IJHSR
ijhighschoolresearch.org
RESEARCH ARTICLE

A Comparative Analysis of Embedding Techniques and Their Importance in the Functionality of Large Language Models

Elaine Jiang

Nikola Tesla STEM High School, 4301 228th Ave NE, Redmond, WA 98053; elainejiang1008@gmail.com

ABSTRACT: Embeddings are fundamental to developing the reasoning of a Large Language Model (LLMs). By allowing raw input data to be represented in a vector space, embeddings enable LLMS to detect patterns and excel in a wide range of tasks, such as text classification or detecting semantic similarity. This paper presents a comparative analysis that aims to analyze the various embeddings that are proposed for building RAG models. Drawing from existing studies, we analyze and compare various embedding techniques. Then, their applications are demonstrated through empirical evaluation using the PAN-PC-11 plagiarism detection dataset. Furthermore, using a subset of embeddings in Azure, we will demonstrate how AI applications' performance can vary with embeddings, thus demonstrating the efficacy of embedding models.

KEYWORDS: Robotics and Intelligent Machines, Machine Learning, Embeddings, Large Language Models, Retrieval-Augmented Generation.

Introduction

In the rapidly advancing field of Artificial Intelligence (AI), large language models (LLMs) have emerged as the future of Generative AI applications, containing the capability of understanding and reasoning with language. Over the next decade, these applications will become the new standard for innovation, becoming essential in fields from law to healthcare. Interwoven through the different fields that utilize these LLMs is a necessity for quality data. At the center of data representation lies embeddings; they allow LLMs to perform large-scale, complex downstream tasks by creating vectorized representations of texts.[1] They act as a connection between raw input and machine reasoning, allowing for downstream tasks such as clustering, searching, and retrieval-augmented generation (RAG). RAG systems implement embeddings with other retrieval mechanisms in order to increase accuracy and decrease hallucinations. For example, OpenAI's GPT-4 relies on advanced textual embeddings to understand nuances in meaning and context.[2] Similarly, LLaMA models often employ specialized embedding models to enhance performance on domain-specific tasks, such as legal or biomedical question answering.[3] Multi-modally, the popular image-generator application DALLE-2 improves performance through CLIP embeddings.[4]

Over the past few years, embedding techniques have evolved and gone through significant transformations. Starting with count-based embeddings, Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) methods counted word occurrences or weighted them by rarity, ignoring word meaning.[5] Then, static dense word embeddings such as Word2Vec and GloVe were developed, using fixed vector representations for each word – capturing meaning but unable to adapt to polysemy (or a word changing meaning depending on context).[6,7] Pioneering contextualized embeddings like ELMo, GPT, and BERT captured bidirectional context, allowing for more accurate downstream reasoning.[8-10]

However, embedding models are not perfect. As a consequence, if these representations are not up to standard, the effectiveness of these extremely powerful applications may crash, leading to hallucinations and inaccurate knowledge. This limitation is especially evident in tasks that rely on detecting semantic similarity rather than exact text overlap, such as plagiarism detection, where meaning can be preserved despite substantial surface-level changes.[11] This issue is particularly significant as LLM applications enter high-stakes domains, where misinformation/misinterpretations can have serious consequences. For example, LEGAL-BERT demonstrates the value of embeddings in legal settings for improving document classification and retrieval accuracy, yet similar context-specific approaches remain underexplored in other fields.[12]

This paper investigates the following research question: What are the strengths and limitations of various embedding techniques, and how do they affect the performance of inference using large language models across downstream tasks?

This question is significant because embeddings are important beyond low-stakes use cases such as keyword-based search or topic classification. Now, they are the bridge between knowledge input and foundational model reasoning. High-stakes applications such as plagiarism detection, legal reasoning, and clinical decision support demand robust semantic representations and, subsequently, high-performance accuracy. In this context, this paper argues that the choice of embedding model is not just a technical factor, but a key factor in shaping the accuracy and reliability of RAG-based AI applications.

To explore this claim, this paper is organized into three main sections:

1. Learning Representations: This section surveys existing work on embedding techniques and compares their strengths and limitations for application in RAG frameworks.

DOI: 10.36838/IJHSR87.281

2. Discussion: This section interprets the results of our literature review and highlights the implications of these findings for developers building RAG models, emphasizing case-to-case considerations. Then, it employs an embedding model directly on one such use case, highlighting an application's reliance on embedding quality and discussing the results of the case study.

3. Conclusion: This section summarizes our findings and proposes directions for future research on embedding evaluation.

Learning Representations

In this section, we will present different embedding techniques that are critical and fuel the application development cycle using foundational models. We will divide them into two major categories – linguistic embeddings and multi-modal embeddings. For each family of embeddings, we will present the strengths and weaknesses of each embedding technique while also presenting some technical architectures of popular open source methods.

2.1. Linguistic Embeddings:

Linguistic embeddings transform raw text into numerical vectors that are able to capture semantic meaning, relationships, and context. In turn, linguistic embeddings have become essential for LLMs to perform tasks such as retrieval and clustering. Originating from count-based methods like BoW and TF-IDF, today's contextualized embeddings are built on transformer architectures and are able to adapt representations based on surrounding text. However, while modern models offer richer semantic representations, factors such as context window size or multilingual support still determine embedding model suitability. The following subsections review key linguistic embedding models – E5, SFR-Embedding, GTE, BGE, and Jina – and evaluate their architectures, strengths, and tradeoffs.

2.1.1. E5 Models:

The E5 family of embedding models, developed by Microsoft, is a group of lightweight yet high-performing models for generating semantic representations of text. These models are built on Transformer-based encoder architectures, primarily using RoBERTa or XLM-RoBERTa as the backbone. Natural language instructions are input through a prefix of "query:" or "passage:" for the model to generalize across various tasks. Then, Contrastive Pre-training uses the InfoNCE contrastive loss function.[13,14]

$$L_{cont} = -\frac{1}{n}\sum_i \quad log\ log\ \frac{e^{s\theta(q_i,p_i)}}{e^{s\theta(q_i,p_i)}+\sum_j e^{s\theta(q_i,p_{ij})}}$$

Finally, fine-tuning is applied with curated datasets such as the MS MARCO or NQ for higher performance on semantic similarity, retrieval, and clustering tasks.[15-17] This formulation and finetuning ensure that the model learns fine-grained semantic distinctions between texts, rather than relying on surface-level overlap. These steps allow the models to have improved performance on retrieval and similarity tasks.

The English-only variant, E5-large-v2, is a 350 million parameter embedding model developed by Microsoft. The model, containing 24 layers, is able to capture complex relationships between words and ideas. And, leveraging weakly-supervised contrastive pre-training, it is able to learn semantic information from large amounts of unlabeled text. These features allow the model to deliver a strong, replicable performance in retrieval, semantic search, and textual matching, even surpassing models with up to 40x more parameters on the MTEB benchmark.[14,18] Its lighter size is commercially usable under the MIT license.

However, the embedding model, while highly effective, has certain constraints. Namely, it truncates inputs longer than 512 tokens, and its required prefixes for inputs are more restrictive compared to instruction-friendly alternatives such as the Jina v2 or GTE-Qwen2. Moreover, the E5-large-v2 lacks multilingual capabilities.

To address this, Microsoft released the Multilingual E5 (mE5) models in mid-2023, using a similar training method. The models are trained on about 1 billion multilingual text pairs with contrastive learning and subsequently fine-tuned using labeled datasets.[19] The instruction tune variant, mE5-large-instruct, demonstrates strong performance in the MTEB benchmark, surpassing BGE-large-en-1.5 by 0.2 points.[18,19]

2.1.2. SFR-Embedding:

SFR-Embedding is a 7B parameter model, developed by Salesforce, that produces 4096-dimensional embeddings, designed for the purpose of research.[20] In particular, the SFR-Embedding-Mistral fine-tunes the E5-mistral-7B-instruct across a wide range of tasks with a batch size of 2048. It delivers one of the highest MTEB scores of 67.6, excelling in diverse tasks such as retrieval, clustering, and classification – often matching or outperforming the GTE-Qwen2-7B in tasks centered around English.[21,22] However, its massive size makes it less practical than lighter models such as the E5, BGE, or Jina v2. In addition, the model does not have commercial usability due to its research-only license. While the E5 offers a balance in performance and versatility, the SFR needs large amounts of GPU resources while lacking nuanced multilingual capabilities, which the GTE excels at. In addition, the SFR is less nuanced than the E5 and is unable to understand context as well.

2.1.3. GTE Models:

The General Text Embedding (GTE) models, developed by Alibaba DAMO Academy, employ a dual-encoder architecture built on top of deep Transformer encoder architectures, like BERT. Given a text input x = (x1, . . . , xn), the encoder first contextualizes token embeddings:[2]

$$h = LM(x) \in R^{n\times d}$$

Then, a simple mean pooling is applied over the token dimension to obtain a text representation:[2]

$$x = \frac{1}{n}\sum_{i=1}^{n} \quad h_i \in R^d$$

Similar to E5 models, GTE is trained using a contrastive loss objective with the InfoNCE loss function.[13] Then, the

training follows a two-stage training process: 1. large-scale unsupervised pre-training on ~800 million naturally occurring text pairs and 2. supervised fine-tuning on smaller, curated datasets.[2] The pre-training hones the model's ability to learn general semantic relationships and to detect nuanced semantic similarities relevant to downstream tasks.

GTE supports a 32,000-token input window, far exceeding E5's 512-token limit and Jina's 8192-token limit, making it ideal for long-context tasks. While previous GTE models contained multilingual restraints, the recently released GTE-Qwen2 models show excellent performance across languages.[23] As of June 16, 2024, the GTE-Qwen2-7B-Instruct achieved number one on the MTEB leaderboards for both Chinese and English tasks.[18] In comparison to the SFR, GTE stands out for its excellent multilingual performance and instruction-following ability. However, both have similarly high computational demands – making them impractical in low-resource environments.

2.1.4. BGE Models:

The BGE family was developed by the Beijing Academy of Artificial Intelligence and are text embedding model optimized for dense retrieval, reranking, and semantic similarity. The architecture of BGE models is based on Transformer encoders and is often initialized from pretrained models such as Mistral-7B.[24,25] They are trained using contrastive learning objectives and Low-Rank Adaptation (LoRA), for parameter-efficient fine-tuning.[25,26] They are trained on a mix of retrieval, reranking, classification, clustering, and semantic textual similarity (STS) tasks across datasets.[25] BGE models span a range of sizes, such as the bge-small, bge-base, and bge-large, and are multilingual. In addition, these models support context windows of up to 8192 tokens and demonstrate a strong performance on the MTEB and C-MTEB benchmarks, outperforming existing Chinese text embedding models by more than 10% as of September 24, 2024, while having relatively lightweight architectures.[27] However, with extremely long or unstructured inputs, embedding performance drops significantly. In addition, procedures to fine-tune the models may be technically complex.

2.1.5. Jina-Embeddings:

Jina-Embeddings: The Jina Embeddings family, developed by Jina AI, is designed to address semantic tasks and their numerous versions geared towards varying tasks. Among these, Jina Embeddings v2 stands out as a lightweight yet high-performing model built on the BERT architecture and fine-tuned using text pairs and hard negatives.[28] Jina Embeddings v2 featured ~100 million parameters and was optimized to be able to run on consumer-grade hardware while rivaling OpenAI's ada-002 on tasks such as classification and reranking, as evaluated on the MTEB.[18,28] The model supports context lengths up to 8192 tokens, allowing better handling of longer documents, transcending the limit of the BERT architecture, and faster speeds due to an optimized parameter size. However, Jina v2 is not without limitations. Its relatively small size may constrain the depth of semantic nuance it can capture, and its large context-window may consequently dilute attention over extended passages.

■ Methodology

For this study, we selected the PAN-PC-11 dataset.[11] This publicly available dataset offers an extensive and comprehensive collection of both original or "source" documents and their corresponding "suspicious" counterparts, some of which are confirmed to be plagiarized directly or through semantic obfuscation. Out of all the datasets researched, PAN-PC-11 stood out for its size, accessibility, and rich metadata, which includes plagiarism labels, source document references, degrees of obfuscation (low or high), and origin type (artificial or manual). Specifically, we narrowed down our data to the Source and Suspicious Documents within the "external" folder contained in Part 1 of the dataset, which provided the thorough metadata. Each folder contains 23 parts or subfolders of 500 documents, with no particular distinction between each part. Currently, we are encoding the first part (or first 500 documents) from the source, and the same for the suspicious documents. For this initial empirical investigation and due to resource constraints, including embedding API costs and runtime, we limited our analysis to the first 100 documents from each category (source and suspicious), all drawn from Part 1 of the dataset.

To demonstrate the importance of embeddings in detecting semantic similarity, we designed a pipeline that evaluates overlap between original source documents and potentially plagiarized counterparts using embedding-based retrieval. The ultimate goal wasn't to measure or benchmark specific embedding models, but to show how the application's overall success depends on the embeddings it operates on.

2.2. Multi-Modal Embeddings:

In the field of multi-modal large language models, encoders play a central role in converting raw information, often in the form of images, audio, or video, into a compact format that can be understood. Rather than training these encoders from scratch, many multi-modal LLMs employ pretrained encoders that have already been aligned. Because these encoders significantly impact downstream LLM performance, it is necessary to understand the strengths and limitations of each encoding technique and select an encoder that is well-suited for integration with the LLM.[29] The following subsections will evaluate the performance of key multi-modal embeddings, including CLIP-ViT-L/14, EVA-CLIP-VIT-G/14, Open-CLIP-ViT-bigG/14, HuBERT, and CLAP.

2.2.1. CLIP-ViT-L/14:

CLIP-ViT-L/14 is a widely used vision encoder that has been pretrained on a dataset of approximately 400 million text pairs. Because of this alignment, zero-shot performance across a wide range of downstream tasks is enabled. Because CLIP is widely used and has a strong initial alignment, it is often favored as the base encoder for many LLMs. However, the CLIP operates at relatively low resolutions (224/336), which can hinder performance on downstream tasks, as previous

DOI: 10.36838/IJHSR87.281

studies cite an improvement in performance with higher resolution.[29-33] Moreover, an analysis of CLIP's performance across a variety of different datasets found that it still falls well below the overall state of the art, and that its zero-shot performance is inadequate on several types of tasks.[34] Compared to EVA-CLIP-ViT-G/14 and Open-CLIP-ViT-bigG/14, CLIP is smaller and more computationally efficient, making it suitable for resource-constrained environments, though at the cost of lower maximum accuracy.

2.2.1. EVA-CLIP-ViT-G/14:

Used in models like MiniGPT-4, EVA-CLIP-ViT-G/14 is a ViT-based encoder enhanced through improved pretraining strategies.[35] With more than 1 billion parameters, the encoder offers more nuanced representations of complex visual inputs. In addition, some EVA models demonstrate higher zero-shot top-1 accuracy than CLIP models while having lower training costs than Open-CLIP.[36] However, the EVA-CLIP-ViT-G/14 encoder contains lower resolution (224), which may affect the performance of downstream tasks.[29] EVA offers a middle ground: more nuanced than CLIP but more resource-efficient than Open-CLIP.

2.2.2. Open-CLIP-ViT-bigG/14:

Open-CLIP-ViT-bigG/14 is trained on the LAION-2B dataset of almost 2 billion image-text pairs and features a significantly larger parameter size of 1.8B +, allowing it to capture high-level semantic features effectively.[37] However, the encoder's large parameter size makes it consequently extremely memory and resource-intensive, making it less practical for certain applications. Compared to CLIP and EVA, it offers the highest maximum accuracy but the lowest efficiency.

2.2.3. HuBERT:

HuBERT is a self-supervised audio encoder designed to address three challenges in speech representation learning: the presence of multiple units per unit of input utterance, the lack of a predefined lexicon, and the changing lengths of audio segments.[38] Rather than relying on high-quality cluster labels, HuBERT starts with basic k-means clustering and refines over multiple iterations. In other words, it depends on the quality of the clustering process itself. With a 1 billion parameter model, HuBERT outperforms or matches wav2vec 2.0 across benchmarks like LibriSpeech and Libri-Light, along with achieving up to 19% relative WER reduction on difficult datasets.[38] However, even with HuBERT's performance, due to its reliance on cluster consistency, poor initial clustering can still degrade downstream performance. In addition, HuBERT has a speech-only focus, not supporting other audio modalities such as background sounds or music.

2.2.3. CLAP:

CLAP applies contrastive learning to align audio and natural language into a shared representation space.[39] It is trained on the LAION-Audio-630K dataset, including over 630,000 audio-text pairs, allowing it to connect these signals to textual descriptions. CLAP achieves state-of-the-art performance in zero-shot audio classification and demonstrates excellent performance in text-to-audio retrieval, while also performing competitively in fully supervised settings.[39] However, CLAP relies on the quality of text captions, allowing low-quality descriptions to degrade model performance. In addition, CLAP's contrastive learning stage is resource-consuming and requires substantial memory.

■ Discussion

3.1. Analysis:

In this section, we synthesize insights from our analysis of embedding techniques and our empirical evaluation on plagiarism detection. While embedding techniques are often compared on the basis of standardized benchmarks, we employ an approach that captures the nuances of real-world applicability, taking into account system-level constraints or ethical consequences. By contrasting low-stakes applications such as homework assistance tools with high-stakes applications such as clinical diagnostics, we highlight how model capabilities must be evaluated in context. In addition, our empirical case study demonstrates how embedding quality directly affects downstream performance on both a technical level and a moral or societal one. Overall, these findings underscore the importance of holistically selecting an embedding model with consideration of both the technical demands and ethical stakes of the application.

Table 1: This table lists a variety of embedding models, covering text, vision, and audio modalities, along with their specifications.

Model	Modality	Parameters	Context Limit	Multilingual	Licensing
E5-large-v2	Text	350 million	512 tokens	No	MIT
mE5-large-instruct	Text	350 million	512 tokens	Yes	MIT
SFR-Mistral	Text	7 billion	8192 tokens	Partial	Research
GTE-Qwen2-7B	Text	7 billion	32,000 tokens	Yes	Apache
BGE Models	Text	Varies	8192 tokens	Yes	Apache
Jina v2	Text	~100 million	8192 tokens	Partial	Commercial
CLIP-ViT-L/14	Vision	304 million	224/336 Resolution	No	Open
EVA-CLIP-ViT-G/14	Vision	1 billion	224 Resolution	No	Open
Open-CLIP-ViT-G/14	Vision	1.8+ Billion	224 Resolution	No	Open
HuBERT	Speech	1 billion	N/A	No	Open
CLAP	Audio	~600 million	N/A	No	Open

Choosing an embedding model is often determined by the constraints and goals of the application. We present two use cases of embedding models on opposite ends of the spectrum and explain how different models would suit one or the other.

In the first case, consider the development of a lightweight homework assistance app that helps high school students with homework through summarizing texts or explaining certain concepts. A requirement for this system is usability on devices with limited processing power and accessibility with minimal latency. Therefore, in this scenario, lightweight embedding models such as Jina Embeddings v2 or E5-large-v2 are well-suited. On the other hand, models based solely on performance, such as the GTE-Qwen2-7B or SFR-Mistral, may

offer higher accuracy or semantic understanding for homework summaries or explanations but are significantly heavier in terms of computing and memory requirements. These models would be ill-suited for lightweight applications due to their size and longer response times, which can degrade usability for students. In addition, designing this app for commercial purposes may not allow for access to many of these embedding techniques, as certain models like SFR-Mistral are restricted to research use only (Table 1). Therefore, model selection must include not only performance goals, but also the system's constraints and user context.

On the other side of the spectrum, consider a high-stakes medical diagnostics system supporting oncologists, which must detect subtle patterns in X-rays while considering patient histories. Here, visual encoders like Open-CLIP-ViT-G/14 or EVA-CLIP-ViT-G/14 are better suited for these scenarios, offering better alignment of textual and visual information while also allowing for a more fine-grained representation of medical imagery. For the text-processing component, models like GTE-Qwen2-7B, though computationally taxing, can provide long-context reasoning and instruction-following ability, which are well-suited to synthesize long medical records or research literature. In contrast, lighter models like Jina v2 or E5-large-v2 are insufficient in this context because they contain limited context windows and lower representational capacity (Table 1), meaning they are at a higher risk of losing critical clinical details. In medical systems where misclassification or inaccuracy may have life-threatening consequences, performance trade-offs are unacceptable.

3.2. Empirical Evaluation:

In the era of large language models, embeddings have become the basis for downstream applications. They compress text into a summary of its meaning and structure, which allows systems to make fast and efficient judgments about language-based tasks. However, this reliance on embeddings introduces a critical assumption that the embedding contains enough information to support the downstream task. If these embeddings are too shallow or insensitive to deeper meaning, then many downstream models could likely fail. This empirical study investigates that statement by isolating and evaluating the role of embeddings in a concrete, high-stakes application – plagiarism detection.

Plagiarism, especially in the era of generative AI, is more than just keyword matching. Instead, it asks whether a model can understand when two texts express the same ideas in different forms. This makes it a compelling lens for exploring how LLM systems rely on embeddings. The task comes in two main levels: surface-level plagiarism, where copied content is nearly identical, and semantic plagiarism, where the idea is stolen but reworded to avoid detection. Surface-level plagiarism can often be caught through n-gram comparisons or cosine similarity of token vectors. But detecting semantic plagiarism, especially with high obfuscation, requires embedding techniques. Most modern applications built on LLMs don't run on raw input each time. Instead, they run on fixed embeddings to make judgments. Therefore, the embedding is not just important internally, but can become a single, critical point of failure. If an embedding fails to capture nuance between two texts, downstream performance may be significantly degraded. With the plagiarism detection case study, this dependency becomes visible – when the embeddings successfully capture semantic similarity, the model succeeds; and when they don't, it fails.

Moreover, plagiarism detection demonstrates the ethical stakes of embeddings. Academic integrity and intellectual property protection are foundational principles that reach across the fields of education and innovation, meaning they must be enforced. If an application cannot reliably flag plagiarized content, these principles may be broken. A model needs to understand that meaning itself can be stolen. And embeddings are a key part in aiding that ability. Therefore, this use case evaluates the actual role of embeddings in high-stakes applications that need thorough, complex language understanding.

3.2.1. Methodology:

For this study, we selected the PAN-PC-11 dataset.[11] This publicly available dataset offers an extensive and comprehensive collection of both original or "source" documents and their corresponding "suspicious" counterparts, some of which are confirmed to be plagiarized directly or through semantic obfuscation. Out of all the datasets researched, PAN-PC-11 stood out for its size, accessibility, and rich metadata, which includes plagiarism labels, source document references, degrees of obfuscation (low or high), and origin type (artificial or manual). Specifically, we narrowed down our data to the Source and Suspicious Documents within the "external" folder contained in Part 1 of the dataset, which provided the thorough metadata. Each folder contains 23 parts or subfolders of 500 documents, with no particular distinction between each part. Currently, we are encoding the first part (or first 500 documents) from the source, and the same for the suspicious documents. For this initial empirical investigation and due to resource constraints, including embedding API costs and runtime, we limited our analysis to the first 100 documents from each category (source and suspicious), all drawn from Part 1 of the dataset.

To demonstrate the importance of embeddings in detecting semantic similarity, we designed a pipeline that evaluates overlap between original source documents and potentially plagiarized counterparts using embedding-based retrieval. The ultimate goal wasn't to measure or benchmark specific embedding models, but to show how the application's overall success depends on the embeddings it operates on.

DOI: 10.36838/IJHSR87.281

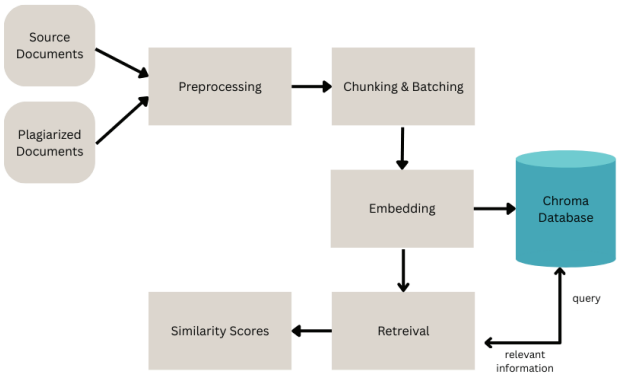

Figure 1: This figure details the pipeline that was designed when employing embedding models and receiving similarity scores for the plagiarized and source documents.

We designed a preprocessing and embedding-based retrieval pipeline to detect plagiarism by measuring semantic similarity between documents.

Preprocessing (Figure 1): In the first step of the pipeline, all source and candidate documents undergo a uniform preprocessing stage to ensure consistency in downstream analysis. Text is converted to lowercase, punctuation and stop words are removed, whitespace is normalized (e.g., replacing newlines with spaces), and excess spacing is trimmed. This cleaning stage reduces noise and enforces a standardized text representation so that irrelevant formatting differences are prevented from affecting similarity scores.

Chunking and Batching (Figure 1): After preprocessing the data, the RecursiveCharacterTextSplitter is employed to split each document into semantically meaningful chunks. Unlike naive splitting, which may fragment meaning, recursive splitting preserves coherence within each chunk, which is necessary for meaningful embeddings. Then, the resulting chunks are batched in order to stay under the total token length limit of Azure OpenAI embedding API (text-embedding-3-large) to prevent truncation or errors. Batching size was set to 20 chunks per API call, and retries were configured for up to 3 attempts with a 10-second delay.

Embedding (Figure 1): Each batch is transformed into a 1536-dimensional vector using the pretrained Azure OpenAI embedding model text-embedding-3-large. These dense embeddings are able to capture semantic meaning and enable vector similarity comparisons.

Storage in Vector Database (Figure 1): The resulting vectors are stored in ChromaDB under the corresponding source document, which enables efficient similarity search at scale and preserves metadata. With ChromaDB's metadata handling, we can identify not only which document it came from but also where in the document it stemmed from. Because documents vary in length, a single document may correspond to multiple embeddings. This vector database acts as the retrieval backend for future similarity queries.

Retrieval (Figure 1): Then, each batch of embeddings for the candidate documents is queried against the ChromaDB using similarity search. Cosine similarity is employed as the similarity metric, as it normalizes for vector magnitude and is widely used in high-dimensional embedding spaces. For each batch, the top five nearest neighbors are retrieved, and the neighbor with the highest similarity is selected as the closest match.

Similarity Scores (Figure 1): To summarize document-level similarity, both the maximum and average similarity scores are computed across all batches. These metrics address two different levels of plagiarism: the maximum similarity score capturing potential moments of direct copying, while the average reflects the overall semantic similarity between the texts. This process is repeated for both genuine source documents and known plagiarized texts.

This allows us to capture both local overlaps and broader semantic similarity, providing a nuanced view of potential plagiarism.

3.2.2. Results:

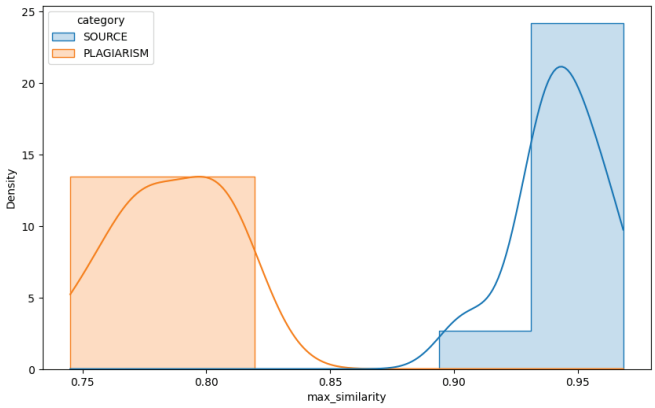

Figure 2: This figure shows the distribution of maximum similarity scores between two different types of documents – source documents, shown in blue, and plagiarized documents, shown in orange. The minimal overlap between the two distributions indicates strong discriminative power between the categories.

After embedding and comparing each from the source and suspicious, or potentially plagiarized, documents, our empirical evaluation revealed a clear distinction in the maximum similarity scores between the original source documents and the known plagiarized texts, demonstrating the effectiveness of embeddings in plagiarism detection. Source documents, when compared to themselves, produced maximum similarity scores tightly clustered between 0.90 and 0.97, with a peak around 0.94 (Figure 2). On the other hand, plagiarized documents, even with obfuscation, demonstrated consistently lower maximum similarity scores, clustering between 0.74 and 0.83, while peaking near 0.80 (Figure 2). This non-overlapping bimodal distribution suggests that, even with heavy obfuscation or paraphrasing, textual embeddings preserved enough meaning in their vector representations to be able to distinguish plagiarized texts from source materials. However, even with a clear distinction between plagiarized and source materials, the embedding model still produced semantic similarity scores that were detectably high for copied content. These results demonstrate that embedding quality directly influences the downstream performance of high-stakes language systems such as plagiarism detection. When embeddings are able to encode nuanced semantic meanings, these tasks demonstrate higher accuracy. Our findings support the idea that embed-

dings deserve explicit attention when designing applications, especially with Generative AI.

3.2.3. Limitations:

While this study highlights the role of embeddings in detecting plagiarism, its findings are constrained to the dataset and models evaluated. Computational constraints restricted the sample size, which may reduce the generalizability and robustness of the results. The evaluation was also limited to text-based plagiarism, leaving out other modalities such as code or multimedia content.

■ Conclusion

In conclusion, this study has explored different types of embedding models – linguistic and multi-modal – and demonstrated that they are components of an application that can significantly influence the ethical and functional performance of AI applications. Benchmarks like MTEB can offer surface-level comparisons, but our analysis shows that embedding effectiveness is deeply tied to application-specific constraints and consequences. We examined two theoretical use cases – lightweight educational tools and high-stakes medical diagnostics – and conducted an empirical investigation into plagiarism detection. With our empirical evaluation, source documents, when compared to themselves, produced maximum similarity scores between 0.90 and 0.97, while plagiarized documents, even with high obfuscation, clustered around 0.74 and 0.83, demonstrating high overlap but a clear distinction between original and copied. Our results, with a non-overlapping bimodal distribution, suggest that high-quality embeddings are essential for nuanced tasks such as identifying plagiarized documents, a task that requires complex representations that capture semantic meaning. Our analysis of different embedding models aims to provide a clear rationale for selecting the most appropriate model based on the specific constraints of different applications, and our domain-specific empirical case study aims to emphasize the significance of embedding techniques in various contexts. Moreover, our findings explore the importance of context-based analysis and the potential significant societal impact. As embeddings become foundational across Generative AI applications, their failure can translate into failures in high-stakes environments, like preserving academic integrity or medical accuracy. For example, in the medical diagnostics context, a poorly chosen embedding model could mean hallucinations and inaccuracies in clinical recommendations, potentially affecting patient vitality.

Moving forward, future research is needed in several different avenues. More systematic comparisons across a wider range of embedding models are needed – especially evaluating how they perform in different contexts, such as education, healthcare, and law. Moreover, as linguistic and multimodal models continue to evolve, deeper exploration can be done to evaluate how these embeddings generalize across languages or modalities. Additionally, future work could investigate the robustness of embedding models to adversarial inputs and integrate more complex metrics into embedding evaluation frameworks.

Ultimately, our work emphasizes the notion that embedding models are not just a minor technical component. Instead, they play a foundational role in shaping how information is represented and utilized in modern Generative AI systems. Careful, context-based selection of embeddings is therefore essential in order to build responsible, high-performing AI systems. Because as AI becomes central to high-stakes decision-making, the importance of high-quality embeddings has never been greater.

■ Acknowledgments

I would like to express my gratitude to Dr. Siddharth Krishnan and Lian Jiang for their guidance, unwavering support, and dedicated time throughout the lifecycle of the project and this research paper.

■ References

1. Li Z, Zhang X, Zhang Y, Long D, Xie P, Zhang M. Towards General Text Embeddings with Multi-stage Contrastive Learning [Internet]. arXiv; 2023 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2308.03281
2. OpenAI, Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. GPT-4 Technical Report [Internet]. arXiv; 2024 [cited 2025 Aug 9]. Available from: http://arxiv.org/abs/2303.08774
3. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and Efficient Foundation Language Models [Internet]. arXiv; 2023 [cited 2025 Aug 9]. Available from: http://arxiv.org/abs/2302.13971
4. Gozalo-Brizuela R, Garrido-Merchan EC. ChatGPT is not all you need. A State of the Art Review of large Generative AI models [Internet]. arXiv; 2023 [cited 2025 Aug 13]. Available from: http://arxiv.org/abs/2301.04655
5. Cao H. Recent advances in text embedding: A Comprehensive Review of Top-Performing Methods on the MTEB Benchmark [Internet]. arXiv; 2024 [cited 2025 Aug 13]. Available from: http://arxiv.org/abs/2406.01607
6. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space [Internet]. arXiv; 2013 [cited 2025 Aug 13]. Available from: http://arxiv.org/abs/1301.3781
7. Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation. In: Moschitti A, Pang B, Daelemans W, editors. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) [Internet]. Doha, Qatar: Association for Computational Linguistics; 2014 [cited 2025 Aug 13]. p. 1532–43. Available from: https://aclanthology.org/D14-1162/
8. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep contextualized word representations [Internet]. arXiv; 2018 [cited 2025 Aug 13]. Available from: http://arxiv.org/abs/1802.05365
9. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving Language Understanding by Generative Pre-Training.
10. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein J, Doran C, Solorio T, editors. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) [Internet]. Minneapolis, Minnesota: Association for Computational Linguistics; 2019 [cited

DOI: 10.36838/IJHSR87.281

2025 Aug 13]. p. 4171–86. Available from: https://aclanthology.org/N19-1423/
11. Potthast M, Stein B, Eiselt A, Barrón-Cedeño A, Rosso P. PAN Plagiarism Corpus 2011 (PAN-PC-11) [Internet]. Zenodo; 2011 [cited 2025 Aug 2]. Available from: https://zenodo.org/records/3250095
12. Chalkidis I, Fergadiotis M, Malakasiotis P, Aletras N, Androutsopoulos I. LEGAL-BERT: The Muppets straight out of Law School [Internet]. arXiv; 2020 [cited 2025 Aug 13]. Available from: http://arxiv.org/abs/2010.02559
13. Chen T, Kornblith S, Norouzi M, Hinton G. A Simple Framework for Contrastive Learning of Visual Representations. In: Proceedings of the 37th International Conference on Machine Learning [Internet]. PMLR; 2020 [cited 2025 July 20]. p. 1597–607. Available from: https://proceedings.mlr.press/v119/chen20j.html
14. Wang L, Yang N, Huang X, Jiao B, Yang L, Jiang D, et al. Text Embeddings by Weakly-Supervised Contrastive Pre-training [Internet]. arXiv; 2024 [cited 2025 July 19]. Available from: http://arxiv.org/abs/2212.03533
15. Bajaj P, Campos D, Craswell N, Deng L, Gao J, Liu X, et al. MS MARCO: A Human Generated MAchine Reading Comprehension Dataset [Internet]. arXiv; 2018 [cited 2025 July 20]. Available from: http://arxiv.org/abs/1611.09268
16. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense Passage Retrieval for Open-Domain Question Answering. In: Webber B, Cohn T, He Y, Liu Y, editors. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) [Internet]. Online: Association for Computational Linguistics; 2020 [cited 2025 July 20]. p. 6769–81. Available from: https://aclanthology.org/2020.emnlp-main.550/
17. Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A, Alberti C, et al. Natural Questions: A Benchmark for Question Answering Research. Lee L, Johnson M, Roark B, Nenkova A, editors. Trans Assoc Comput Linguist. 2019;7:452–66.
18. Muennighoff N, Tazi N, Magne L, Reimers N. MTEB: Massive Text Embedding Benchmark. In: Vlachos A, Augenstein I, editors. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics [Internet]. Dubrovnik, Croatia: Association for Computational Linguistics; 2023 [cited 2025 July 20]. p. 2014–37. Available from: https://aclanthology.org/2023.eacl-main.148/
19. Wang L, Yang N, Huang X, Yang L, Majumder R, Wei F. Multilingual E5 Text Embeddings: A Technical Report [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2402.05672
20. Yavuz RM Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, Semih. SFR-Embedding-Mistral: Enhance Text Retrieval with Transfer Learning [Internet]. Salesforce. 2024 [cited 2025 July 20]. Available from: https://www.salesforce.com/blog/sfr-embedding/
21. Caspari L, Dastidar KG, Zerhoudi S, Mitrovic J, Granitzer M. Beyond Benchmarks: Evaluating Embedding Model Similarity for Retrieval Augmented Generation Systems [Internet]. arXiv; 2024 [cited 2025 July 19]. Available from: http://arxiv.org/abs/2407.08275
22. Tao C, Shen T, Gao S, Zhang J, Li Z, Tao Z, et al. LLMs are Also Effective Embedding Models: An In-depth Overview [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2412.12591
23. Yang A, Yang B, Hui B, Zheng B, Yu B, Zhou C, et al. Qwen2 Technical Report [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2407.10671
24. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D de las, et al. Mistral 7B [Internet]. arXiv; 2023 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2310.06825
25. Li C, Qin M, Xiao S, Chen J, Luo K, Shao Y, et al. Making Text Embedders Few-Shot Learners [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2409.15700
26. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-Rank Adaptation of Large Language Models [Internet]. arXiv; 2021 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2106.09685
27. Xiao S, Liu Z, Zhang P, Muennighoff N, Lian D, Nie JY. C-Pack: Packed Resources For General Chinese Embeddings [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2309.07597
28. Günther M, Ong J, Mohr I, Abdessalem A, Abel T, Akram MK, et al. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents [Internet]. arXiv; 2024 [cited 2025 July 20]. Available from: http://arxiv.org/abs/2310.19923
29. Yin S, Fu C, Zhao S, Li K, Sun X, Xu T, et al. A Survey on Multimodal Large Language Models. Natl Sci Rev [Internet]. 2024 Nov 14 [cited 2025 July 23];11(12). Available from: http://arxiv.org/abs/2306.13549
30. Bai J, Bai S, Yang S, Wang S, Tan S, Wang P, et al. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond [Internet]. arXiv; 2023 [cited 2025 July 23]. Available from: http://arxiv.org/abs/2308.12966
31. Li Z, Yang B, Liu Q, Ma Z, Zhang S, Yang J, et al. Monkey: Image Resolution and Text Label Are Important Things for Large Multi-modal Models [Internet]. arXiv; 2024 [cited 2025 July 23]. Available from: http://arxiv.org/abs/2311.06607
32. Liu H, Li C, Li Y, Lee YJ. Improved Baselines with Visual Instruction Tuning [Internet]. arXiv; 2024 [cited 2025 July 23]. Available from: http://arxiv.org/abs/2310.03744
33. McKinzie B, Gan Z, Fauconnier JP, Dodge S, Zhang B, Dufter P, et al. MM1: Methods, Analysis & Insights from Multimodal LLM Pre-training [Internet]. arXiv; 2024 [cited 2025 July 23]. Available from: http://arxiv.org/abs/2403.09611
34. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning Transferable Visual Models From Natural Language Supervision [Internet]. arXiv; 2021 [cited 2025 July 23]. Available from: http://arxiv.org/abs/2103.00020
35. Zhu D, Chen J, Shen X, Li X, Elhoseiny M. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models [Internet]. arXiv; 2023 [cited 2025 July 25]. Available from: http://arxiv.org/abs/2304.10592
36. Sun Q, Fang Y, Wu L, Wang X, Cao Y. EVA-CLIP: Improved Training Techniques for CLIP at Scale [Internet]. arXiv; 2023 [cited 2025 July 25]. Available from: http://arxiv.org/abs/2303.15389
37. Cherti M, Beaumont R, Wightman R, Wortsman M, Ilharco G, Gordon C, et al. Reproducible scaling laws for contrastive language-image learning. In 2023 [cited 2025 July 25]. p. 2818–29. Available from: http://arxiv.org/abs/2212.07143
38. Hsu WN, Bolte B, Tsai YHH, Lakhotia K, Salakhutdinov R, Mohamed A. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units [Internet]. arXiv; 2021 [cited 2025 July 26]. Available from: http://arxiv.org/abs/2106.07447
39. Wu Y, Chen K, Zhang T, Hui Y, Nezhurina M, Berg-Kirkpatrick T, et al. Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation [Internet]. arXiv; 2024 [cited 2025 July 26]. Available from: http://arxiv.org/abs/2211.06687

Authors

Elaine Jiang is a 12th grader at Nikola Tesla STEM High School in Washington with a strong passion for machine learning and Generative AI. She is particularly intrigued by the exploration of how artificial intelligence and technology can be harnessed to address real-world problems.

DOI: 10.36838/IJHSR87.281