had to expand the setup by adding additional server. This server is running **QDrant**, a database developed for vector search. The advantage of QDrant over Redis is that the former can store vector embeddings not in memory only, but also on disk. The access speed when using SSD is practically the same as RAM, but SSD can handle much larger volumes of data. QDrant provides benchmarks on the website showing it to be the fastest vector database available:

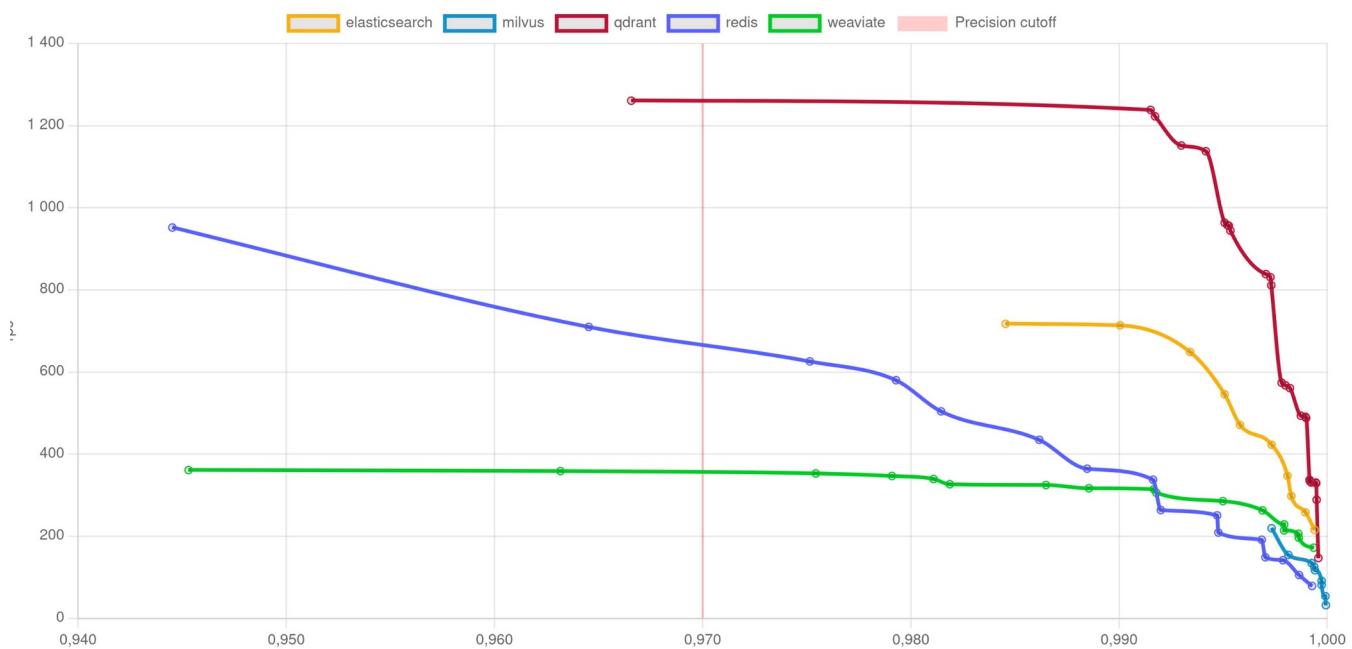


Figure 5. QDrant as compared to other vector storages in terms of rate of operations per second [39]

evaluation

Quantitative / in numbers

Quantitative evaluation of an embedding model is not straightforward, because no correct embedding can be given in advance to compare the results with. Still, even in that case there are methods that allow to estimate how well the model performs by calculating a score.

Triplet Cosine Distance Accuracy

The metric is computed over a dataset of sentences grouped as triplets. The first sentence in the triplet is the query, the second is positive example – sentence relevant to the query, and the last one is negative example – sentence that is not similar or not relevant to the query. If the embedding model performs well, the positive sample should be located in a vector space closer to the query than negative, i.e. distance to the first sentence should be smaller than to the second. If that is the case, the sample is counted as a correct one. The resulting metric is the percentage of the correct samples.

The algorithm is already implemented in sentence_transformers evaluation package. For dataset I will use **scidocs-reranking** dataset that consists of 4k labeled triplets relevant to the domain of science [40]. The result for the retrieval models used are provided in Table 3.

	gte-large	BioBERT	AI-RoBERTa	AI-BERT-GPL
score	0.94180943	0.85071644	0.86098910	0.79142114

Table 3. Triplet Cosine Distance Accuracy for retrieval models

The accuracy of all models is high except the last one. Specialized models performed similar according to this metric and worse than general-purpose model – but that can be expected since specialized models are supposed to perform better only in the area they were trained on.

The following metrics are used to evaluate the list of retrieved documents, given that boolean relevance score for every document to the query is known in advance.

Mean reciprocal rank, MRR

Reciprocal rank of a given query is calculated as an inverse of the position (rank) of the first relevant document in the list. The mean RR is calculated over all queries.

Formula 2. Mean Reciprocal Rank [41]

where Q is a set of queries tested; from retrieved documents, top k results are considered; r_q is the rank of the first relevant document in the list

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \begin{cases} 1/(1 + r_q), & r_q < k \\ 0, & r_q \geq k \end{cases}$$

Normalized discounted cumulative gain, NDCG

The measure is a sum of scores for every relevant document divided by its position in the result list: displaying relevant document later is penalized. The sum is normalized by its maximum possible value.

Formula 3.

Normalized Discounted Cumulative Gain [42]

from retrieved documents, top k results are considered; i is the position of the document in the rank and rel_i is the relevance score of the document at position i

$$NDCG@k = \frac{DCG@k}{IDCG@k}$$

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$IDCG@k = \sum_{i=1}^k \frac{2^{rel_{\sigma(i)}} - 1}{\log_2(i+1)}$$

Mean average precision, MAP

Average precision equals to the area under precision / recall curve and can be calculated as a normalized sum of precision scores (estimated as a number of relevant documents to the number of retrieved) for the number of results displayed from $n = 1..r$ where r is the rank of the last relevant document. Mean AP calculated over all queries.

Formula 4. Mean Average Precision [41]

where Q is a set of all queries tested, q is the query, z_{qi} is a set of documents relevant to q, i is the index of document in a set and r_{qi} is a rank of the document in the output

$$AP(q) = \sum_{i:z_{qi}=1} \frac{1+i}{1+r_{qi}}$$

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q)$$

The sentence_transformers evaluation package implements algorithm for calculating MRR, DCG and MAP scores in RerankingEvaluator. The dataset must be an array of triplets, but this time multiple positive and negative sentences must be provided for each query. I will be using scidocs dataset available on Hugging Face that provides 30 positive and negative samples for 1000 queries [43]. As can be seen from Table 3, AI-BERT-GPL scored higher than other special models on all metrics, despite getting low cosine distance accuracy scores. For other models, even though different metrics and datasets were used, results were similar to those displayed in previous table with gte-large model at the top of the list and special models are performing noticeably worse but not much different from each other.

	gte-large	BioBERT	AI-RoBERTa	AI-BERT-GPL
MRR@10	0.99283333	0.91068095	0.90570238	0.93495714
NDCG@10	0.95424540	0.77984771	0.78139278	0.80093784
MAP	0.92281592	0.71697288	0.71533471	0.74431640

Table 4. Reranking metrics scores for retrieval models

queries / second performance

Large Language Models are computationally hard, and typically require special hardware such as GPU to be installed. At the same time, an information search engine running online can receive large number of requests in a short time. It is therefore important to estimate the maximum number of requests that can be processed by different model in a fixed amount of time.

Using phi-3-mini-4k-instruct model, I generated a pool of random search queries (Appendix A6) Then I ran a script for every model, that would calculate embeddings for a random query selected from the pool in an infinite loop for 60 seconds, and record the total number of queries processed. The script would run from 1 to 8 processes in parallel to simulate the work of search engine in production mode. As a more complex model, gte-large was the slowest one, while special models based on reduced version of BERT performed two times faster than original BERT:

n of processes	gte-large	BioBERT	AI-RoBERTa	AI-BERT-GPL
1	4857	12485	16456	17123
2	8266	17947	27651	28721
3	8829	21247	32200	33284
4	9015	22072	35919	37423
5	8905	22577	38200	39450
6	8513	22601	39599	40988
7	8339	22183	40656	41260
8	8261	21986	40462	41776
average maximum speed	150 query/sec	377 query/sec	677 query/sec	696 query/sec

Table 5. The number of queries processed by every model in 60 sec. on a single H100 GPU

It can be concluded from the results above that the system is quite efficient and can be potentially be used under high loads, providing relevant results with the delay of less than 1 sec. and precision of 0.8 on average (0.9 for general and at least 0.7 for specialized models) which translates to the number of relevant results equal to 80% and higher.

Qualitative metrics

Even though numerical metrics are good to get a general understanding of the model performance, the nuanced evaluation can be done by human subject only. This sections provides results of qualitative evaluation that I performed myself. I have ran some queries and analyze top 4 results given by different models. The results are provided in the Appendix A7-A11.

For the «**ethics of AI**» query the most relevant results were given by AI-BERT-GPL and gte-large models. BioBERT outputed mostly papers on bioethics instead of AI, while output of AI-Roberta was relevant to AI ethics field, but papers provided were obscure. Despite BioBERT and AI-RoBERTa got the same score on test, this query shows how different result is given according to the model specialization.

More interestingly, for the query «**AI**» itself most relevant results were given by AI-BERT-GPL and AI-RoBERTa models. Although gte-large provided links to papers important in the AI field, these papers seem to be about different methods in AI – not about artificial intelligence in general. The output of BioBERT and word2vec was completely irrelevant.

The word «**embedding**» is a special term in the AI field, and here it has a different meaning than when used in other contexts – which makes it especially interesting to test how different models will interpret the word. Surprisingly, in this case word2vec gave results relevant to AI field. The same was true for gte-large and AI-RoBERTa models, while BioBERT completely ignored the AI meaning of the word, providing results related to biology medicine. Top results displayed by AI-BERT-GPL were in the field of mathematics.

The «**llama**» is another term with meaning specific to the modern AI field. On this term the output of AI-BERT-GPL gave perfectly relevant results, all coming from the AI field. The output of gte-large and BioBERT was based upon biological meaning of the word llama, however, BioBERT included one paper from AI in top 4 results. To much surprise, AI-RoBERTa model did not capture any of the meanings and the output was basically random for this term.

The results of asking a question «**how does chat GPT work?**» were surprising. The gte-large model responded with completely irrelevant results, interpreting GPT as glucose pump test. At the same time, both AI-BERT-GPL, AI-RoBERTa and even BioBERT responded with relevant articles, with answer of AI-RoBERTa being a little bit more relevant, because the first result included article about technology behind ChatGPT, not only its performance on various tasks. To make sure that query was processed by gte-large model and not by BioBERT due to some bug in the program, I reloaded the application with special models disabled in the config (i.e. they were not loaded at startup) but the response of gte-large remained the same. This testing case is an example where specialized models perform better than general one, even when the latter has higher accuracy scores when testing.

Overall, performance of special models yielded results that are much more relevant to the field they were trained upon in some cases, but that effect was not consistent. The models can be improved further by training on more data.

conclusion

In this project an alpha version of the semantic search engine for scientific articles: Rhenium Search – was implemented. The search is performed over the database of 10,953,706 open access academic publications, that includes articles from arXiv, PubMed and ACL repositories. The search engine is running on a web server accessible through the Internet and is ready to use. Results are provided to the user with low latency, with delays less than 1 sec. For queries provided in question form, a short answer is provided in a separate box.

The website provides an option to select the Large Language Model used for answer generation and for document retrieval. Two models: word2vec and gte-large are supported for general-domain search. The default model gte-large retrieves documents that are relevant to the query, with accuracy of more than 0.9 according to the test results.

There is an option to use fine-tuned language model for two areas of research: Biology and AI. The accuracy scores of these models are 0.8 and higher. Two models: AI-RoBERTa and AI-BERT-GPL are available for the domain of artificial intelligence. These models were trained on a dataset of 800k articles relevant to the AI topic.

The goals set in the project proposal were fully achieved. However, the result is only an alpha version of the search engine. To compete with well-established academic search engines such as Scopus or Google Scholar, the number of indexed documents must reach at least 100 million. For example, in current version only part of the PubMed collection was included because of time and hardware limitations. It must be noted that expanding the database will be easy due to modular structure of the project.

Future work on this project will include training additional language models to provide fine-tuned search for large number of research areas. There are many possible improvements that could potentially increase the performance of fine-tuned models. These include, but not limited to: training on a larger corpus; using SciBERT as a base model; training on a balanced dataset – most articles in the dataset now come from biomedical domain; and others. These possibilities for improvement must be carefully explored and tested to achieve best performance for fine-tuned models.

references

1. Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), 11-21.
2. Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *Nist Special Publication Sp*, 109, 109.
3. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
4. Kenton, J. D. M. W. C., & Toutanova, L. K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (Vol. 1, p. 2)
5. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
6. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
7. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
8. Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
9. Khanuja, S., Dandapat, S., Sitaram, S., & Choudhury, M. (2020). A new dataset for natural language inference from code-mixed conversations. *arXiv preprint arXiv:2004.05051*.
10. Sadat, M., & Caragea, C. (2022). Scinli: A corpus for natural language inference on scientific text. *arXiv preprint arXiv:2203.06728*.
11. Khot, T., Sabharwal, A., & Clark, P. (2018, April). Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
12. Romanov, A., & Shivade, C. (2018). Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*.
13. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
14. Ghosh, J., & Strehl, A. (2006). Similarity-based text clustering: A comparative study. In *Grouping Multidimensional Data: Recent Advances in Clustering* (pp. 73-97). Berlin, Heidelberg: Springer Berlin Heidelberg.
15. Hajeer, I. (2012). Comparison on the effectiveness of different statistical similarity measures. *International Journal of Computer Applications*, 53(8).

16. Dancker, J. (2022) A brief introduction to Distance Measures. Medium. <https://medium.com/@jodancker/a-brief-introduction-to-distance-measures-ac89cbd2298>
17. Karthikeyan, K. (2022) Time complexity for document similarity measures. Kaggle. <https://www.kaggle.com/code/kirankarthikeyan/time-complexity-for-document-similarity-measures>
18. Yunanda, G., Nurjanah, D., & Meliana, S. (2022). Recommendation system from microsoft news data using TF-IDF and cosine similarity methods. *Building of Informatics, Technology and Science (BITS)*, 4(1), 277-284.
19. Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4), 824-836.
20. arXiv.org submitters. (2024). arXiv Dataset [Dataset]. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/7548853>
21. Patry, N. (2023). ncbi/pubmed [Dataset]. Hugging Face. <https://huggingface.co/datasets/ncbi/pubmed>
22. Bird, S., Dale, R., Dorr, B. J., Gibson, B. R., Joseph, M. T., Kan, M. Y., ... & Tan, Y. F. (2008, May). The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In LREC.
23. Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2022). MTEB: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316.
24. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.
25. Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., Wolf, T. (2023) Open LLM Leaderboard. Hugging Face. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard
26. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., ... & Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763), 95-98.
27. Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676.
28. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234-1240.
29. Shin H. C. et al. BioMegatron: larger biomedical domain language model //Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). - 2020. - C. 4700-4706.
30. Deng, C., Zhang, T., He, Z., Chen, Q., Shi, Y., Zhou, L., ... & He, J. (2023). Learning a foundation language model for geoscience knowledge understanding and utilization. arXiv preprint arXiv:2306.05064.
31. Bi, Z., Zhang, N., Xue, Y., Ou, Y., Ji, D., Zheng, G., & Chen, H. (2023). Oceangpt: A large language model for ocean science tasks. arXiv preprint arXiv:2310.02031.

32. Nguyen, T. D., Ting, Y. S., Ciucă, I., O'Neill, C., Sun, Z. C., Jabłońska, M., ... & Schawinski, K. (2023). Astrollama: Towards specialized foundation models in astronomy. arXiv preprint arXiv:2309.06126.
33. Wang, K., Thakur, N., Reimers, N., & Gurevych, I. (2021). GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. arXiv preprint arXiv:2112.07577.
34. Xu, L., Xie, H., Qin, S. Z. J., Tao, X., & Wang, F. L. (2023). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. arXiv preprint arXiv:2312.12148.
35. Deka, P., Jurek-Loughrey, A., & P, D. (2022, October). Evidence extraction to validate medical claims in fake news detection. In International Conference on Health Information Science (pp. 3-15). Cham: Springer Nature Switzerland.
36. Hinepo (2024) Domain Adaptation with - MLM. Kaggle. <https://www.kaggle.com/code/hinepo/domain-adaptation-with-mlm>
37. Aarsen, T. (2024, March) training_nli.py UKPLab/sentence-transformers. GitHub. https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/nli/training_nli.py
38. Ubiquitous Knowledge Lab. gpl. GitHub <https://github.com/UKPLab/gpl/tree/main>
39. Vector Database Benchmarks (2023, February). QDrant. <https://qdrant.tech/benchmarks/>
40. mteb/scidocs (2024) Hugging Face. <https://huggingface.co/datasets/mteb/scidocs>
41. Chakrabarti, S., Khanna, R., Sawant, U., & Bhattacharyya, C. (2008, August). Structured learning for non-smooth ranking losses. In Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 88-96).
42. Currie, D. (2023, November) Statistical Stories - NDCG (Normalized Discounted Cumulative Gain). Medium. <https://medium.com/@Currie32/statistical-stories-ndcg-normalized-discounted-cumulative-gain-0be22d61a87d>
43. mteb/scidocs-reranking (2022) Hugging Face. <https://huggingface.co/datasets/mteb/scidocs-reranking>

APPENDIX

& design features

The interface of the search engine is shown on Figure 6. There is a settings menu on the left, where a user can select language model for retrieval and for answering questions. The user has to enter the query in search box and either click the button on the right, or press Enter to display the results. At first, top 32 most relevant results will be displayed (Appendix A2). As the user scrolls down the page, more and more search results will be loaded dynamically. In addition to infinite scroll, the web service also supports pagination to make navigation easier.

For every search result, its rank and similarity score will be displayed. By clicking on the article link a separate page opens, that provides more detailed information about the article, including full abstract and link to the website (Appendix A3). Furthermore, a list of similar articles will be shown, with percentage of similarity for each article specified.

The search engine has some additional features implemented: it supports filtering the results by year and source of publication. However, filtering sometimes can make responses slower.

question answering

For those user requests that are formulated as questions, the search engine will automatically generate an answer, using information from most relevant documents as a context. The search engine supports two generative models: LLaMA3-ChatQA and phi-3-mini. The first model gives short and the second more elaborate answers. The answers are generated in real-time and do not require waiting from the user.



Figure 6. Main page of the search engine

APPENDIX

design and features



settings

embedding model used for retrieval:

general

gte-large-en-v1.5

word2vec

special

ai-roberta-800

BioBERT

language model to answer questions:

LLaMA3-ChatQA-8B

phi-3-mini

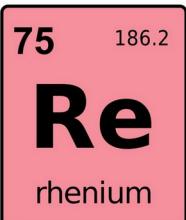
published in

1963 - 2024

databases

acl arxiv
 pubmed

RAG enabled



Re search

how does chat GPT work?



results for: **how does chat GPT work?**

embedding model: ai-roberta
retrieval took 0.3275 sec.

answer by **phi-3-mini**: ChatGPT works by using a type of artificial intelligence model known as a Transformer, specifically a variant called the Generative Pre-trained Transformer (GPT). Here's a simplified explanation of how it works:

1. Pre-training: ChatGPT is initially trained on a large dataset of text from the internet. This dataset includes a wide variety of topics, genres, and writing styles. During this phase, the model learns to predict the next word in a sentence by analyzing the context provided by the words that come before it. This process is called unsupervised learning because it doesn't require labeled data.

1. Sakib Shahriar and Kadhim Hayawi

0.8735 **Let's have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations** (2023)

The emergence of an AI-powered chatbot that can generate human-like sentences and write coherent essays has caught the world's attention. This paper discusses the historical overview of chatbots and the technology behind Chat Generative Pre-trained Transformer, better known as ChatGPT. Moreover, potential applications of ChatGPT in various domains, including healthcare, education, and research, are highlighted. Despite promising results, there are several privacy...

2. C. Or{\u{a}}san

0.8606 **ChatGPT for translators: a survey** (2023)

This article surveys the most important ways in which translators can use ChatGPT. The focus is on scenarios where ChatGPT supports the work of translators, rather than tries to replace them. A discussion of issues that translators need to consider when using large language models, and ChatGPT in particular, is also provided.

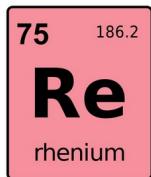
3. Ryan Hardesty Lewis, Junfeng Jiao

0.855 **ComputeGPT: A computational chat model for numerical problems** (2023)

Language models are not accurate in numerical problems. Their architecture does not allow for anything less than a probabilistic next word. This paper introduces ComputeGPT: an approach of creating a chat model able to answer computational problems through running on-demand code. ComputeGPT converts each question to relevant code, runs the code, and returns the computed answer as part of the chat. We...

APPENDIX

design and features



search

2023

Let's have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations

Sakib Shahriar and Kadhim Hayawi

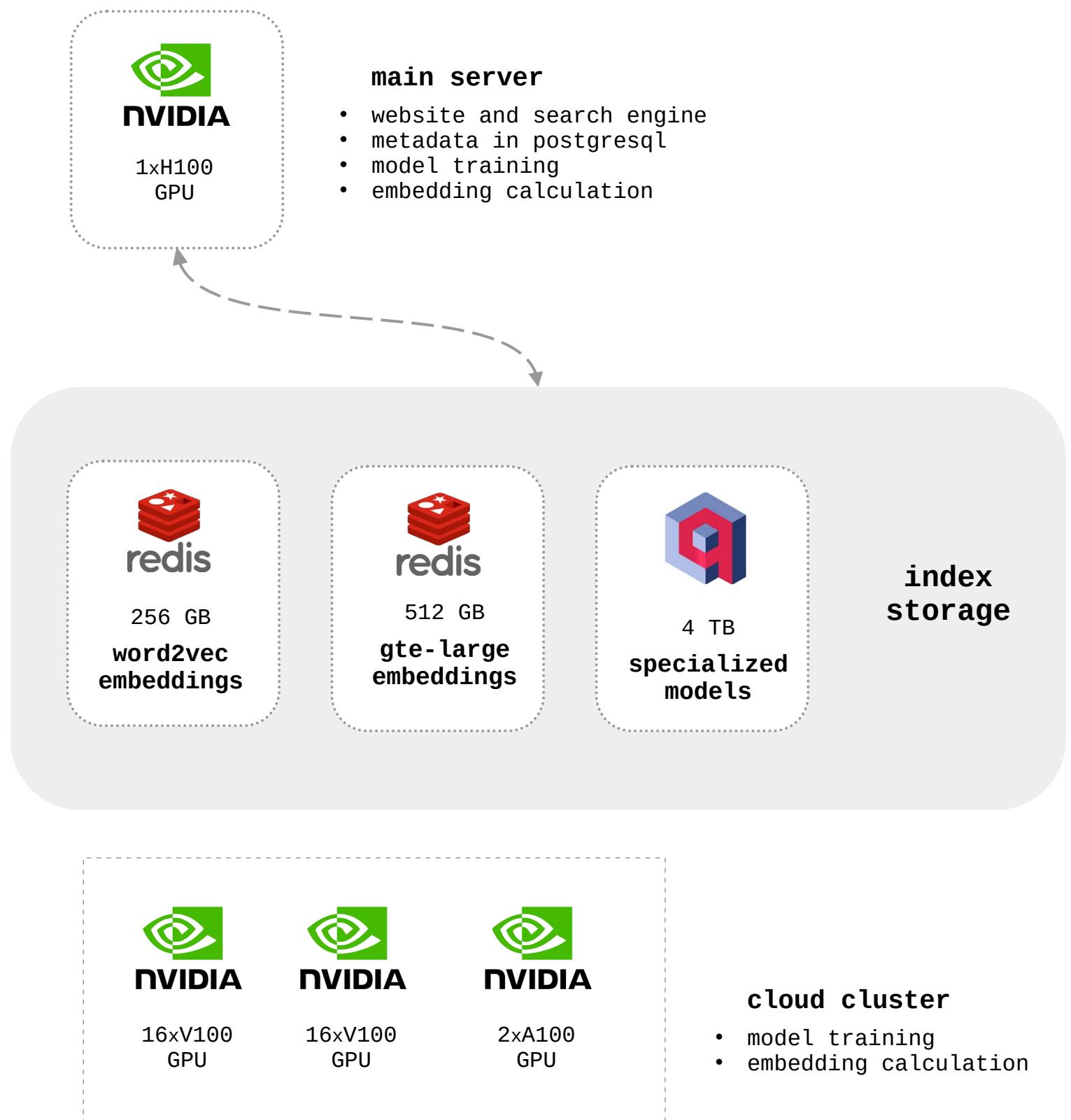
The emergence of an AI-powered chatbot that can generate human-like sentences and write coherent essays has caught the world's attention. This paper discusses the historical overview of chatbots and the technology behind Chat Generative Pre-trained Transformer, better known as ChatGPT. Moreover, potential applications of ChatGPT in various domains, including healthcare, education, and research, are highlighted. Despite promising results, there are several privacy and ethical concerns surrounding ChatGPT. In addition, we highlight some of the important limitations of the current version of ChatGPT. We also ask ChatGPT to provide its point of view and present its responses to several questions we attempt to answer.

similar articles

- 2023 91% ChatGPT: Applications, Opportunities, and Threats
- 2023 90% Unlocking the Potential of ChatGPT: A Comprehensive Exploration of its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing
- 2023 90% Decoding ChatGPT: A Taxonomy of Existing Research, Current Challenges, and Possible Future Directions
- 2023 87% Unveiling Security, Privacy, and Ethical Concerns of ChatGPT
- 2023 87% ChatGPT in the Age of Generative AI and Large Language Models: A Concise Survey
- 2023 86% Revolutionizing Customer Interactions: Insights and Challenges in Deploying ChatGPT and Generative Chatbots for FAQs
- 2023 86% A Review of ChatGPT Applications in Education, Marketing, Software Engineering, and Healthcare: Benefits, Drawbacks, and Research Directions
- 2023 86% ChatGPT and a New Academic Reality: Artificial Intelligence-Written Research Papers and the Ethics of the Large Language Models in Scholarly Publishing
- 2023 86% Transformative Effects of ChatGPT on Modern Education: Emerging Era of AI Chatbots
- 2023 86% Ethical ChatGPT: Concerns, Challenges, and Commandments
- 2023 85% Critical Role of Artificially Intelligent Conversational Chatbot

APPENDIX

technical setup



All servers are running Ubuntu. I got free access to cloud cluster for testing.

APPENDIX

change of semantic meaning of the word «intelligence» in word2vec when trained on the domain-specific texts

**word2vec trained
on Google News
dataset**

The word intelligence belongs to cluster: interrogator, Terrorist, KSM, terrorism, VBIED, civilians, Westerners, intelligence, defector, PKK, aqi, PoW, operatives, INtelligence, terrorists, jihadist, intel, ReBeL, diplomatic, WMD, rearm, attacks, Iraq, Taliban, aTtack, militant, terror, weapons, iNTelligence, rearming, air strikes, extremists, bombs, rebel, Intelligence, enemy, bombing, exfiltrating, OBL, Afghanistan, Syria, tortured, spies, gunfighting, Baghdad, ATtack, exfiltrate, insurgency, landmine, Pakistani, combatants, dissidents, prisoners, Mosul, infiltrations, missiles, IARA, bomb, explosives, interrogators, terrorist, CIA, Iraqi, forces, spy, NIE, IED, defectors, Tehran, insurgents, Iran, insurgent, attack, bioweapons, peacekeepers, Saddam, tradecraft, weaponization, AQI, infiltrator, interrogation, interrogations, extremist, informants, Karadzic

**word2vec
trained on
arXiv dataset**

The word intelligence belongs to cluster: humans, chemistries, understanding, brains, intelligence, animals, beings, minds, interaction, Dilemmas, Innate, INteraction, life, impairments, INtelligence, C0operation, Childhood, lIving, reflection, conscious, compositionality, Morality, Chemistries, liFe, Creatures, FiSH, mind, sciences, organisms, Bodies, cognitive, appraisal, INTELLIGENCE, thinking, gameplay, embodiment, iNTelligent, creatures, mathematicians, ToM, living, cooperation, Functioning, childhood, perceptions, interactiOn, empathy, creativity, Mathematicians, Metaphors, metaphors, societies, fish, COgnitive, Beings, morality, Embodiment, collaboration, infants, CoGnition, formation, consciousness, Perceptions, expertise, requisites, involvement, faculties, nuances, developmental, agency, understanding, Organisms, Societies, COGnitive, undERstanding, workings, bodies, InteractiOn, cognition, interaction, Empathy, wisdom, self-awareness, Observers, functioning, dilemmas, Appraisal, machines, innate, immune, observers, underStAnding, eXpertise, inteRactiOn, wrongness, capacities, curiosity, EMPATHY

APPENDIX

generating random search queries for performance evaluation

results for: can you generate 32 random search queries (not in question form) related to science, and format the result as JSON array?

embedding model: aibertgpl
retrieval took 0.3587 sec.

```
answer by phi-3-mini: ```json [ "effects of climate change on polar bear habitats", "impact of deforestation on global carbon levels", "advances in renewable energy technologies", "breakthroughs in quantum computing", "recent discoveries in exoplanet atmospheres", "latest research on CRISPR gene editing", "innovations in artificial intelligence algorithms", "new findings in dark matter and dark energy", "breakthroughs in cancer immunotherapy", "advancements in nanotechnology for drug delivery", "recent developments in gravitational wave detection", "latest trends in sustainable agriculture", "emerging technologies in space exploration", "breakthroughs in neuroprosthetics", "recent discoveries in deep-sea ecosystems", "innovations in 3D bioprinting", "latest research on black holes and singularities", "advancements in battery storage technology", "breakthroughs in synthetic biology", "recent developments in fusion energy", "latest findings in human genome editing", "innovations in water purification systems", "emerging trends in smart city infrastructure", "breakthroughs in personalized medicine", "recent discoveries in marine biology", "latest research on antibiotic resistance", "advancements in space telescope technology", "innovations in biodegradable plastics", "recent developments in autonomous vehicles", "breakthroughs in organ transplantation", "latest findings in climate modeling", "advancements in virtual reality technology", "emerging trends in cybersecurity", "breakthroughs in sustainable urban design", "recent discoveries in ancient DNA", "latest research on brain-computer interfaces", "innovations in clean energy storage", "recent developments in space debris mitigation", "breakthroughs in personalized nutrition", "latest findings in marine conservation", "advancements in 3D bioprinting for human organs", "emerging trends in space tourism", "breakthroughs in gene therapy for rare diseases", "recent discoveries in dark energy", "latest research on AI-driven drug discovery", "innovations in water desalination technology", "recent developments in quantum cryptography", "breakthroughs in regenerative medicine", "latest findings in space weather prediction", "advancements in sustainable packaging materials", "emerging trends in space mining", "breakthroughs in personalized learning", "recent discoveries in deep-sea microbiology", "latest research on AI-driven climate modeling", "innovations in energy-efficient building design", "recent developments in space-based solar power", "breakthroughs in gene editing for agriculture", "latest findings in space-time research", "advancements in AI-driven traffic management", "emerging trends in space-based internet", "breakthroughs in personalized healthcare", "recent discoveries in space-based telescopes", "latest research on AI-driven climate change mitigation", "innovations in space habitat design", "recent developments in space-based water extraction", "breakthroughs in
```

APPENDIX

retrieval results

Query:
ethics of AI

word2vec

1. Overview of psychiatric ethics V: utilitarianism and the ethics of duty.
2. Overview of psychiatric ethics III: principles-based ethics.
3. [Forensic expertise in the Defense Ministry of Russian Federation: modern status and perspectives of development].
4. Analysis of Texas & New Mexico Hospice Organization's new Code of Ethics.

gte-large-en-v.15

1. The Ethics of AI Ethics -- An Evaluation of Guidelines
2. Ethical Considerations in Artificial Intelligence Courses
3. From the Ground Truth Up: Doing AI Ethics from Practice to Principles
4. Artificial Intelligence: the global landscape of ethics guidelines

BioBERT

1. Principles alone cannot guarantee ethical AI
2. Ethical aspects of sexual medicine.
3. Would someone please explain what it means to be ethical?
4. Are there limits to respect for autonomy in bioethics?

AI-RoBERTa

1. Kunstliche Intelligenz, quo vadis?
2. Philosophy in the Face of Artificial Intelligence
3. [From the Ethics of AI to the Ethics of Consciousness: Ethical Aspects of Computational Psychiatry]
4. Towards AI Forensics: Did the Artificial Intelligence System Do It?

AI-BERT-GPL

1. From the Ground Truth Up: Doing AI Ethics from Practice to Principles
2. AI and Ethics -- Operationalising Responsible AI
3. The Ethics of AI Ethics -- An Evaluation of Guidelines
4. Ethics in the Age of AI: An Analysis of AI Practitioners' Awareness and Challenges

APPENDIX

retrieval results

Query:

AI

word2vec

1. [Expression and deletion analysis of EcoR II endonuclease and methylase gene].
2. A method for production of antibodies to human T-cell receptor beta-chain variable regions.
3. Diversity of T-cell receptor V alpha, V beta, and CDR3 expression by myelin basic protein-specific human T-cell clones.
4. Human cDNA clones for an alpha subunit of Gi signal-transduction protein.

gte-large-en-v.15

1. Long short-term memory.
2. Efficient Estimation of Word Representations in Vector Space
3. An efficient learning procedure for deep Boltzmann machines.
4. Deep, narrow sigmoid belief networks are universal approximators.

BioBERT

1. Purification of an angiotensin II binding protein by using antibodies to a peptide encoded by angiotensin II complementary RNA.
2. AAMP, a newly identified protein, shares a common epitope with alpha-actinin and a fast skeletal muscle fiber protein.
3. An ARIA-interacting AP2 domain protein is a novel component of ABA signaling.
4. A Geometric Realization of Symmetric Pairs of Type AIII

AI-RoBERTa

1. Can Artificial Intelligence Do Everything That We Can?
2. AI-Powered Non-Contact In-Home Gait Monitoring and Activity Recognition System Based on mm-Wave FMCW Radar and Cloud Computing
3. Tentacular Artificial Intelligence, and the Architecture Thereof, Introduced
4. Artificial intelligence applications in the intensive care unit.

AI-BERT-GPL

1. Towards Verified Artificial Intelligence
2. Artificial Intelligence (AI) Methods in Optical Networks: A Comprehensive Survey
3. Industrial Artificial Intelligence
4. AI applications to medical images: From machine learning to deep learning.

APPENDIX

retrieval results

Query:
embedding

word2vec

1. Learning to Embed Categorical Features without Embedding Tables for Recommendation
2. Robust Image Watermarking based on Cross-Attention and Invariant Domain Learning
3. EmbeddingTree: Hierarchical Exploration of Entity Features in Embedding
4. Deep Recursive Embedding for High-Dimensional Data.

gte-large-en-v.15

1. Efficient Estimation of Word Representations in Vector Space
2. Long short-term memory.
3. Building Location Embeddings from Physical Trajectories and Textual Representations
4. Embedding Projector: Interactive Visualization and Interpretation of Embeddings

BioBERT

1. [Rapid method of obtaining ultrathin sections in monolayer embedding].
2. Embedding mouse embryos and tissues in wax.
3. EVA-enhanced embedding medium for histological analysis of 3D porous scaffold material.
4. Improved Epon embedding for biomaterials.

AI-RoBERTa

1. A Yoneda-Style Embedding for Virtual Equipments
2. Embedding Projector: Interactive Visualization and Interpretation of Embeddings
3. EmbeddingTree: Hierarchical Exploration of Entity Features in Embedding
4. Multitask Text-to-Visual Embedding with Titles and Clickthrough Data

AI-BERT-GPL

1. Embeddings of representations
2. Maximising the number of regions when embedding a N-cycle graph
3. Tutorial on NLP-Inspired Network Embedding
4. Coxeter embeddings are injective

APPENDIX

retrieval results

Query:

llama

word2vec

1. Esophageal dysfunction in four alpaca crias and a llama cria with vascular ring anomalies.
2. Pneumocystis carinii infections in zoo animals.
3. Isolation of dermatophytes from domestic animals in Norway.
4. Antibody to porcine, feline and rat parvoviruses in various animal species.

gte-large-en-v.15

1. Traditional llama husbandry and breeding management in the Ayopaya region, Bolivia.
2. Characterization of plasma immunoglobulin G concentrations of llamas.
3. Lamanema chavezii (Nematoda: Molineidae): epidemiological data of the infection in South American camelids of Northwest Argentina.
4. Presence of enamel on the incisors of the llama (*Lama glama*) and alpaca (*Lama pacos*).

BioBERT

1. [Comparative histological study of the female reproductive system in the llama (*Lama guanicoe glama*). II. Oviduct, uterus, cervix, vagina].
2. Is Bigger and Deeper Always Better? Probing LLaMA Across Scales and Layers
3. Morphometric analysis of llama (*Lama glama*) sperm head.
4. Traditional llama husbandry and breeding management in the Ayopaya region, Bolivia.

AI-RoBERTa

1. [Determination of verbascoside in herba of *Galeobdolon Chinense* by RP-HPLC].
2. [RP-HPLC fingerprint of milk *Stellera chamaejasme*].
3. Reply to Comment on "Topological Oscillations of the Magnetoconductance in Disordered GaAs Layers"
4. Ruminal ciliated protozoa in bison.

AI-BERT-GPL

1. LLAMA: Leveraging Learning to Automatically Manage Algorithms
2. LLaMA: Open and Efficient Foundation Language Models
3. Embed_Llama: Using LLM Embeddings for the Metrics Shared Task
4. LLAMA: The Low-Level Abstraction For Memory Access

APPENDIX

retrieval results

Query:

how does chat GPT work?

word2vec

1. How to Network Action Alerts, So That Others Can Help.
2. Conquer job overload.
3. Accountability stops here.
4. Voluntary work abroad.

gte-large-en-v.15

1. Clinical validation of glucose pump test (GPT) compared with ultrasound dilution technology in arteriovenous graft surveillance.
2. Partial cloning and expression of mRNA coding choline acetyltransferase in the spinal cord of the goldfish, *Carassius auratus*.
3. Comparison of glucose pump test and urea test in measuring blood access flow.
4. Cytochrome P450-mediated herb-drug interaction potential of Galgeuntang.

BioBERT

1. ChatGPT Performance on Standardized Testing Exam -- A Proposed Strategy for Learners
2. Unveiling Security, Privacy, and Ethical Concerns of ChatGPT
3. An Empirical Study of Using ChatGPT for Fact Verification Task
4. Chat GPT Integrated with Voice Assistant as Learning Oral Chat-based Constructive Communication to Improve Communicative Competence for EFL earners

AI-RoBERTa

1. Let's have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations
2. ChatGPT for translators: a survey
3. ComputeGPT: A computational chat model for numerical problems
4. Conversational Ontology Alignment with ChatGPT

AI-BERT-GPL

1. Can Chat GPT solve a Linguistics Exam?
2. Is ChatGPT the ultimate Data Augmentation Algorithm?
3. ChatGPT for translators: a survey
4. ChatGPT as a Java Decompiler

APPENDIX

figures

Figure 1	An example 1024-dimension embedding vector (truncated) generated for a phrase «A quick brown fox jumped over the lazy dog» using gte-large language model	2
Figure 2	Different approaches to measure distance between two given points in a coordinate space	4
Figure 3	Search results based on word2vec. On the screenshot the user interface of the early version of search engine is shown	5
Figure 4	Example answer generated by phi-3-mini-4k-instruct model	7
Figure 5	QDrant as compared to other vector storages in terms of rate of operations per second	10
Figure 6	Main page of the search engine	A1

abbreviations

ACL	Association for Computational Linguistics
AI	Artificial Intelligence
AI-BERT-GPL	BERT model fine-tuned for the AI domain using GPL method
AI-RoBERTa	RoBERTa model fine-tuned for the AI domain using MLM
BERT	Bidirectional Encoder Representations from Transformers
BM25	Okapi Best Match 25
DistilBERT	Distilled BERT
GPT	Generative Pre-trained Transformer
HNSW	Hierarchical Navigable Small World
LLaMA	Large Language Model Meta AI
MAP	Mean average precision
MRR	Mean reciprocal rank
NDCG	Normalized discounted cumulative gain
NLI	Natural Language Inference
RAG	Retrieval Augmented Generation
RIOT	Redis Input/Output Tools
RoBERTa	Robustly Optimized BERT Pretraining Approach
TF-IDF	Term Frequency Inverse Document Frequency