Evaluating Embedding APIs for Information Retrieval

Ehsan Kamalloo[†] **Xinyu Zhang**[†] **Odunayo Ogundepo**[†] **Nandan Thakur**[†]
David Alfonso-Hermelo[§] **Mehdi Rezagholizadeh**[§] **Jimmy Lin**[†]

[†] David R. Cheriton School of Computer Science, University of Waterloo
[§] Huawei Noah's Ark Lab

ekamalloo@uwaterloo.ca

Abstract

The ever-increasing size of language models curtails their widespread availability to the community, thereby galvanizing many companies into offering access to large language models through APIs. One particular type, suitable for dense retrieval, is a semantic embedding service that builds vector representations of input text. With a growing number of publicly available APIs, our goal in this paper is to analyze existing offerings in realistic retrieval scenarios, to assist practitioners and researchers in finding suitable services according to their needs. Specifically, we investigate the capabilities of existing semantic embedding APIs on domain generalization and multilingual retrieval. For this purpose, we evaluate these services on two standard benchmarks, BEIR and MIRACL. We find that re-ranking BM25 results using the APIs is a budget-friendly approach and is most effective in English, in contrast to the standard practice of employing them as first-stage retrievers. For non-English retrieval, re-ranking still improves the results, but a hybrid model with BM25 works best, albeit at a higher cost. We hope our work lays the groundwork for evaluating semantic embedding APIs that are critical in search and more broadly, for information access.

1 Introduction

Language models (LMs), pre-trained on a massive amount of text, power dense retrieval models in ad hoc retrieval (Lin et al., 2021b). Dense retrievers (Lee et al. 2019; Karpukhin et al. 2020; Xiong et al. 2021; Khattab and Zaharia 2020; Hofstätter et al. 2021; Izacard et al. 2022; *inter alia*) essentially measure relevance via similarity between the representations of documents and queries. As LMs are rapidly scaling up to gigantic models (Radford et al. 2019; Brown et al. 2020; Lieber et al. 2021; Chowdhery et al. 2022; Smith et al. 2022, *inter alia*), their use as the backbone of dense retrieval models has become limited primarily because large language models (LLMs) are computationally expensive and deploying them on commodity hardware is cumbersome and often impractical.

To alleviate this problem, many companies, e.g., OpenAI, and Cohere, set out to offer access to their proprietary LLMs through a family of APIs. For dense retrieval, semantic embedding APIs are designed to provide LLM representations for queries as well as documents. These APIs are especially appealing in the IR ecosystem because they afford practitioners and researchers the benefit of scale and allow for wider outreach of LLMs in IR. However, although nowadays, the surge of companies offering such APIs with various model sizes has given us more options, a lack of thorough analysis of these APIs has made it more difficult to determine one's best option for a particular use-case. Besides, LLM-based APIs are often expensive and experimenting with all of them to determine the most suitable is prohibitively costly.

In this paper, we analyze embedding APIs for various realistic scenarios in ad hoc retrieval. To this end, we select three embedding APIs available on the market, i.e., OpenAI, Cohere, and Aleph-Alpha, and assess their usability and effectiveness on two crucial directions that stand at the core of most IR applications.

First, we study domain generalization where retrieval is conducted over collections drawn from a broad range of domains. Understanding for which domains embedding APIs work well or poorly elucidates their limitations while setting the stage for their wide adoption in their successful domains. We leverage the widely adopted BEIR benchmark (Thakur et al., 2021) for this purpose. On BEIR, we use the APIs as re-rankers on top of BM25 retrieved documents because the large size of document collections in BEIR makes full-ranking (i.e., first-stage retrieval) via the APIs impractical. Our results show that embedding APIs are reasonably effective re-rankers in most domains, suggesting

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 518–526

that re-ranking is not only budget-friendly, but also is effective. However, we find that on datasets collected via lexical matching, they struggle. In particular, BM25 outperforms the full-fledged embedding APIs on BioASQ (bio-medical retrieval) and Signal1M (tweet retrieval).

We also explore the capabilities of embedding APIs in multilingual retrieval where they are tested in several non-English languages, ranging from low-resource to high-resource languages. More precisely, we use MIRACL (Zhang et al., 2022), a large-scale multilingual retrieval benchmark that spans 18 diverse languages. The manageable size of the corpora allow us to evaluate the APIs as full-rankers as well as re-rankers. We find that the winning recipe for non-English retrieval is not re-ranking, unlike retrieval on English documents. Instead, building hybrid models with BM25 yields the best results. Our experiments also indicate that the APIs are powerful for low-resource languages, whereas on high-resource languages, open-source models work better.

Overall, our findings offer insights on using embedding APIs in real-world scenarios through two crucial aspects of IR systems. In summary, our key contributions are:

- We extensively review the usability of commercial embedding APIs for realistic IR applications involving domain generalization and multilingual retrieval.
- We provide insights on how to effectively use these APIs in practice.

We hope our work lays the groundwork for thoroughly evaluating APIs that are critical in search and more broadly, for information access.

2 Related Work

Sentence Embeddings. Numerous studies have attempted to build universal representations of sentences using supervision via convolutional neural networks (Kalchbrenner et al., 2014), recurrent neural networks (Conneau et al., 2017), or Transformers (Cer et al., 2018). Other approaches learn sentence embeddings in a self-supervised fashion (Kiros et al., 2015) or in an unsupervised manner (Zhang et al., 2020; Li et al., 2020). Recent techniques frame the task as a contrastive learning problem (Reimers and Gurevych, 2019; Li et al., 2020; Gao et al., 2021; Kim et al., 2021; Ni et al., 2022). Embedding APIs largely follow a similar strategy to generate sentence representations (Neelakantan et al., 2022).

Dense Retrieval. While the paradigm has been around for a long time (Yih et al., 2011), the emergence of pre-trained LMs brought dense retrieval (Lee et al., 2019; Karpukhin et al., 2020) to the mainstream in IR. Recent dense retrieval models adopt a bi-encoder architecture and generally use contrastive learning to distinguish relevant documents from non-relevant ones (Lin et al., 2021b), similar to sentence embedding models. LMs are shown to be an effective source to extract representations (Karpukhin et al., 2020; Xiong et al., 2021; Hofstätter et al., 2021; Khattab and Zaharia, 2020; Izacard and Grave, 2021; Izacard et al., 2022). This essentially means that with LMs as the backbones and analogous objectives, dense retrievers and sentence embedding models have become indistinguishable in practice.

3 APIs

Semantic embedding APIs are generally based on the so-called *bi-encoder* architecture, where queries and documents are fed to a fine-tuned LM in parallel (Seo et al., 2018; Karpukhin et al., 2020; Reimers and Gurevych, 2019). The key ingredient of bi-encoders is contrastive learning, whose objective is to enable models to distinguish relevant documents from non-relevant ones. In our experiments, we adopt the following semantic embedding APIs, presented alphabetically:[1]

Aleph-Alpha: This company has trained a family of multilingual LMs, named `luminous`,[2] with three flavours in size, `base` (13B), `extended` (30B), and `supreme` (70B). The `luminous` models support five high-resource languages: English, French, German, Italian, and Spanish. However, no information is available about the data on which these LMs are trained. We used $\texttt{luminous}_{\texttt{base}}$ that projects text into 5120-dimension embedding vectors.

Cohere: This company offers LMs for producing semantic representations in two sizes: `small` (410M) and `large` (6B), generating 1024-dimension and 4096-dimension embedding vectors,

[1] Information about the number of parameters is obtained from `https://crfm.stanford.edu/helm/latest/`, accessed on May 15, 2023.

[2] `https://docs.aleph-alpha.com/docs/introduction/luminous/`

respectively. Models are accompanied by model cards (Mitchell et al., 2019).[3] Cohere also provides a multilingual model, `multilingual-22-12`,[4] that is trained on a large multilingual collection comprising 100+ languages. The data consists of 1.4 billion question/answer pairs mined from the web. The multilingual model maps text into 768-dimension embedding vectors.

OpenAI: The company behind the GPT models (Radford et al., 2019; Brown et al., 2020; Ouyang et al., 2022) also offers an embedding service. We use the recommended second-generation model, `text-embedding-ada-002` (Neelakantan et al., 2022) that embeds text into a vector of 1536 dimensions. The model, initialized from a pre-trained GPT model, is fine-tuned on naturally occurring paired data with no explicit labels, mainly scraped from the web, using contrastive learning with in-batch negatives.

All the APIs described above use Transformer-based language models (Vaswani et al., 2017), but differ from each other in various ways:

- **Model architecture:** The companies built their models in different sizes, with differences in the number of hidden layers, number of attention heads, the dimension of output layers, etc. Other subtle differences in the Transformer architecture are also likely, e.g., where to apply layer normalization in a Transformer layer (Xiong et al., 2020). Additional differences lie in the vocabulary because of different tokenization methods in the pre-trained LM that was used to initialize these embedding models for fine-tuning.

- **Training:** While contrastive learning is at the core of these models, they may vary substantially in details, e.g., the contrastive learning objective and negative sampling strategies. Also, the choice of hyper-parameters such as the number of training steps, learning rate, and optimizer is another key difference.

- **Data:** Chief among the differences is the data on which the embedding models are trained. As OpenAI and Cohere state in their documentation, the data is mostly mined from the web, but the details of the data curation process remain largely unknown. In addition, considering that each company has its own models, differences in pre-training corpora form yet another important variable in the complex process of building embedding APIs.

These distinctions may potentially lead to substantial differences in the overall effectiveness of the embedding APIs. Nevertheless, due to the non-disclosure of several details by the API providers, it remains challenging to identify the specific factors that contribute to the strengths and weaknesses of embedding models. Yet, as the number of such APIs continues to grow, we believe that high-level comparisons on standard benchmarks can provide valuable insights into how well these models operate under various practical scenarios. For practitioners building systems on top of these APIs, this comparative analysis is useful as they are primarily interested in the end-to-end effectiveness and performance of these APIs and are not typically concerned with their minutiae.

3.1 Usability

One of the critical advantages of using embedding APIs is their ease-of-use. For IR applications, even running an LLM to encode large document collections requires hefty resources, let alone training retrieval models. Thus, the emergence of such APIs makes LLMs more accessible to the community and paves the way for faster development of IR systems—this is most definitely a positive development. However, these advantages rest on the usability of the APIs. In this section, we briefly overview some factors that affect the usability of embedding APIs.

Setup. Basic information on how users can set up the proper environment to use the embedding APIs is the first step. All three companies provide detailed introductory documentation for this purpose. The procedure is nearly identical for all three at a high level: users need to create an account and generate an API key for authentication. The companies also furnish web interfaces that enable users to monitor their usage history and available credit, in addition to configuring limits to prevent unintended charges.

Client libraries. All three companies have developed open-source client libraries to facilitate access to the APIs. OpenAI provides libraries for Python and Node.js; there are also unofficial community libraries for other programming languages. Cohere

[3] https://docs.cohere.ai/docs/representation-card
[4] https://txt.cohere.ai/multilingual/

offers development toolkits in Python, Node.js, and Go. Aleph-Alpha provides a library in Python.

All libraries are structured in a similar way. One difference we notice is that Cohere has a text truncation feature when the input text exceeds the API's input length limit. OpenAI and Aleph-Alpha raise an error in this case, meaning that API users need to implement additional checks to avoid such exceptions. On the other hand, Cohere's API can truncate text from the left or the right, and can also provide an average embedding for long texts up to 4096 tokens by averaging over 512-token spans.

Documentation. All three companies provide a technical API reference, explaining inputs, responses, and errors of their APIs. Additionally, all companies provide tutorials and blog posts with examples on how to use their client libraries.

Latency. The APIs are all offered with a liberal rate limit, i.e., OpenAI at 3K requests per minute, and Cohere at 10K requests per minute.[5] We find that API calls are mostly reliable and request service errors are scattershot. Each API call takes up to roughly 400ms, consistent across all three companies (at least at the time of our experiments). However, latency presumably depends on the server workload and other factors because we observe variability at different points in time.

We also find that latency depends on the input length, as computing embeddings for queries is generally faster than computing embeddings for documents (as expected). Finally, we appreciate that Cohere's and OpenAI's APIs support bulk calls of up to 96 and 2048 texts per call, respectively, whereas for Aleph-Alpha, only one text can be passed in each API call. This bulk call feature considerably speeds up encoding document collections.

Cost. Our analysis is based on information reported as of Feb 1, 2023. OpenAI and Aleph-Alpha charge based on the number of tokens and model size: `ada2` and $\text{luminous}_{\text{base}}$ cost \$0.0004 USD and €0.078 ≈ \$0.086[6] per 1,000 tokens. On the other hand, Cohere follows a simpler cost structure, charging based only on the number of API calls, i.e., \$1.00 USD per 1,000 calls. Our re-ranking experiments on BEIR cost around \$170 USD on OpenAI, whereas it would cost roughly \$2,500 USD on Cohere based on their quoted prices. The cost of our re-ranking experiments on MIRACL for three languages (German, Spanish, and French) hovers around €116 ≈ \$128 using Aleph-Alpha and Cohere. Cohere offers a free-tier access with a restricted API call rate limit of 100 calls per minute, which we opted for, albeit sacrificing speed.

[5]We were not able to find the rate limits for Aleph-Alpha.
[6]€1.00 ∼ \$1.10 as of Feb 1, 2023

4 Experiments

In this section, our main goal is to evaluate embedding APIs in two real-world scenarios that often arise in IR applications: domain generalization and multilingual retrieval.

4.1 BEIR

We first evaluate the generalization capabilities of embedding APIs across a variety of domains. To this end, we measure their effectiveness on BEIR (Thakur et al., 2021), a heterogeneous evaluation benchmark intended to gauge the domain generalization of retrieval models. BEIR consists of 18 retrieval datasets across 9 domains and Thakur et al. (2021) showed that BM25 is a strong baseline, surpassing most dense retrieval models.

We adopt the embedding API as a re-ranking component on top of BM25 retrieved results. Re-ranking is a more realistic scenario, compared to full ranking, because the number of documents to encode in re-ranking is commensurate with the number of test queries, which is orders of magnitude smaller than the collection size, usually comprising millions of documents. Thus, re-ranking is more efficient and cheaper than full ranking.

For the BM25 retrieval, we use Anserini (Yang et al., 2018) to index the corpora in BEIR and retrieve top-100 passages for each dataset. Then, the queries and the retrieved passages are encoded using the embedding APIs. We reorder the retrieval output based on the similarity between query embeddings and those of the passages.

In addition to BM25, our baselines include the following dense retrieval models:

- TASB (Hofstätter et al., 2021), a prominent dense retrieval model that leverages topic-aware sampling of queries during training to construct more informative and more balanced contrastive examples in terms of sample difficulty. TASB is built on DistilBERT (Sanh et al., 2019) and is fine-tuned on MS MARCO (Bajaj et al., 2018).

- cpt (Neelakantan et al., 2022), an earlier version of OpenAI's embedding service.

Task	Domain	*Full-ranking*			*BM25 Top-100 Re-rank*			
		BM25	**TASB**	**cpt-S**	**TASB**	**Cohere$_{\texttt{large}}$**	**Cohere$_{\texttt{small}}$**	**OpenAI$_{\texttt{ada2}}$**
TREC-COVID	Bio-Medical	0.595	0.319	0.679	0.728	0.801	0.776	**0.813**
BioASQ	Bio-Medical	**0.523**	0.481	-	0.467	0.419	0.423	0.491
NFCorpus	Bio-Medical	0.322	**0.360**	0.332	0.334	0.347	0.324	0.358
NQ	Wikipedia	0.306	0.463	-	0.452	**0.491**	0.453	0.482
HotpotQA	Wikipedia	0.633	0.584	0.594	0.628	0.580	0.523	**0.654**
FiQA-2018	Finance	0.236	0.300	0.384	0.308	**0.411**	0.374	**0.411**
Signal-1M	Twitter	**0.330**	0.288	-	0.329	0.306	0.295	0.329
TREC-NEWS	News	0.395	0.377	-	0.436	0.461	0.447	**0.495**
Robust04	News	0.407	0.428	-	0.399	0.489	0.467	**0.509**
ArguAna	Misc.	0.397	0.427	0.470	0.436	0.467	0.438	**0.567**
Tóuche-2020	Misc.	**0.442**	0.163	0.285	0.292	0.276	0.275	0.280
CQADupStack	StackEx.	0.302	0.314	-	0.324	**0.411**	0.384	0.391
Quora	Quora	0.789	0.835	0.706	0.841	**0.886**	0.866	0.876
DBPedia	Wikipedia	0.318	0.384	0.362	0.389	0.372	0.344	**0.402**
SCIDOCS	Scientific	0.149	0.149	-	0.156	**0.194**	0.182	0.186
FEVER	Wikipedia	0.651	0.700	0.721	0.728	0.674	0.617	**0.773**
Climate-FEVER	Wikipedia	0.165	0.228	0.185	0.243	**0.259**	0.246	0.237
SciFact	Scientific	0.679	0.643	0.672	0.661	0.721	0.670	**0.736**
Avg. nDCG@10		0.424	0.414	-	0.453	0.476	0.450	**0.500**

Table 1: Results (nDCG@10) on the BEIR benchmark for full-ranking and BM25 re-ranking experiments. **cpt-S** is the predecessor of `ada2` with the same number of parameters; results are copied from Neelakantan et al. (2022).

The results are presented in Table 1.[7] TASB re-ranking results show a +4% increase over TASB full-ranking on average, showing that re-ranking via bi-encoder models is indeed a viable method. We observe that OpenAI's `ada2` is the most effective model, surpassing TASB and Cohere$_{\texttt{large}}$ by +4.7% and +2.4% on average, respectively. However, Cohere$_{\texttt{large}}$ outperforms `ada2` on 5 tasks. Specifically, Cohere$_{\texttt{large}}$ achieves the highest nDCG@10 on NQ (question answering), SCIDOCS (citation prediction), Climate-FEVER (fact verification), and both duplicate question retrieval tasks, i.e., CQADupStack, and Quora. Also, we observe that Cohere$_{\texttt{small}}$ trails Cohere$_{\texttt{large}}$ by 2.6% on average and is nearly on par with TASB.

Finally, an interesting observation is that BM25 leads all other models on 3 tasks: BioASQ, Signal-1M, and Tóuche-2020. These are datasets collected based on lexical matching, suggesting that embedding APIs struggle in finding lexical overlaps.

4.2 Multilingual Retrieval: MIRACL

We further assess the embedding APIs in the multilingual retrieval setting, where the aim is to build retrieval models that can operate in several languages while maintaining their retrieval effectiveness across languages. For this purpose, we use MIRACL (Zhang et al., 2022), a large-scale multilingual retrieval benchmark that spans 18 languages with more than 725K relevance judgments collected from native speakers.

We test Cohere's multilingual model as well as Aleph-Alpha's `luminous` on MIRACL. OpenAI does not recommend using their embeddings service for non-English documents and thus their API was omitted from this experiment. Analogous to the previous experiment, we adopt a re-ranking strategy on top-100 passages retrieved by BM25. For Cohere, we carry out full-ranking retrieval to draw a comparison with first-stage retrieval models. We also construct a hybrid model combining BM25 and Cohere by interpolating their normalized retrieval scores, following Zhang et al. (2022). The baselines are also taken from that paper: BM25, mDPR, and the hybrid model mDPR+BM25. We reuse the indexes provided in Pyserini (Lin et al., 2021a) to generate the baseline runs. For all models, we measure nDCG@10 and Recall@100.

The results on the MIRACL dev set are reported in Table 2. Re-ranking BM25 via Cohere yields better overall results (0.542), compared to full-ranking (0.512), which is consistent with our observation on BEIR. However, while the two re-ranking models, `luminous` and Cohere, surpass BM25 on all languages, they lag behind the full-ranking hybrid models. The results show that the winning recipe here is to build a hybrid model, i.e., first perform retrieval on the entire corpus and then combine the

[7]We did not test Aleph-Alpha's `luminous` on BEIR due to budget constraints.

Language	# Q	*Full-ranking* BM25	mDPR	mDPR+BM25	Cohere	*BM25 Top-100 Re-rank* Cohere+BM25	Cohere	luminous$_{base}$
Arabic	2,896	0.481	0.499	0.673	0.617	**0.686**	0.667	-
Bengali	411	0.508	0.443	0.654	0.594	**0.676**	0.634	-
German	305	0.226	0.490	**0.565**	0.436	0.468	0.414	0.396
Spanish	648	0.319	0.478	**0.641**	0.233	0.349	0.507	0.482
Persian	632	0.333	0.480	**0.594**	0.471	0.520	0.484	-
Finnish	1,271	0.551	0.472	0.672	0.634	**0.716**	0.675	-
French	343	0.183	0.435	**0.523**	0.462	0.434	0.443	0.415
Hindi	350	0.458	0.383	0.616	0.493	**0.623**	0.573	-
Indonesian	960	0.449	0.272	0.443	0.446	**0.565**	0.505	-
Japanese	860	0.369	0.439	**0.576**	0.460	0.557	0.516	-
Korean	213	0.419	0.419	**0.609**	0.496	0.597	0.546	-
Russian	1,252	0.334	0.407	**0.532**	0.469	0.528	0.447	-
Swahili	482	0.383	0.299	0.446	**0.611**	0.608	0.543	-
Telugu	828	0.494	0.356	0.602	0.613	**0.686**	0.638	-
Thai	733	0.484	0.358	0.599	0.546	**0.678**	0.606	-
Yoruba	119	0.019	0.396	0.374	**0.762**	0.735	0.629	-
Chinese	393	0.180	0.512	**0.526**	0.365	0.416	0.389	-
Avg. nDCG@10		0.364	0.420	0.567	0.512	**0.579**	0.542	-

Table 2: Results (nDCG@10) on the MIRACL dev set across 17 languages for the full-ranking and re-ranking experiments. **# Q** indicates the number of queries in the dev set. Luminous only supports German, Spanish, and French.

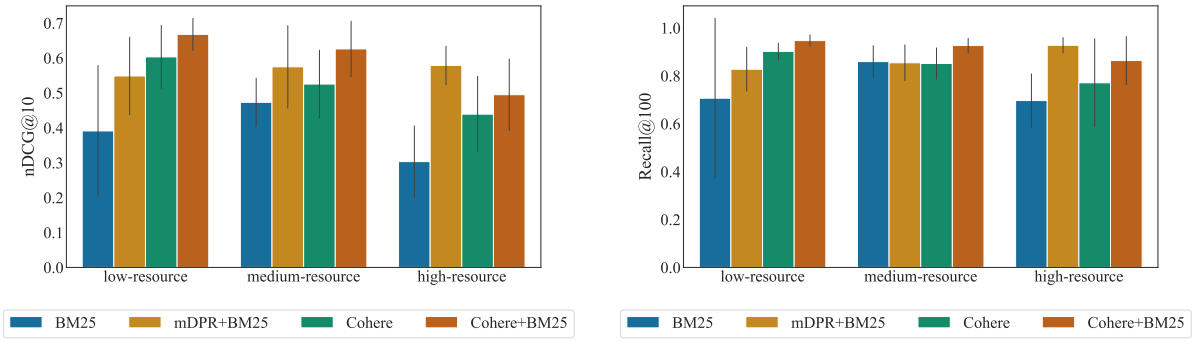

Figure 1: Average nDCG@10 (left) and Average Recall@100 (right) of full-ranking models on the MIRACL dev set for different categories of languages in terms of their available resources: low (Bengali, Hindi, Swahili, Telugu, Thai, and Yoruba), medium (Finnish, Indonesian, and Korean), and high (Arabic, German, Spanish, Persian, French, Japanese, Russian, and Chinese). The error bars show the standard deviation of nDCG@10 and Recall@100.

results with BM25. In particular, Cohere+BM25 achieves the highest average nDCG@10, outperforming the other models on 7 languages. The second best model overall is the other hybrid model, mDPR+BM25, trailing Cohere+BM25 by -1.2%.

We further investigate how the models perform on low-, medium-, and high-resource languages. To this end, following the categorization of Wu and Dredze (2020), we group languages into three categories based on the number of articles they contain in Wikipedia, reported in Zhang et al. (2022): low-resource (<200K), medium-resource (>200K but <600K), and high-resource (>600K). We measure the average nDCG@10 and Recall@100 for each language category. The results are visualized in Figure 1. The effectiveness of BM25 on low-resource languages is nearly on par with its effectiveness on high-resource languages. Interestingly, mDPR+BM25 consistently performs well across the three language categories. On the other hand, Cohere's hybrid and standalone models excel on low-resource languages and are competitive with mDPR+BM25 on medium-resource languages. However, on high-resource languages, mDPR+BM25 outperforms Cohere's hybrid model due in part to the prevalence of text in these languages during mBERT pre-training (Wu and Dredze, 2020) in mDPR.

5 Conclusion

The incredible capabilities of Transformer-based language models at scale have attracted a handful of companies to offer access to their proprietary LLMs via APIs. In this paper, we aim to qualitatively and quantitatively examine semantic embedding APIs that can be used for information retrieval. Our primary focus is to assess existing APIs for domain generalization and multilingual retrieval. Our findings suggest that re-ranking BM25 results is a suitable and cost-effective option for English; on the BEIR benchmark, $\text{OpenAI}_{\texttt{ada2}}$ performs the best on average. In multilingual settings, while re-ranking remains a viable technique, a hybrid approach produces the most favorable results. We hope that our insights aid practitioners and researchers in selecting appropriate APIs based on their needs in this rapidly growing market.

Limitations

Similar to other commercial products, embedding APIs are subject to changes that could potentially impact their effectiveness, pricing, and usability. Thus, it is important to note that our findings are specific to the APIs accessed during January and February 2023. Nevertheless, we believe our evaluation framework can serve to thoroughly assess future releases of these APIs.

Moreover, we limit our focus to the effectiveness and robustness of semantic embedding APIs. Nonetheless, safe deployment of retrieval systems for real-world applications necessitates the evaluation of their fairness as well as additional considerations. Despite their scale, language models have been found to learn, and sometimes perpetuate societal biases and harmful stereotypes ingrained in the training corpus (Bender et al., 2021). Consequently, it is crucial to assess potential biases in the embedding APIs with respect to protected and marginalized groups. This paper does not delve into this aspect of API evaluation and further research is required to examine these and other issues in real-world applications.

Acknowledgements

We thank Aleph-Alpha for providing us with credits to explore and test their APIs. This research was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. Association for Computing Machinery.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling

language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122. Association for Computing Machinery.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48. Association for Computing Machinery.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2528–2540, Online. Association for Computational Linguistics.

Jamie Kiros, Yukun Zhu, Russ R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28, pages 3294–3302. Curran Associates, Inc.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. Technical report, AI21 Labs.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021b. Pretrained transformers for text ranking: BERT and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229. Association for Computing Machinery.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744. Curran Associates, Inc.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Minjoon Seo, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 559–564, Brussels, Belgium. Association for Computational Linguistics.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008. Curran Associates, Inc.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, volume 119, pages 10524–10533. PMLR.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)*, 10(4):1–20.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2022. Making a MIRACL: Multilingual information retrieval across a continuum of languages. *arXiv preprint arXiv:2210.09984*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610, Online. Association for Computational Linguistics.