

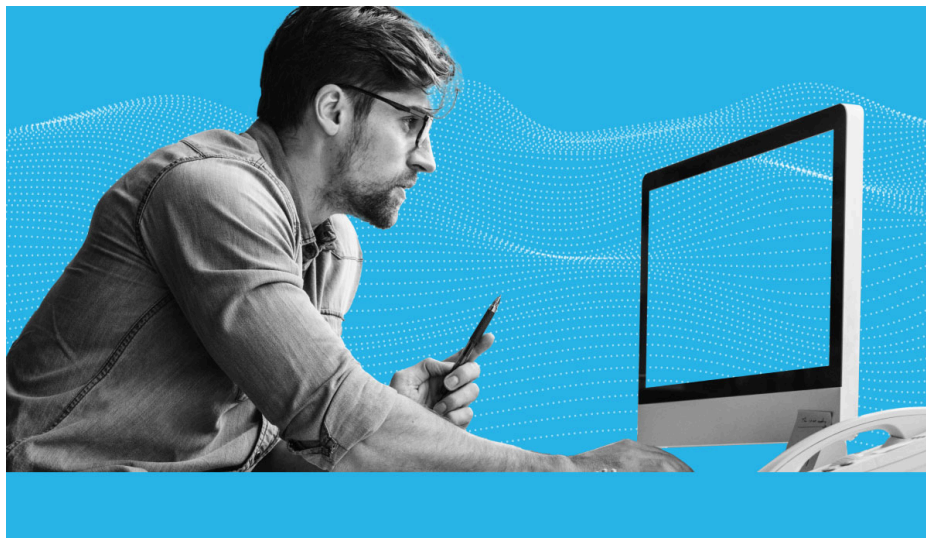
JUN 17, 2024

AUTHOR

Snowflake AI Research

# Snowflake Launches the World's Best Practical Text-Embedding Model for Retrieval Use Cases

SHARE



Today Snowflake is launching and open-sourcing with an Apache 2.0 license the Snowflake Arctic embed family of models. Based on the Massive Text Embedding Benchmark (MTEB) Retrieval Leaderboard, the largest Arctic embed model with only 334 million parameters is the only one to surpass average retrieval performance of 55.9, a feat only less practical to deploy models with over 1 billion parameters are able to achieve. The family of five models are available on [Hugging Face](#) for immediate use and in Snowflake Cortex embed function (in private preview), available soon. The models, which deliver leading retrieval performance, give organizations a new edge when combining proprietary datasets with LLMs as part of a Retrieval Augmented Generation (RAG) or semantic search service. These impressive embedding models directly implement the technical expertise, proprietary

search knowledge, and research and development that **Snowflake acquired last May via Neeva.**

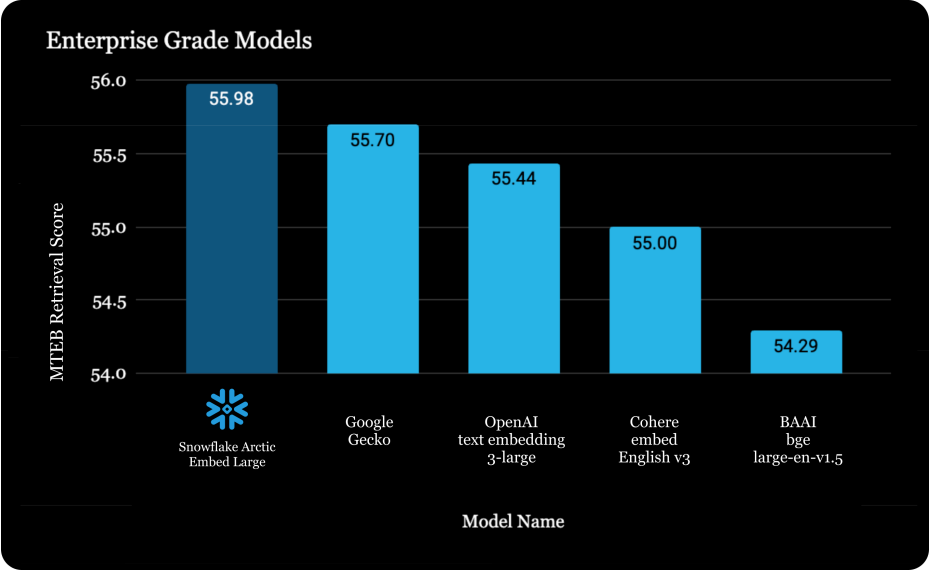
Highlights from this family of embedding models include:

A suite of models is available in five sizes ranging from x-small (xs) to large (l), which deliver state-of-the-art retrieval performance on the Massive Text Embedding Benchmark (MTEB) retrieval benchmark.

The large (l) model, which stands at 334 million parameters, can beat the performance of closed-source models estimated to be roughly 4x the size, such as those available via OpenAI and Cohere text embedding APIs.

The medium (m) sized model includes a long-context version that supports long document retrieval with extended context support of up to 8192 tokens.

Compared to embedding models of comparable retrieval quality, the generally smaller size of each embedding model gives organizations an edge in reducing latency and lowering the total cost of ownership (TCO).







**Scalable, accurate and efficient enterprise search**

AUTHOR

Snowflake AI Research

Embedding models are a crucial component of most modern AI workloads. From powering search to empowering RAG agents with proprietary information, the ability to find the most relevant content is a foundation of AI systems. As part of Snowflake's commitment to AI excellence, we sought to deeply understand text embedding models to deliver the best experience for customer products leveraging Snowflake for their search needs. Leveraging our rich expertise in search and the state-of-the-art research in this space, we set out to create the best open-source text embedding models from the ground up.

SHARE

Starting with our understanding of what is needed to make search     ining it with state-of-the-art research led us to rebuild text embedding from the ground up. As we alluded to in our recent discussion on [how to train a state-of-the-art embedding model](#) and [how to use Snowflake to process training data](#), we analyzed text embedding models from first principles and then went ahead and trained these models end to end in Snowflake. This tooling and our deep understanding of search allowed us to create this suite of models, which outperform previous state-of-the-art models across all our embedding variants. Simply stated, our models are unmatched for their quality and TCO for enterprises of any size seeking to power their embedding workflows.

## Evaluating model quality

Evaluating the quality of retrieval systems in a nonproprietary way can be difficult; building on decades of academic research, the Massive Text Embedding Benchmark (MTEB) has emerged as a standard benchmark. It is a collection of tasks that measures the performance of retrieval systems across seven tasks: classification, clustering, pair classification, re-ranking, retrieval, STS (semantic textual similarity), and summarization. It includes 56 datasets from various domains and with various text lengths. The Snowflake Arctic embed models firmly focus on empowering real-world retrieval workloads, and as a result, we focus on the retrieval portion of MTEB.

As of April 2024, each of our models is ranked first among embedding models of similar size, and our largest model is only outperformed by open-source models with over 20 times (and

four times for closed models) the number of parameters or closed-source models that do not disclose any form of model characteristics. We understand that testing against a single benchmark can both understate and overstate the impact of variation of embedding models on customer workloads, which is what Snowflake ultimately cares about. This is why we are working on new benchmarks focusing on real-world use cases with real-world datasets in the next benchmarking phase. We will update the broader community when we have a large enough sample.

AUTHOR

Snowflake AI Research

The models

SHARE



snowflake’s arctic-embed models are a family of 5 text embedding models that range in context window and size (number of parameters). The model sizes range from 23 to 334 million parameters, and one of the models has an extended context window to give enterprises a full range of options that best match their latency, cost, and retrieval performance requirements. In the table below, we cover model sizes and their relative performance improvement compared to the prior best-performing model with a similar size. The metric for quality is NDCG@10 on the [MTEB Retrieval leaderboard](#), and we compare each model’s performance with associated open-source models with similar parameter count.

Model Name	Model Size (Millions of Parameters)	Embedding Dimensions	Context Window	Average MTEB Retrieval Performance	Prior SoTA model	Improvement over Prior SoTA
snowflake-arctic-embed-xs	23	384	512	50.15	GIST-all-MiniLM-L6-v2	+5.03
snowflake-arctic-embed-s	33	384	512	51.98	bge-small-en-v1.5	+0.30
snowflake-arctic-embed-m	110	768	512	54.90	bge-base-en-v1.5	+1.65
snowflake-arctic-embed-m-long	137	768	8192	54.83	nommic-embed-text-v1	+1.82
snowflake-arctic-embed-l	334	1024	512	55.98	bge-large-en-v1.5	+1.32

As of April 16, 2024, snowflake-arctic-embed-l is the most capable open-source model that can be used in production based on its performance-to-size ratio. Models that outperform snowflake-arctic-embed-l, such as SFR-Embedding-Mistral, widely available among embedding model providers, have a vector dimensionality four times larger (1024 vs. 4096) and have over 20x more parameters (334 million vs. 7.1 billion). With the Apache 2 licensed Snowflake Arctic embed family of models, organizations now have one more open alternative to black-box API providers such as Cohere, OpenAI, or Google.

As shown in the table below, snowflake-arctic-embed-l outperforms retrieval performance, having one-quarter of the estimated parameters and one-third of the dimensions compared to Open AI.

AUTHOR  
Snowflake AI Research

Provider	Model Name	Model Parameters (Millions)	Embedding Dimensions	Retrieval Performance
Snowflake	snowflake-arctic-embed-l	334	1024	55.98
Google	google-gecko.text-embedding-preview-0409	1200	768	55.7
Open AI	text-embedding-3-large	1200 (Estimated)	3072	55.44
Cohere	Cohere-embed-english-v3.0	1200 (Estimated)	1536	55
BAAI	bge-large-en-v1.5	335	1024	54.29

SHARE

f si in sr /flake-arctic-embed

Our models are incredibly easy to integrate with your existing search stack. Available directly from Hugging Face with an Apache 2 license, you can use this model for retrieval with five simple lines of Python.

```
import torch
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('Snowflake/snowflake-arctic-embed-l')
model = AutoModel.from_pretrained('Snowflake/snowflake-arctic-embed-l')

documents = ['Snowflake is the Data Cloud!', 'Snowflake is the Data Cloud!']
embeddings = model(**tokenizer(documents, padding=True, truncation=True))
```

### Secret sauce

Now, for the part, everyone is likely curious about: how are these models so good? The answer is simple: effective techniques from web searching are equally applicable to training text embedding models. While we will discuss all our findings in an upcoming in-depth technical report, we outline many of our findings on [how to train a state-of-the-art embedding model](#) and [how to use Snowflake to process training data](#). At a high level, we found that improved sampling strategies and competence-aware hard-negative mining could lead to massive improvements in quality. We would also be remiss to say that we did not build on the shoulders of giants, and in our training, we leverage initialized models such as ([bert-base-uncased](#), [nomic-embed-text-v1-](#)

[unsupervised](#), [e5-large-unsupervised](#), [sentence-transformers/all-MiniLM-L6-v2](#)).

AUTHOR

[Snowflake AI Research](#)

When our findings were combined with web search data and a quick iteration loop to gradually improve our model until the performance was something we were excited to share with the broader community. Notably, none of our significant improvements came from a massive expansion of the computing budget. All of our experiments used 8 H100 GPUs!

## Looking ahead

SHARE

This release is the first of many steps in our commitment to providing our customers with the best models for use in common enterprise use cases such as RAG and search. Using our deep expertise in search derived from the Neeva acquisition, combined with the incredible data processing power of Snowflake's Data Cloud, we have shared with the community a set of efficient models that provide the retrieval quality our customers require. We are rapidly expanding the types of models we train and the targeted workloads. Not only are we working on models, but we are also developing novel benchmarks that we will use to guide the development of the next generation of models. If you are interested in improving our models, have any suggestions, or want to join us in building the future, please reach out.

## Get started and learn more:

Access any of the embedding models in [Hugging Face](#)

Try snowflake-arctic-embed-m as part of Snowflake Cortex [embed function](#)

Come to our [meetup in San Francisco on Wednesday](#), April 17, to learn more about the Snowflake Arctic embed models and the people who made this release possible.

## Acknowledgments

We thank our modeling engineers, Danmei Xu, Luke Merrick, Gaurav Nuti, and Daniel Campos, for making these great models possible. We thank our leadership, Himabindu Pucha, Kelvin So,

Vivek Raghunathan, and Sridhar Ramaswamy, for supporting this work. We also thank the open-source community for producing the great models we could build on top of and making these releases possible. Finally, we thank the researchers who created BEIR and MTEB benchmarks. It is largely thanks to their tireless work to define what “better” looks like that we could improve model performance.

AUTHOR

Snowflake AI Research

SHARE



SHARE



Start your 30-day free trial

START NOW!



PLATFORM	SOLUTIONS	RESOURCES	EXPLORE	ABOUT
Cloud Data Platform	Snowflake for Financial Services	Resource Library	News	About Snowflake
Pricing		Webinars	Blog	Investor Relations
Marketplace	Snowflake for Advertising, Media, & Entertainment	Documentation	Trending	Leadership & Board
Security & Trust		Community	Guides	
	Snowflake for Retail & CPG	Procurement	Developers	Snowflake Ventures
	Healthcare & Life Sciences Data Cloud	Legal		Careers
	Snowflake for Marketing Analytics			Contact

Sign up for  
Snowflake  
Communications

diana.shaw@sno United States

By submitting this form, I understand Snowflake will process my personal information in accordance with their **Privacy Notice**. Additionally, I consent to my information being shared with Event Partners in accordance with Snowflake’s **Event Privacy Notice**. I understand I may withdraw my consent or update my preferences **here** at any time.

AUTHOR

Snowflake AI Research

SUBSCRIBE NOW

SHARE



[Privacy Notice](#) | [Terms](#) | [Cookie Settings](#) | [Do Not Share My Personal Information](#)

© 2024 Snowflake Inc. All Rights Reserved | If you'd rather not receive future emails from Snowflake, [unsubscribe here](#) or [customize your communication preferences](#)

