Unsupervised Evaluation of Semantic Retrieval by Generating Relevance Judgments with an LLM Judge

Motivation

Deep Neural Networks have proven themselves to learn powerful semantic representations (embeddings) that can be exploited to retrieve relevant texts according to their actual meaning - called Semantic Retrieval. Semantic retrieval has recently gained more attention since it is a central component in the popularized Retrieval Augmented Generation (RAG) setup. RAGs rely on semantic retrieval to select passages from an existing knowledge base that are then utilized by a generative LLM to answer an input user query. Successful semantic retrieval is therefore essential for the feasibility of a RAG architecture. In particular, retrieval performance is directly dictated by the embedding model's semantic capabilities on the given knowledge base and therefore a performant model with respect to the knowledge base should be selected.

I have a domain-specific knowledge base. How do I select the best embedding model?

Conducting model selection for an arbitrary dataset in a traditional manner is often impractical since generating ground truth retrieval labels is usually very tedious. Traditional information retrieval datasets for supervised evaluation possess human-generated relevance labels for each passage of a text corpus given a certain query. Human labeling however is deemed too laborious in many use cases where time and resources are limited. Thus, we want to address this dilemma, by investigating un-/semi-supervised retrieval evaluation methods that rely on relevance judgments provided by a generative LLM instead.

Our approach utilizes an LLM judge to generate synthetic relevance labels. Given a question and retrieved text passages, the LLM judge gives each passage a binary relevance label. In this manner, we can gauge how many relevant passages a certain retriever found. Furthermore, we refine the approach by providing each question a ground truth answer that is then used to compare with the retrieved passages. We consider this variant with the Question/Answer (Q/A) pairs as "semi-supervised". We believe that generating questions and also Q/A pairs is feasible for an arbitrary dataset within a reasonable amount of time and resources. Thus, our method is easily transferable to any unlabeled dataset to conduct retrieval model selection.

TL;DR of the experiment results:

> ***Our results reveal that LLMs are capable of detecting general performance trends. The information retrieval metrics that were computed from the LLM judgments strongly correlate with those that were computed from human relevance labels - we can report a 0.91 Pearson correlation coefficient. In general, the ranking of embedding models by retrieval performance using the classic supervised evaluation approach is mostly reproduced by our proposed unsupervised approach. Thereby, the LLM's judgments are***

more aligned with the ground truth human labels when making use of Q/A pairs. We believe the information of the desired answer adds valuable steering information for the LLM judge. Conducting retriever model selection with an LLM judge is therefore a viable option in an un-/semi-supervised setting.

Diving deeper into the problem

There have been indications that some retrievers using DNNs that are performing well on common information retrieval benchmark datasets struggle with out-of-domain data. Assuming that DNNs can interpolate quite well, their performance on benchmark datasets can only be an indicator of how they generalize when given a specific niche use case data distribution. In this context, we want to know which embedding model to select based on their retrieval performance given a specific text corpus. Besides that, we want to be able to evaluate embedding models during the fine-tuning process for a domain-specific retrieval task.

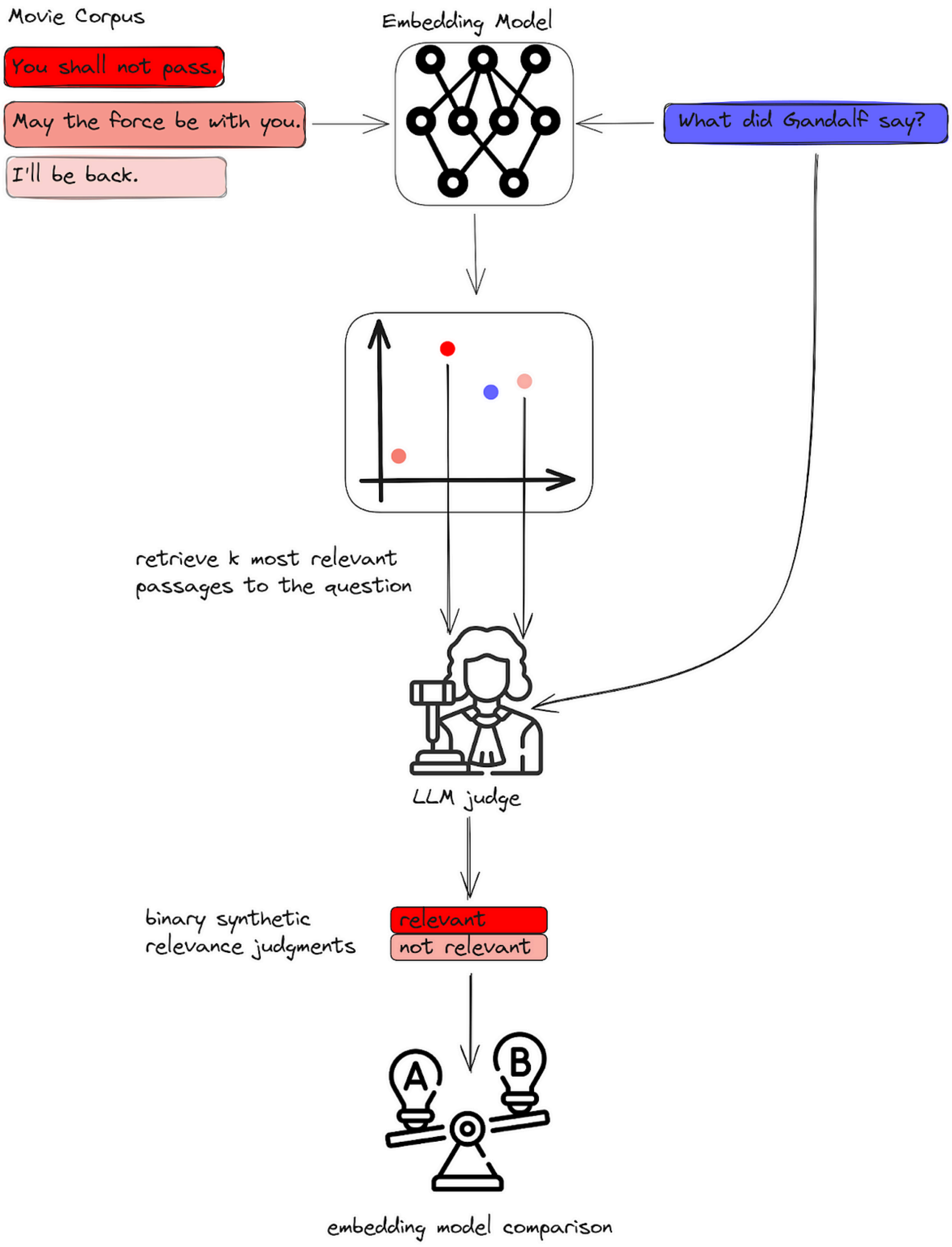
As it is difficult to make assumptions on how DNNs perform on a data distribution they did not learn, they are commonly evaluated in a data-driven manner. For the task we are concerned with, we use Information retrieval datasets for evaluation. Such datasets consist of a text corpus split into passages, queries and corresponding ground truth relevance judgments. Relevance Judgments specify for each passage how relevant it is with respect to a query. Judgments are often binary (a passage is relevant or irrelevant) due to simplicity, although they could be more fine-grained. A good collection of such datasets is BEIR. Datasets included in BEIR often rely on pre-existing links and heuristics to create relevance judgments. For a specific text corpus, we often do not have access to such a dataset and it is costly to compile one from scratch. Given *q* example queries and *p* passages, creating such a dataset requires *q*p* relevance judgments. Additionally, relevance is subjective by nature thus one would have to account for noise in the dataset. Assuming p can be quite large even in small-scale projects this exceeds the resources of most projects by far.

Still, we might get our hands on an evaluation dataset consisting of questions asked about the text corpus and even related answers. Such datasets are smaller than the ones described above by orders of magnitude. They could be compiled from scratch with less effort or already exist for a specific use case - imagine a company’s internal documentation that is queried by lexicographical search as of now. We propose to use such a dataset without access to ground truth relevance labels while relevance judgments are generated by an LLM.

This leads us to the question: can we use an unsupervised evaluation approach where an LLM judge generates relevance judgments for retrieved passages as a proxy for the performance of embedding models for Semantic Retrieval instead of relying on ground truth relevance labels?

How to compare embedding models in an unsupervised manner?

We apply a generative language model to automate the unsupervised evaluation process of semantic information retrieval models. Within this, we use the language models' general semantic capabilities to judge the relevance of retrieved passages.


Movie Corpus
You shall not pass.
May the force be with you.
I'll be back.
Embedding Model
What did Gandalf say?
retrieve k most relevant passages to the question
LLM judge
binary synthetic relevance judgments
relevant
not relevant
A
B
embedding model comparison

For an embedding model and an information retrieval dataset we compute embedding vectors for all passages in the dataset. Now, given a query for which we want to find the passages that answer it, we also compute its embedding using the same model.

We apply a distance metric to retrieve the passages whose embedding vectors are closest to the query embedding vector and imply that those are most semantically similar. We retrieve the k most similar passages and assume these are considered the most relevant by the embedding model to answer the query. We execute this procedure for all embedding models to be evaluated and compared.

In the next step, the LLM judges the relevance of each passage with respect to a certain context. Therefore, we can simply call it the "LLM judge". For this purpose, we designed the two following approaches.

- **Question (Q) - unsupervised:** In this setting, we assume that representative questions are available with regard to a text corpus. We provide a question and a passage as context to the LLM judge as well as a steering prompt to provide the output in a desired format and ask the LLM judge to generate a synthetic binary relevance judgment of the passage with respect to the question. This setting results in a relevance classification task that was shown to be difficult for LLMs.
- **Question-Answer (QA) - semi-supervised:** In this setting we assume that representative questions and respective correct answers are available with regard to text corpus. Binary relevance judgments are generated similarly to Q while providing the question, answer and passage in the LLM judges' context. This setting reduces the relevance interpretation work for the LLM as it includes a text similarity judgment as well. Thus, we assume that QA is the less noisy approach.

We will assess the capabilities of those two approaches separately in the following.

For one embedding model, this results in k relevance judgments per sample. Of course, the LLM judgments cannot be considered similar to ground truth labels due to induced noise in their generation process. The judgments are binary vectors - 0 to indicate that a passage is irrelevant and 1 for relevant passages. From these vectors we are able to compute standard information retrieval (IR) metrics. These IR metrics can then be used to compare the embedding models' performances and to select the most suitable one based on a specific use case.

The main advantage of this method is that generating labels for the Q and Q/A settings is not as much work as conventional human-generated IR labels. While falling back on a domain expert's knowledge or crowdsourcing one should be able to compile such a dataset in orders of magnitude less time compared to a conventional IR dataset. There are ways to automate the process of compiling such datasets using LLMs as well.

But does this actually work?

We conduct experiments to verify whether our method delivers robust results. Overall, we want to find out if the labels produced by the LLM judge can be used to compute information retrieval metrics that allow us to compare the embedding models' ability as if we had used ground truth relevance labels. This implies if model A outperforms model B significantly in an IR metric based on the ground truth labels, a similar picture should be drawn when computing the metric from the synthetic labels originating from the LLM judge.

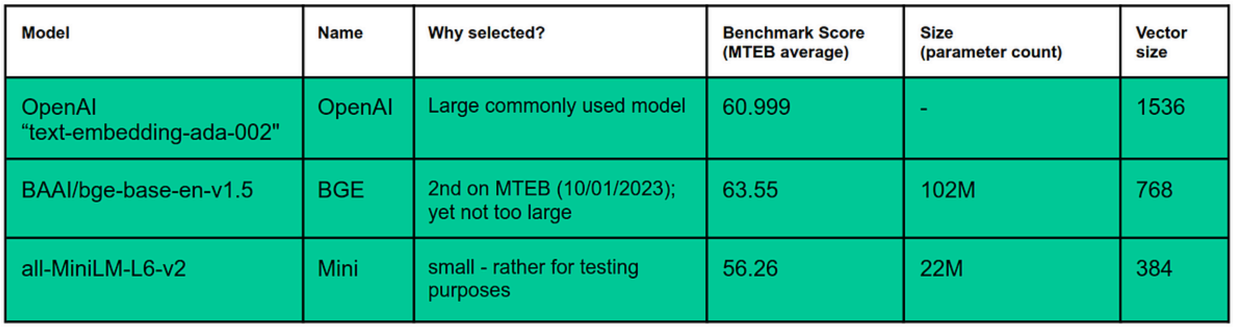
We have to compile a meaningful dataset, select embedding models to compare and design prompts for the LLM judge.

What does the dataset look like?

We used a custom biomedical dataset Mini-BioASQ consisting of a text corpus from a specific domain with 40k passages at around 350 words each. We derived it from the biomedical information retrieval dataset from the BioASQ challenge which is part of BEIR as well. We reduced its corpus of all PubMed abstracts to only the ones that are relevant to any question which made it manageable within our small-scale setup. Alongside we provide a 100-sample evaluation dataset consisting of: question—correct answer—relevant passages from the text corpus to answer the question. The ground truth relevance labels consist of a list of indices of the relevant passages. We believe that a dataset of this form and size is sufficient to generate indicative results. The following statistics sum up the dataset we compiled:

- Domain: biomedical
- # Passages: 40k
- Average # Tokens per Passage: 350
- # Queries: 4700
- Average # Tokens per Query: 13
- Average # Tokens per Answer: 56
- Average # Relevant Passages per Query: 9
- Maximum Relevant Passages per Query: 160
- Minimum Relevant Passages per Query: 0

Which embedding models are we applying?



Model	Name	Why selected?	Benchmark Score (MTEB average)	Size (parameter count)	Vector size
OpenAI "text-embedding-ada-002"	OpenAI	Large commonly used model	60.999	-	1536
BAAI/bge-base-en-v1.5	BGE	2nd on MTEB (10/01/2023); yet not too large	63.55	102M	768
all-MiniLM-L6-v2	Mini	small - rather for testing purposes	56.26	22M	384

How to effectively prompt an LLM in our use case as a judge?

We use “gpt-3.5-turbo” (LLM judge) to create the relevance judgments. In order to only get binary judgments (relevant or not) we added the following steering prompt (<binary>) into the prompts that we sent to the LLM judge. The prompts were inspired by Evaluating Open-QA Evaluation.

> NEVER give another answer than “YES” or “NO”.
> If you can’t answer “YES” or “NO”, answer: XXX
> ALWAYS answer in capital letters.
> NEVER answer anything that does not follow the format.

In our experiments on a small custom Wikipedia IR dataset Mini-Wikipedia it yielded 79% YES, 16% NO and 5% XXX—nothing else.

For *Q* we use:

> Here are a question and a retrieved passage from a text corpus from the same domain as the question.
> Can you judge whether an answer to the question can be derived from the retrieved passage, simply answer either “YES” or “NO”.
> *<binary>*
> Question: {question}; Retrieved Passage: {passage}.

For *QA* we use:

> Here are a question, the correct answer and a retrieved passage from a text corpus from the same domain as the question.
> Can you judge whether the correct answer to the question can be derived from the retrieved passage, simply answer either “YES” or “NO”.
> *<binary>*
> Question: {question}; Correct Answers: {answer} Retrieved Passage: {passage}.

How were the experiments run?

For 100 questions, we retrieve the 10 ranked passages most similar to each question. To achieve this, we use an architecture similar to Dense Passage Retrieval (DPR) computing the semantic similarity of a query and passage as the cosine similarity of their embedding vectors. To store and efficiently search for the closest embedding vectors we use a Weaviate vector database. We made extensive use of the Langchain framework as well for LLM interaction and prompt templating.

What did we find out through the experiments?

We find that although the LLM judge does not show satisfactory relevance classification capabilities, the information retrieval metrics results computed upon Q and QA judgments allow us to compare and select retrievers that also perform well in a classic supervised information retrieval evaluation setting.

Let's first look at a qualitative example

To give an intuition about the dataset and how we compare the approaches we show one qualitative example from Mini-BioASQ. Recall that our retriever gives us a list of passages that seem semantically similar to a question. Similarity is determined by using distance metrics on the embedding vector of the two texts. Thus, the list of passages can be sorted with the most similar/ closest passage at the top. We decided that we only consider the top-10 most similar passages. We can then do a look-up into our list of actually relevant passages, how well our retriever worked. Mind that it is possible that there are less than 10 relevant passages for a question - still, we would always retrieve 10.

We examine the question:

> Where in the body would the navicular bone be found?

With the answer:

> The navicular bone is located in the foot.

Using the OpenAI embedding model the following PubMed abstract was retrieved as third most semantically similar. Despite BioASQ's ground truth data it is not relevant with respect to the question.

> …The segmentation software, MIMICS was used to generate the 3D images of the bony structures of normal and varus malalignment lower extremity. Except the spaces between the adjacent surface of the phalanges fused, metatarsals, cuneiforms, cuboid, navicular, talus and calcaneus bones were independently developed to form foot and ankle complex…

Besides this abstract, the Q and QA LLM judge and ground truth labels are aligned in classifying the first two retrieved abstracts as relevant and the rest as irrelevant.

Only providing the question to the LLM judge, it finds the abstract relevant. When providing both question and answer to it, the abstract is classified as not relevant. In this particular example, we cannot say that the Q-judge is entirely wrong in its relevance judgment as the abstract seems intuitively relevant - it just lacks the knowledge about the correct answer.

It is noteworthy that for the passages retrieved by the Mini embedding model on the one hand the synthetic relevance and ground truth judgments are all the same - only the first retrieved abstract is relevant. On the other hand only one of the two actually relevant passages was retrieved at all and the abstract from above was not retrieved either which indicates that the embedding model is far from perfect.

Do LLMs know what is relevant?

To report common classification metrics, we assume that the retriever considers all 10 retrieved passages as relevant and compare them to the synthetic and ground truth relevance judgments in a classification task where 1= relevant/ 0 = irrelevant. From these insights we can assess how much noise was induced by each of the unsupervised approaches. This can help us to estimate to what degree the metrics that will be computed upon the synthetic labels could be skewed.

Judge's performance using...	Accuracy	True Positive	False Positive	False Negative	True Negative	Precision	Recall	F1
Q	0.757	0.25	0.12	0.12	0.50	0.673	0.680	0.677
QA	0.791	0.26	0.095	0.11	0.53	0.733	0.694	0.713

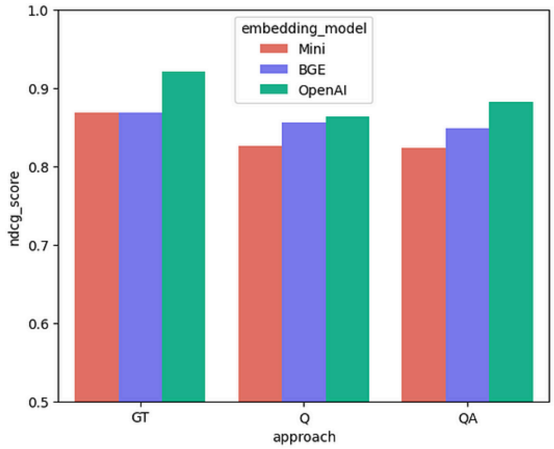
Correlation of Information

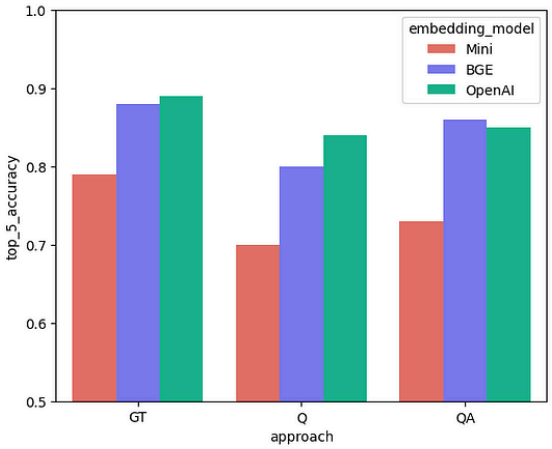
Next, we take the 10 retrieved passages, assuming again they are all predicted as relevant by the retriever and ranked according to their relevance. Consequently, we have a prediction vector of all-ones that we can set against the "real labels" that we get from the LLM judge or Ground Truth respectively. To evaluate the retriever, we apply standard IR metrics that have proven themselves in similar contexts. Amongst other standard IR metrics, we most importantly computed normalized discounted cumulative gain (NDCG) and Top-1,5,10-accuracy.

For convenience, we will call the supervised evaluation approach and a dataset that uses ground truth human-generated relevance labels **Ground Truth (GT)**. Recall, this is the form of dataset we did not have access to in an exemplary case which caused us to think about a less sophisticated and significantly smaller evaluation dataset.

Looking at two IR metrics (NDCG and Top-5-accuracy in the plots below) as an example and how the embedding models perform for each evaluation approach, we can derive the following interpretations and first assumptions:

- Only providing the question to the judge results in a worse metric performance perhaps due to more induced noise in the judgment process. A pattern that can be found across all information retrieval metrics.
- The relative bar heights in the QA bar charts are more similar to GT than the ones of Q. Thus QA seems to be a slightly better proxy for GT than Q.
- Neither when looking at only one metric nor comparing both of them, there is a clear ranking of embedding models amongst approaches. Promisingly, selecting a model based on Q or QA would have been a good (not necessarily the best) choice according to GT as well.



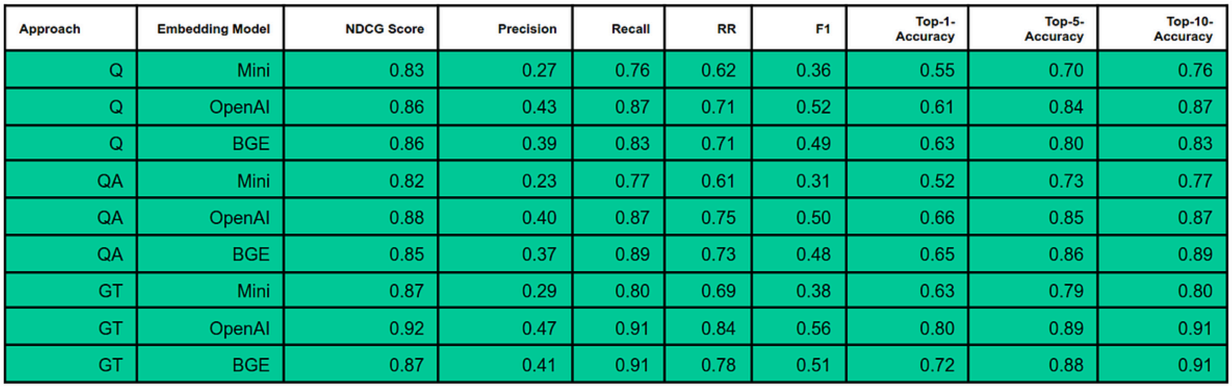


If we only had access to those two metrics, we would decide for the OpenAI embedding model. This aligns with the ground truth evaluation results which indicates the effectiveness of our method.

Yet to quantify how good of a proxy for Ground Truth each of the unsupervised approaches is we can compute the Pearson correlation coefficient between variable X (all metric outcomes for all embedding models using Q approach) and Y (all metric outcomes for all embedding models using GT). Consider that we standardized each metric for this task to make them comparable to each other. And the same for QA respectively.

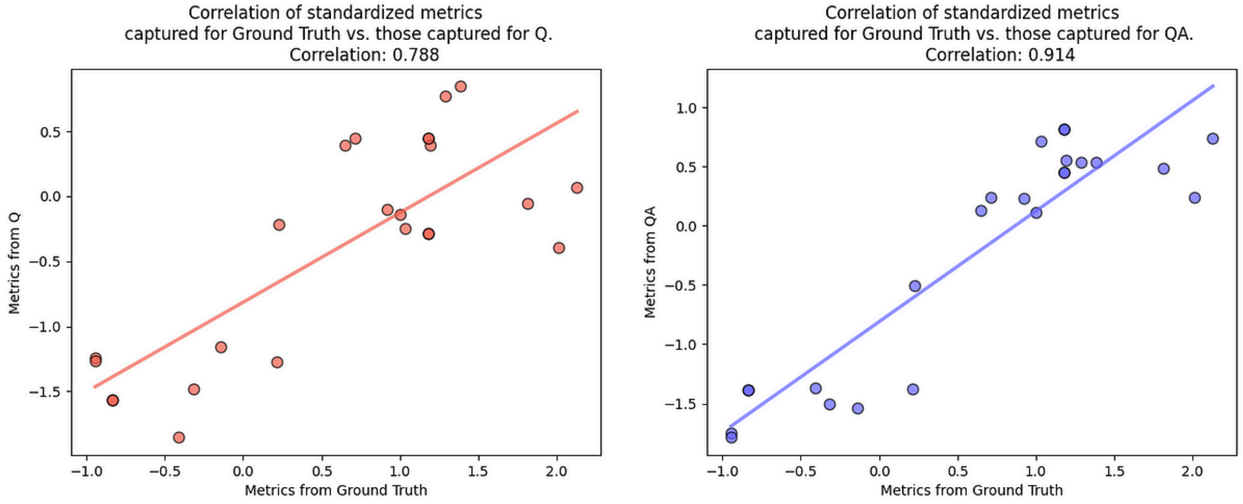
To put it another way, yield the correlation measure for Q computing the following with the values from our experiments stated below.

corr(
[ndcg_openai_q, ndcg_bge_q,… ,top-10_bge_q, 10_mini_q],
[ndcg_openai_gt,… , top-10_mini_gt]
)



Approach	Embedding Model	NDCG Score	Precision	Recall	RR	F1	Top-1-Accuracy	Top-5-Accuracy	Top-10-Accuracy
Q	Mini	0.83	0.27	0.76	0.62	0.36	0.55	0.70	0.76
Q	OpenAI	0.86	0.43	0.87	0.71	0.52	0.61	0.84	0.87
Q	BGE	0.86	0.39	0.83	0.71	0.49	0.63	0.80	0.83
QA	Mini	0.82	0.23	0.77	0.61	0.31	0.52	0.73	0.77
QA	OpenAI	0.88	0.40	0.87	0.75	0.50	0.66	0.85	0.87
QA	BGE	0.85	0.37	0.89	0.73	0.48	0.65	0.86	0.89
GT	Mini	0.87	0.29	0.80	0.69	0.38	0.63	0.79	0.80
GT	OpenAI	0.92	0.47	0.91	0.84	0.56	0.80	0.89	0.91
GT	BGE	0.87	0.41	0.91	0.78	0.51	0.72	0.88	0.91

A high correlation would indicate a good proxy evaluation approach—this implies that one could use this approach instead of the go-to supervised approach while rankings of the models would not have to be exactly the same. This means the IR metrics derived from this approach should enable you to come to a similar model choice as if you had used sophisticated GT for evaluation.



Conclusion, Discussion & Future Work

The experiments indicate that an LLM judge can be used to evaluate semantic information retrieval performance in an unsupervised fashion. This includes that the evaluation results can be used as a good proxy to select the embedding model-based retriever that is going to be applied in a specific use case.

From the bar plots and correlations we can conclude that QA is a much better proxy for GT than Q. This might be but is not necessarily related to a slightly more precise relevance classification ability of the judge when an Answer is provided alongside the Question that we could observe as well.

We see a precision in the relevance classification task lower than 75% for both approaches which would not be acceptable in an end-user setting. However, they are sufficient to conduct model comparison, since we expect the noise introduced by the generative model to statistically even out between model evaluations. Consequently, with neither of the approaches we are able to classify relevant passages sufficiently well in such a way that we could safely fall back on an LLM judge instead of ground truth relevance labels in the first place.

Thus, we have to rely on aggregating the relevance judgments using IR metrics and use them to draw conclusions. From the bar charts above, we see that different IR metrics draw a different picture of the comparative performance of the embedding models. Thus, we cannot use one metric alone to tell whether one of the unsupervised approaches can be used as a proxy for the supervised one. With this, we choose correlations of all IR metrics between approaches to give a hint about if one of the approaches could be replaced using ground truth relevance judgments. With a far better Pearson correlation coefficient for QA than for Q we can see that having access to a dataset of question-answer pairs gives an advantage. The correlation figures indicated that using either of the unsupervised approaches results in IR metrics that strongly tend in the direction of the GT metrics. Consequently, both proxy approaches can be considered if for the use case at hand using a heuristic evaluation is sufficient.

Eventually, one has to consider the added benefits and costs for one's use case to compile a Q vs. QA vs. GT dataset.

We should take the presented results with a grain of salt. It should be kept in mind that they stem from a very specific small domain-specific dataset and that only three models were compared. Thus, the previous statements should only be seen as pointers for what might be promising and worth trying.

Moving forward with the presented approaches one should definitely broaden the scope of datasets and embedding models to compare them on for more representative results. Firstly, not necessarily a larger dataset but LLM judgments on more samples from the one presented will be needed. Secondly, a broader range of domain-specific datasets that contain all the presented features has to be compiled as there are really few of them as of now. Such datasets can be contributed and collected here. An added benefit of these datasets is that they can be used to evaluate an end-to-end RAG pipeline as well.

The presented approaches could be improved, focusing on prompt engineering for LLM judges and it would be valuable to learn about the noise they induce. Beyond that, it can be evaluated how mixed approaches influence the LLM judge. In this realm, one could think of few-shot prompting providing one relevant passage alongside the Question and Answer to enhance its judging capabilities. Beyond that, ideas to generate a purely synthetic question-answer and relevance dataset are worth looking at.

The main benefit of using a less sophisticated evaluation dataset and applying an LLM judge is that you will be able to bring your solution to life quicker. All while still being able to make an educated model choice, as we have demonstrated that the comparative retrieval performance of embedding models can be persistent between unsupervised and supervised evaluation approaches. There might be cases where unsupervised approaches can be the only way to yield some form of evaluation, as it is not feasible to create a perfect evaluation dataset at all. Whereas for most environments and scenarios it should be easy to compile one of the simpler datasets that were discussed in this post. Although broader experiments are needed to verify our findings, our results give promising suggestions on how to evaluate and select embedding-based retrievers, especially if you are applying them to niche domain text. Perhaps in your next RAG project.