Representation model fine-tuning

Trung Nguyen
Applied Research Scientist, Snorkel AI

Key ideas

- Understanding representation models
- Fine-tuning best practices
- Applications of fine-tuning: RAG

Representation models

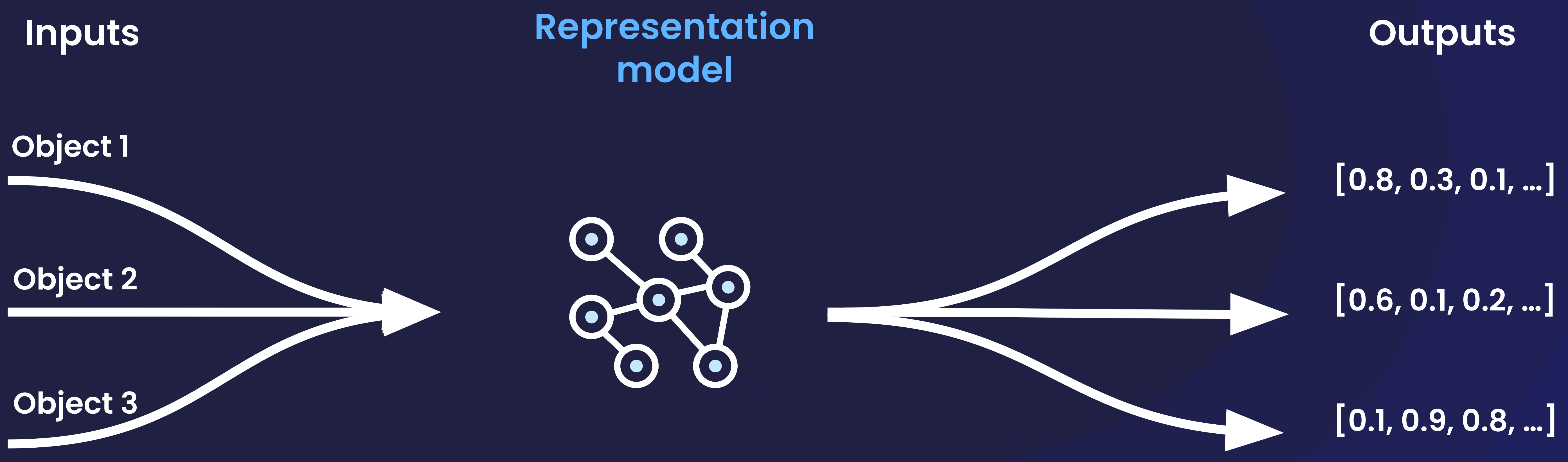
Introduction to representation models

- **Representation models** are a class of machine learning models designed to **capture and encode meaningful features** from the raw data it was trained on

- These representation of these are often in the form of **dense numerical vectors**

- The output of these models are used in downstream tasks such as **classification, clustering, similarity calculation, information retrieval, etc.**

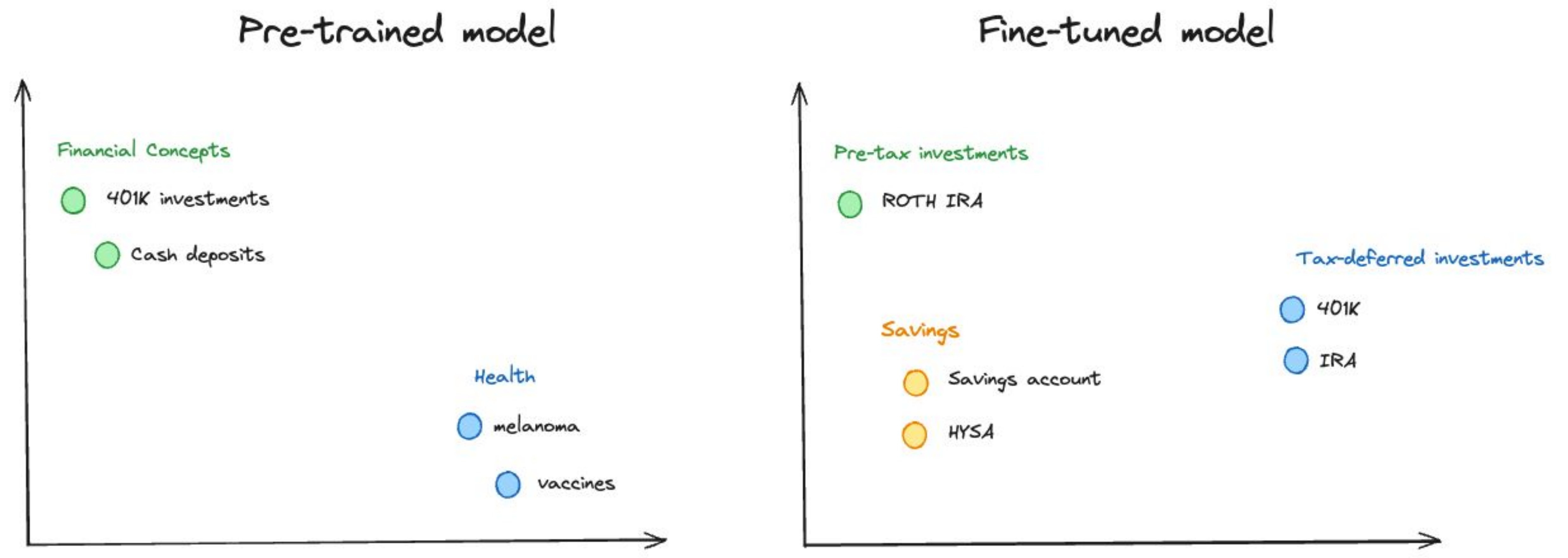
Representation Models



Representation model keep conceptually similar objects close together in the latent feature space

Snorkel

Representation models



Pre-trained models are general knowledge, fine-tuned models are specialized and more nuanced in a particular domain or task

Snorkel

The need for fine-tuning

- **Fine-tuning** is the process of **adapting/re-training** the model with your specific data and task
- **Pre-trained models are often general-purpose**, while **fine-tuned models are specialized for tasks**
- Fine-tuning pre-trained model can often lead to **better performance;** they can encode nuances of your specific data or domain

Fine-Tuning Deep Dive

Benefits of fine-tuning models

Improved performance

Faster convergence time

Transfer learning

Fine-tuning techniques

- **Task adaption -** fine-tune a portion of a network's layer(s). This approach typically demands **a smaller amount of data**.

- **Full model fine-tuning -** Training the entire model on a task-specific dataset. This typically involves a lot more data than just fine-tuning a subset of the layers.

Representation model fine-tuning best practices

- **Selectively curate and/or augment your dataset**
  - To be robust to new data points, consider data augmentation and/or generating synthetic data
- **Different loss functions can be used depending on your data**
  - **Triplet** or **contrastive loss** can be used if you have both positive and negative examples
  - **Multiple Negatives Ranking Loss** can be used when you only have positive pairs of text
- **Selecting an appropriate evaluation metric is key**
  - Evaluation can be difficult, map the evaluation to your downstream task, e.g., binary clf, semantic similarity, information retrieval, etc.

Quality data scarcity

- Fine-tuning representation models typically **require sizable task-specific datasets**. However, in practice, you may only have a small dataset that is of sufficient quality.

- This is common in use case such as **information retrieval**, where users may only have pair of positive text(query and relevant text answers), but not negative pairs (irrelevant answers)

How to overcome this issue?

Snorkel

Data augmentation can bootstrap dataset size and create robust representation model!

Data augmentation with Snorkel Flow

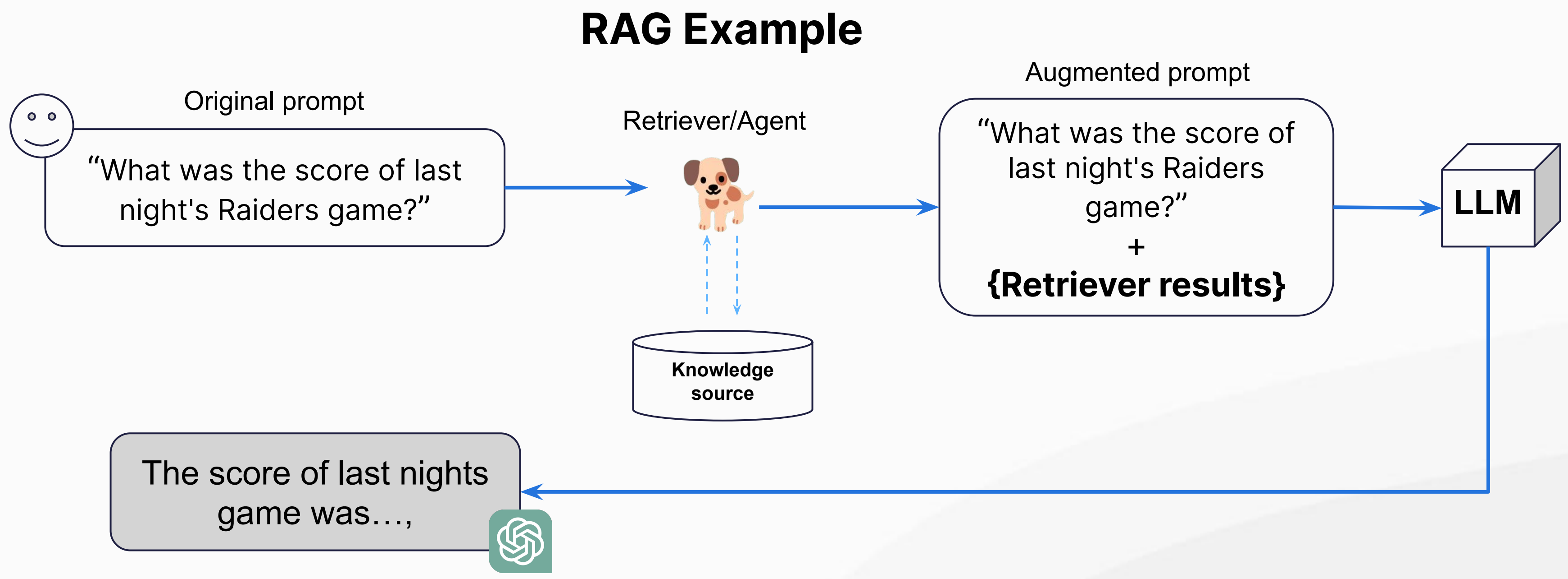
Fine-tuning representation models general requires **lots of data,** which is not scalable/expensive to label/collect. **Snorkel Flow** is an effective way to perform data augmentation to **curate a larger and higher quality dataset**

- Use **embedding and clustering techniques** to determine similar or dissimilar datapoint for dataset construction
- **Annotation workflow with your SMEs** to encode their expertise and find **hard negative** examples
- **Slice and filter your data** to determine where the errors are coming from, to focus additional augmentation efforts

Representation models in RAG

Retrieval augmented generation

- **Retrieval Augmented Generation (RAG)** is an AI technique that attempts to optimize the output of a LLM by incorporating an additional information retrieval step that enhances/augments the original prompt with the relevant search results

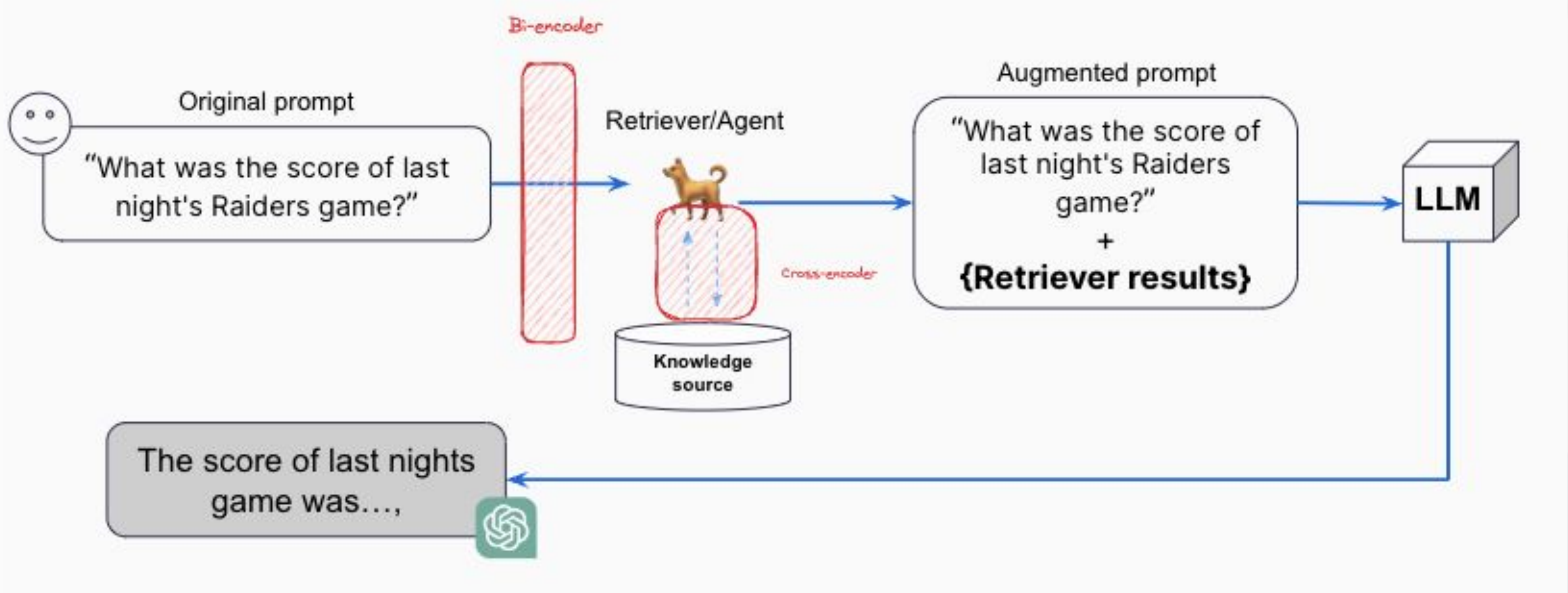


Representation models in RAG

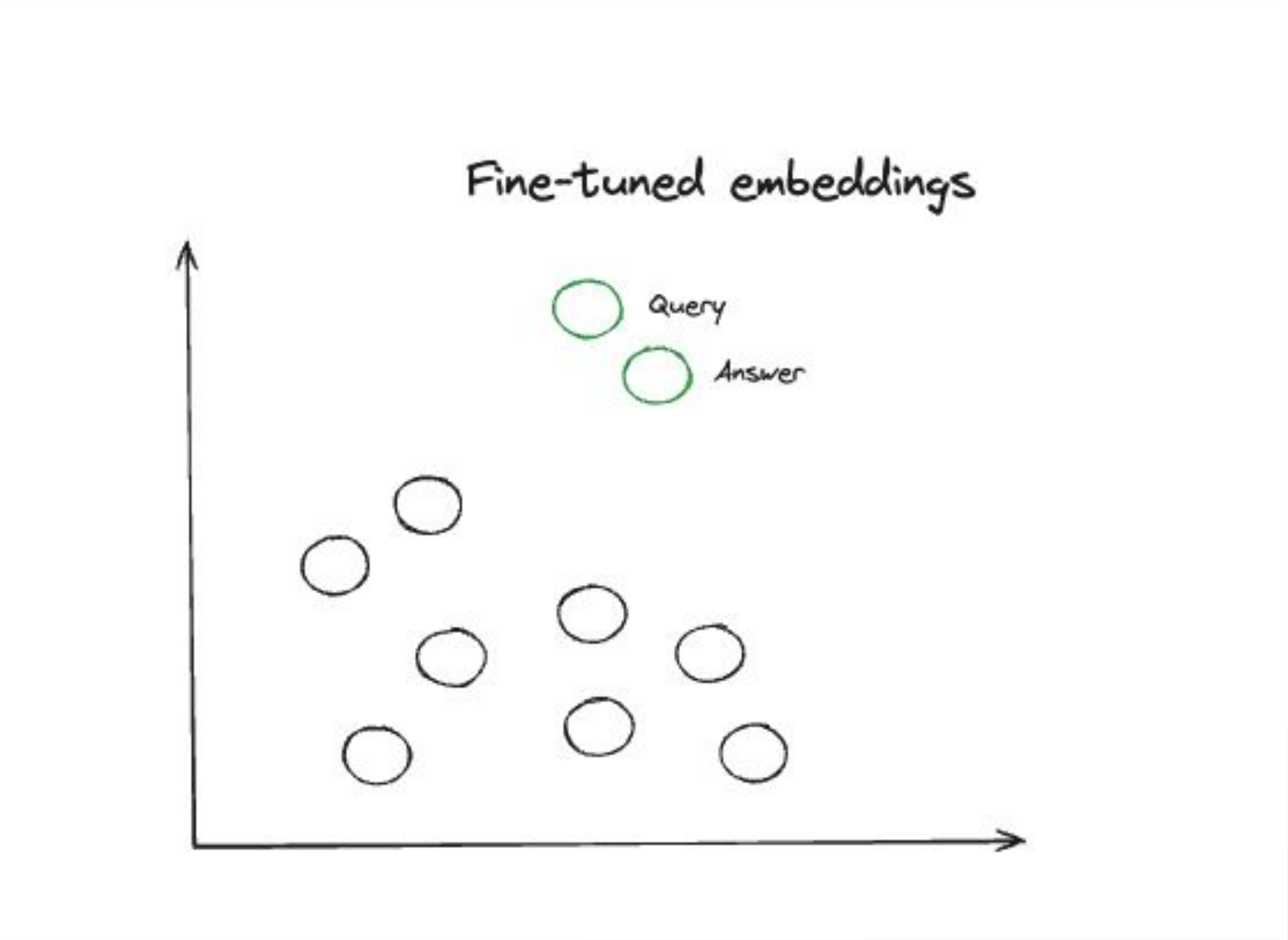
- **Bi-encoder** - family of models that take in a **single input** and returns a dense vector, these are typically used for creating the embedding **used for indexing and retrieval**

- **Cross-encoder -** family of models that take in **pairs of input** and return a class label or a similarity scores, **used in re-ranking**

Where does it fit in a RAG system?



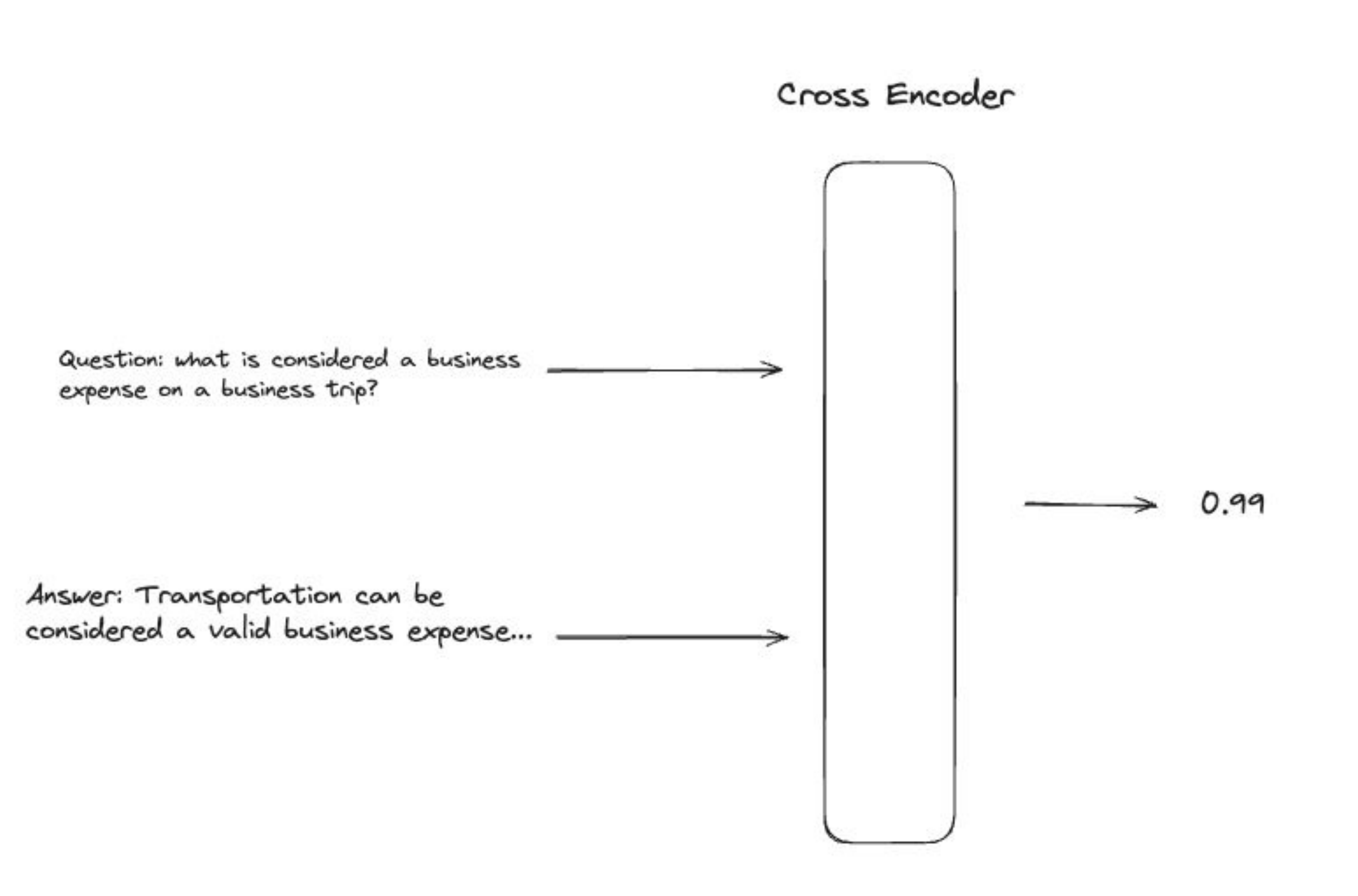
Fine-tuning Bi-encoder



Trains the model to map the **query embedding** and the **answer embedding** closer together

Snorkel

Fine-tuning Cross-Encoder



Trained on pairs of text, learns the **joint representation** of the input pairs. Outputs a similarity score or class label.

Snorkel

Fine-tuning Cross-Encoder

Retrieving relevant documents for the query “**what is considered a business expense on a business trip?**”

Before re-ranking

Text	Score
Business expense are defined as…	0.98
…	…
Transportation can be considered a valid business expense …	0.95
…	…
…	…

After re-ranking

Text	Score
Transportation be considered a valid business expense …	0.99
…	…
Business expense are defined as…	0.75
…	…
…	…

By representing both pair of text together, it provides a more effective way of evaluating relevancy

Benefits of fine-tuning in RAG

- Embed your **query and answer** closer together for **more effective retrieval of candidate documents**

- **Improve re-ranking of retrieved documents** to better use context for the answer synthesis by the LLM, by having relevant answer near the top

Snorkel

Recap

- Representation models **encode information about your data**
- **Fine-tuning can often improve performance of your task**. Data development is crucial and Snorkel Flow can allow you to perform data development efficiently
- There are multiple components of an information retrieval system like **RAG** that can be **fine-tuned such as the index generation and retrieval modules**

Thank you!

Trung Nguyen

trung.nguyen@snorkel.ai