NUDGE: LIGHTWEIGHT NON-PARAMETRIC FINE-TUNING OF EMBEDDINGS FOR RETRIEVAL

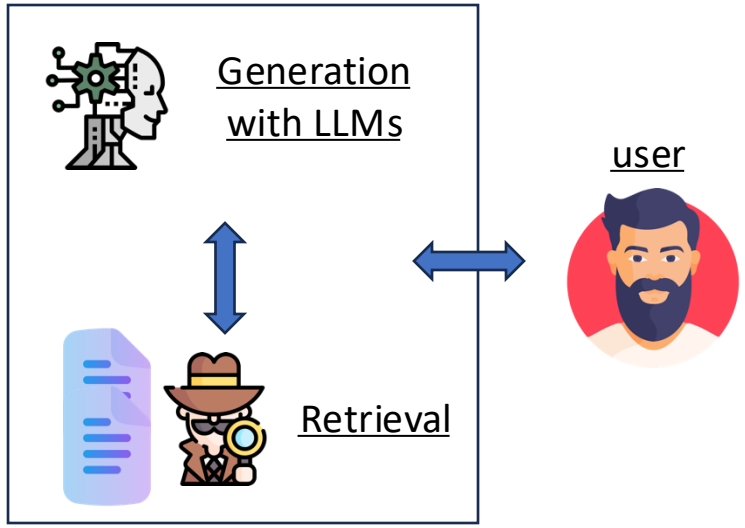
Sepanta Zeighami, Zac Wellmer, Aditya Parameswaran

UC Berkeley

k-Nearest Neighbor Retrieval

Text and image retrieval

- Traditional search systems
- Retrieval Augment Generation (RAG) pipelines
  - Summarize news articles about climate change
  - Find police officers with misconduct from court cases
  - Find what bicycle parts fit each other based on manual

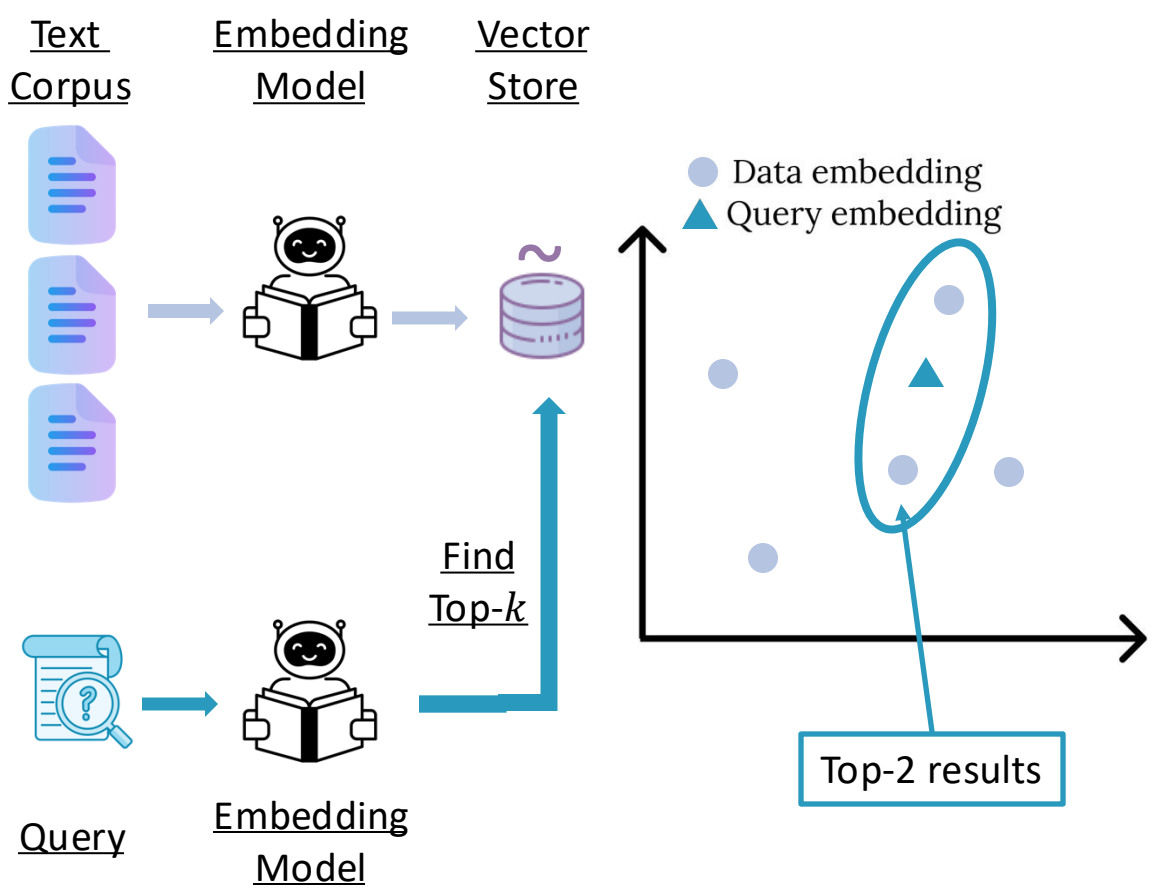


k-Nearest Neighbor (k-NN) Retrieval is de-factor standard

Simple, effective and efficient

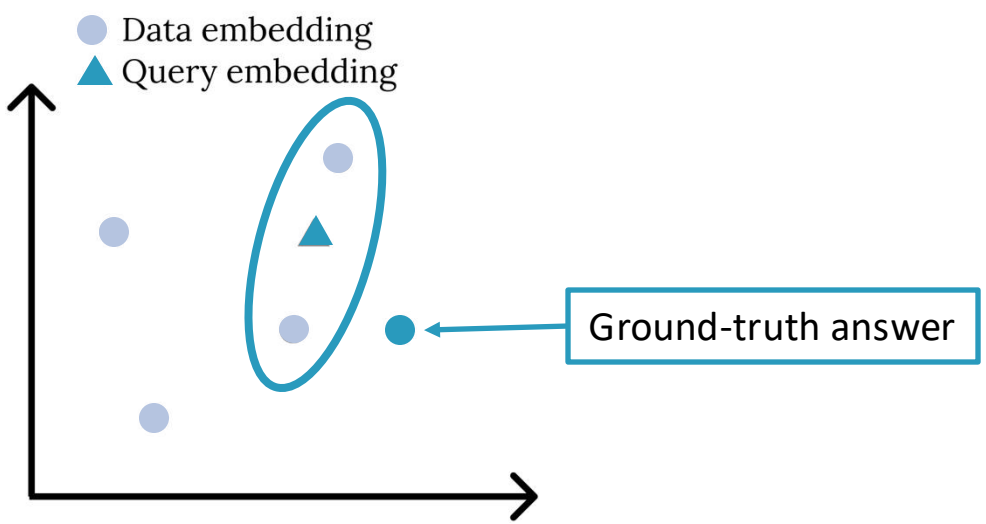
k-Nearest Neighbor Retrieval

- Embed documents
  - Embedding captures semantics
  - Pre-trained embedding model
  - Often stored in Vector Databases for efficiency
- To answer queries
  - Embed query
  - Retrieve top-k documents
    - Cosine similarity or dot product



Problem

- Retrieval may miss relevant records



Example:

Q: What are sources of chemical contamination in people?

Ground-truth Document:

Background: Meat could be involved in bladder carcinogenesis via multiple potentially carcinogenic meat-related compounds related to cooking and processing, including nitrate, nitrite, heterocyclic amines (HCAs), and polycyclic aromatic hydrocarbons (PAHs). The authors comprehensively investigated the association between meat and meat components and bladder cancer.

Q: What chemicals cause cancer?

Should answer related questions correctly next time

Bridge the semantic gap for the specific dataset in hand

Challenges:

Semantic gap between queries and documents

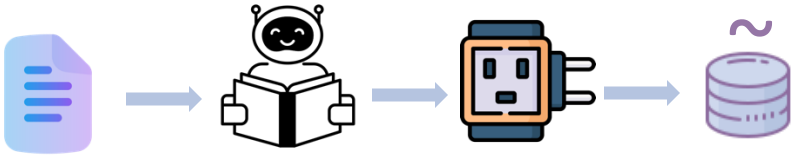
Domain-specific vocabulary unseen during training

Inaccuracies in embedding model

Solution:

Fine-tune embeddings to improve accuracy!

Fine-Tuning of Embeddings



Fine-Tuning Embedding Models

Modify parameters of the embedding model

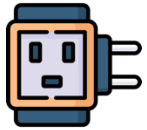
Improves accuracy 🙂

High runtime and computational resource ☹️

Needs access to model parameter ☹️

Additional hosting and maintenance costs ☹️

Training Adaptors



Train a new Adaptor model to modify the output of the embedding model

Limited accuracy gain ☹️

Efficient 🙂

Doesn't need model parameters 🙂

Additional hosting and maintenance costs ☹️

Both approaches are *parametric*: modify model parameters to change embeddings

No approach that is efficient, effective and easy to use

NUDGE

- NUDGE is a lightweight *non-parametric* fine-tuning method
  - Modifies embeddings not model parameters

Accurate

- Often boosts accuracy by at least 10%
- No out-of-distribution regression

Efficient

- Runs in minutes on CPU
- No corpus re-embedding after fine-tuning
- No extra model inference at test time

Model Agnostic

- Doesn't need access to model parameters
- Works with closed-source models

Low-Maintenance

- Efficient insertion support
- No model hosting costs

NUDGE Overview

- NUDGE solves constrained non-parametric optimization problems to fine-tune embeddings

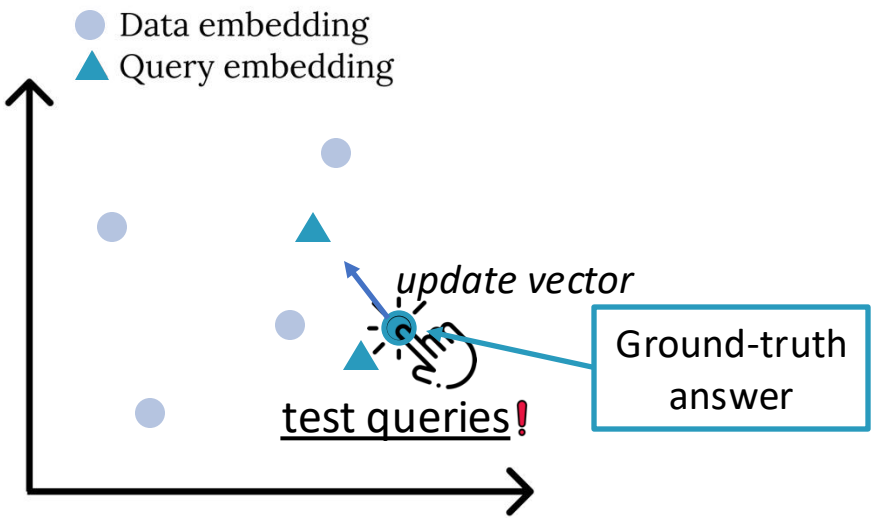
Q: What are sources of chemical contamination in people?

Background: Meat could be involved in bladder carcinogenesis via multiple potentially carcinogenic meat-related compounds related to cooking and processing, including nitrate, nitrite, heterocyclic amines (HCAs), and polycyclic aromatic hydrocarbons (PAHs). The authors comprehensively investigated the association between meat and meat components and bladder cancer.

Q: What chemicals cause cancer?

<u>"nudges" embeddings to maximize similarity</u>

- Find update vectors to change embeddings



NUDGE Overview

- NUDGE solves constrained non-parametric optimization problems to fine-tune embeddings

Q: What are sources of chemical contamination in people?

Background: Meat could be involved in bladder carcinogenesis via multiple potentially carcinogenic meat-related compounds related to cooking and processing, including nitrate, nitrite, heterocyclic amines (HCAs), and polycyclic aromatic hydrocarbons (PAHs). The authors comprehensively investigated the association between meat and meat components and bladder cancer.

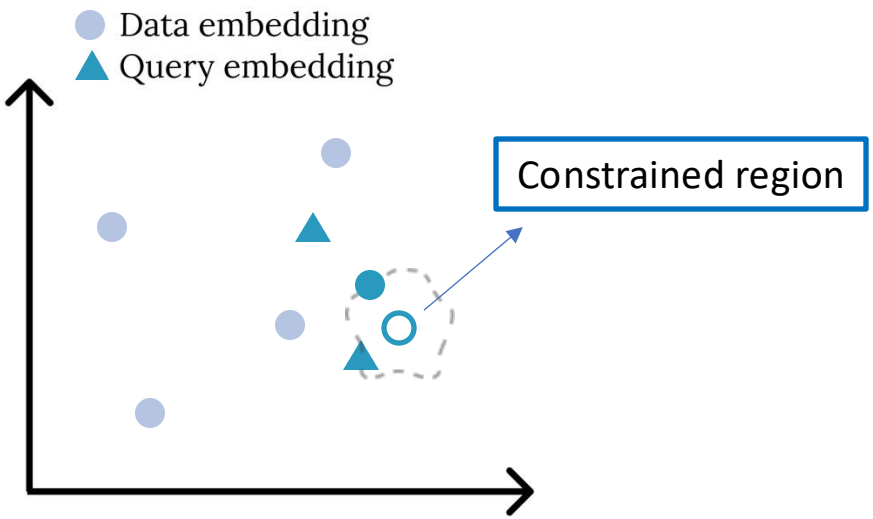
Q: What chemicals cause cancer?

<u>"nudges" embeddings to maximize similarity</u>

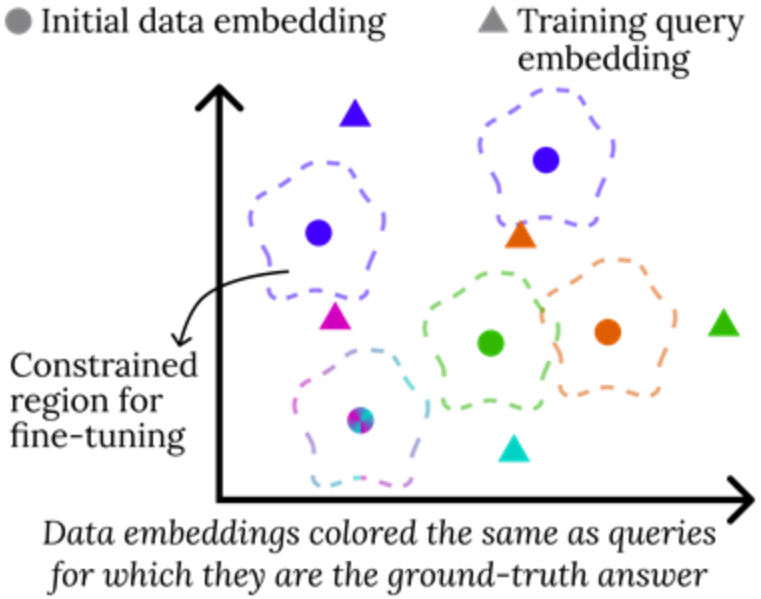
- Find update vectors to change embeddings

<u>Constrain how and how much embedding change</u>

- Avoids overfitting
- Does not distort pre-trained semantics



NUDGE Overview



- Given a set of queries and ground-truth answers
- Modify data embeddings to maximize similarity between queries and ground-truth answer
- Within a constrained region

Different NUDGE Variants based on constraints used

I'll discuss NUDGE-N, see paper for other variants

Fine-tuning through constrained non-parametric optimization

Non-trivial to solve in general

- Maximizing *accuracy* instead of *similarity* is NP-Hard

Problem formulation resolves *computational* and *overfitting challenges*

NUDGE-N

Normalized

NUDGE-N Optimization Problem

Given a set of queries and ground-truth answers

Modify data embeddings to maximize similarity between queries and ground-truth answer

Such that

- New embeddings are normalized
- Embeddings change by at most γ

With γ that maximizes validation accuracy

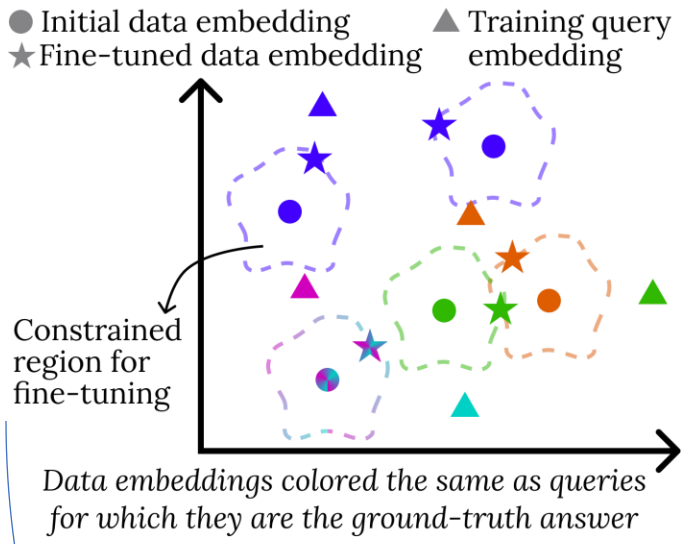
Bi-level optimization problem

NUDGE-N is the optimal solution

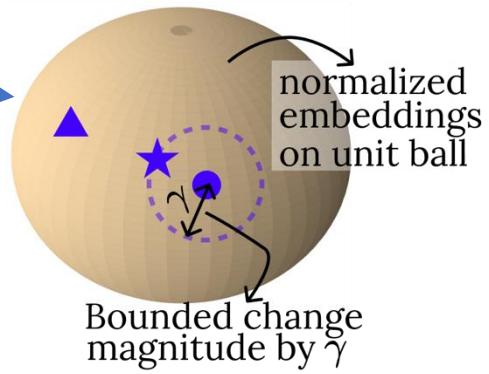
Finds optimal γ

Closed-form update formula applied to all embeddings with simple matrix addition

Runs in time (almost) equivalent to a single training and validation iteration of parametric methods



NUDGE-N Constraints



Summary of Experimental Results

- Results across 9 standard text/image benchmarks (from BEIR, KILT, COCO and Flickr)
- 5 different embedding models

NDCG@10 and Recall@10

NUDGE often achieves accuracy at least **10% higher than existing methods**

Runs in around **10 minutes** even on CPU

3.3x and 4.3x higher accuracy boost compared with Adapter and fine-tuning the pre-trained model

200x and 3x faster than baselines

Conclusion

- NUDGE runs in minutes and significantly boosts accuracy
  - It solves constrained optimization problems to fine-tune embeddings
  - It presents *non-parametric* fine-tuning as a novel effective means of improving pre-trained models
- Simply add it to your RAG pipelines after you embed the corpus at ingestion time

- Use it for any RAG application, check it out on github, or try it out in LlamaIndex!

```
pip install nudge-ft
```

```
github.com/szeighami/nudge
```

LlamaIndex

Thanks! Q&A

Results

Runs within minutes even on CPU

Method	Time GPU (mins.)	Time CPU (mins.)
NUDGE-N	2.18	11.0
Adaptor	7.99	77.8
PTFT	447	N/A

Runtime to obtain BGE-S results

- Evaluate accuracy gain from fine-tuning
- Results are averaged across 7 standard text benchmarks from BEIR and KILT
- R@k is Recall@k

Emb. Model → ↓ Method	BGE-S (*33m #params, 384 embedding dim.*)			GTE-L (*434m #params, 1024 embedding dim.*)			TE3-L (*OpenAI Text-Embedding-Large-3, 3072 embedding dim.*)		
NUDGE-N	**52.0**	**72.6**	**61.1 (+12.4)**	**53.4**	**74.8**	**62.7 (+9.4)**	**55.2**	**76.0**	**63.9 (+11.7)**
Adapter	39.5	65.5	51.6 (+2.9)	45.1	68.4	55.7 (+2.4)	46.9	66.2	54.4 (+2.2)
PTFT	40.9	66.1	52.5 (+3.8)	N/A	N/A	N/A	N/A	N/A	N/A
No Fine-Tuning	37.0	62.4	48.7	41.6	67.0	53.3	40.0	67.5	52.2

Fine-tuning pre-trained model parameters

Significant accuracy boost using NUDGE across models!

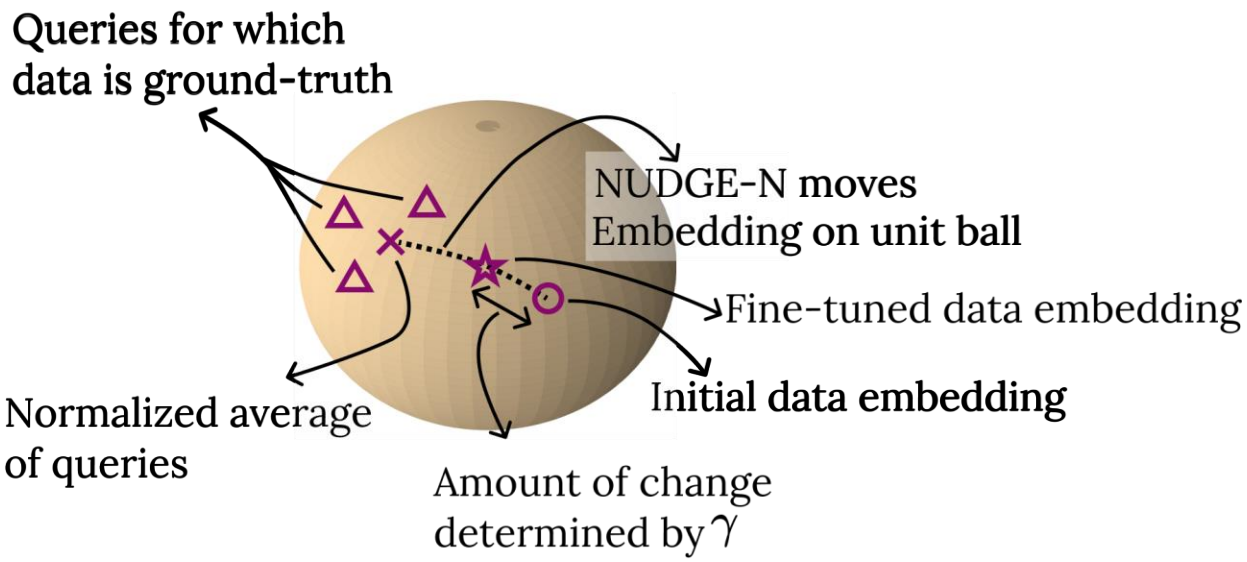
Can't be run due to computational constraints

Can't be run because model is closed-source

NUDGE-N Algorithm

- NUDGE-N solves the problem optimally in closed-form

Solution Overview



Derives an update formula applied to all embeddings with simple matrix addition	Finds optimal γ	Runs in time (almost) equivalent to a single training and validation of parametric methods

Out-Of-Distribution (OOD) Results

- Queries clustered into two sets using k-means
  - Trained and tested on the same cluster for in-distributions
  - Trained on one cluster, tested on another for out-of-distribution

Method	In-Distribution			Out-of-Distribution		
	R@1	R@10	NDCG@10	R@1	R@10	NDCG@10
NUDGE-M	49.9	67.2	57.6 (+8.8)	28.9	45.9	36.6 (-11.6)
NUDGE-N	**51.6**	**71.9**	**60.8 (+12.0)**	**38.7**	**63.7**	**49.9 (+1.7)**
Adaptor	38.9	64.7	51.0 (+2.3)	35.9	62.8	47.7 (-0.5)
PTFT	41.0	67.4	53.7 (+4.9)	36.1	62.5	48.2 (+0)
No Fine-Tuning	36.9	62.3	48.8	37.0	62.3	48.2

Shows importance of normalization

Outperforms other methods even in OOD setting

Parametric methods may regress in OOD setting