

EEP 596: LLMs: From Transformers to GPT || Lecture 10

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 10, 2025

Deep Learning References

Deep Learning

Great reference for the theory and fundamentals of deep learning: Book by Goodfellow and Bengio et al [Bengio et al](#)

[Deep Learning History](#)

Embeddings

[SBERT and its usefulness](#) [SBert Details](#)

[Instacart Search Relevance](#) [Instacart Auto-Complete](#)

Transformers and Attention

[Illustration of attention mechanism](#)

House Keeping

- MP2 (Mini-Project 2) part 1 and 2 assigned
- MP2 focuses on the use of Pine-cone Vector DB and also building a Streamlit application for a functional chat-bot
- MP2 Part 3 will focus more on the design aspects of the chatbot
- Any questions/concerns?

Previous Lecture

- Prompt Engineering Principles and In-class coding exercise

Today's lecture

- Instacart Product Recommendations use-case
- Prompt Engineering Best practices
- Advanced PE Code walk through
- Advanced PE Coding exercise

Instacart Recommendations

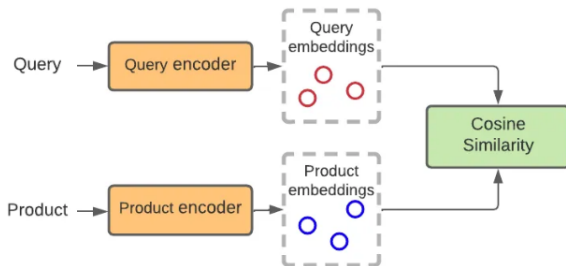


Figure 1. Conceptual diagram of a two-tower model

Two Tower Architecture

Two Towers

Self-explanatory, but there are two towers that represent two distinct objects (e.g. sentence A and sentence B or query and product or customer and product, etc).

Two Tower Architecture

Two Towers

Self-explanatory, but there are two towers that represent two distinct objects (e.g. sentence A and sentence B or query and product or customer and product, etc).

SBERT Two Tower

Is a **Siamese Two Tower**, where the weights and layers of the two towers are *identical*. In the training of a Siamese two-tower, the weights are said to be tied together between the two towers and gradients are computed keeping the tying in place.

Two Tower Architecture

Two Towers

Self-explanatory, but there are two towers that represent two distinct objects (e.g. sentence A and sentence B or query and product or customer and product, etc).

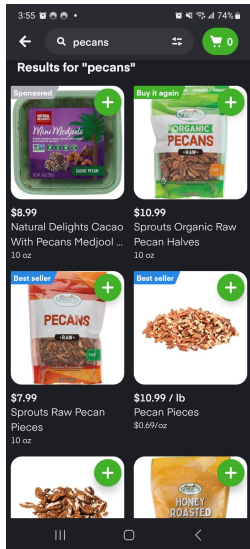
SBERT Two Tower

Is a **Siamese Two Tower**, where the weights and layers of the two towers are *identical*. In the training of a Siamese two-tower, the weights are said to be tied together between the two towers and gradients are computed keeping the tying in place.

Instacart/Recommendations Two Tower

In this example, the two towers don't refer to the same kind of object (e.g. sentence) but refer to a product and query. Hence the two towers have distinct weights learned from the data.

Positive Examples

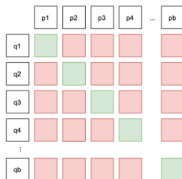


High-quality Positive Examples

Converted Products for Search Query "Orange"
Navel Oranges
Clementines
Mandarins
...
Bananas
...
Strawberries

Negative Examples

Vanilla In-batch Negative



In-batch Negative with Self-adversarial Re-weighting

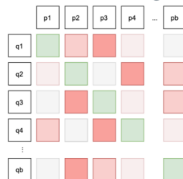
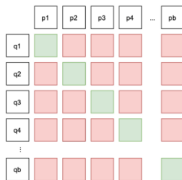


Figure 3. (Left) In the vanilla implementation of in-batch negative, all off-diagonal negative samples are given the same weight. (Right) In our implementation with self-adversarial re-weighting, harder examples are given more weight (darker color), making the task more challenging for the model.

Negative Examples

Vanilla In-batch Negative



In-batch Negative with Self-adversarial Re-weighting

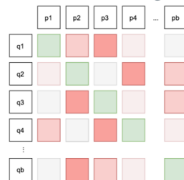


Figure 3. (Left) In the vanilla implementation of in-batch negative, all off-diagonal negative samples are given the same weight. (Right) In our implementation with self-adversarial re-weighting, harder examples are given more weight (darker color), making the task more challenging for the model.

Self-adversarial data annotation

Easy Negative examples: Tortilla → Coffee mug

Hard Negative examples: Tortilla → Tostitos Tortilla Chips

Data Augmentation for Data Set expansion

Two kinds of Data Augmentation/Data Expansion

- 1 **Expanding Product Signals:** This refers to not just using product titles but also product description or even images (multi-modal signals) for better *Product Embedding*
- 2 **Expanding Cold Start Data:** Products that just got launched or are new to the Instacart ecosystem get surfaced through data augmentation. Here - (Query, Product) examples are **synthetically** created as training data for the model so it can learn to recognize and recommend new products.

Data Augmentation for Data Set expansion

Data Augmentation in LLM context

This is a fairly common strategy that gets used in NLP tasks and in the use of LLMs. For instance - Microsoft's **Phi** model, which is a **Small Language Model**(SLM) was trained in part with high-quality *textbook data*, where the textbooks themselves got generated using a more powerful GPT model!

Data Augmentation for Data Set expansion

Data Augmentation in LLM context

This is a fairly common strategy that gets used in NLP tasks and in the use of LLMs. For instance - Microsoft's **Phi** model, which is a **Small Language Model**(SLM) was trained in part with high-quality *textbook data*, where the textbooks themselves got generated using a more powerful GPT model!

LLMs as annotators and paraphraser

Also used often, analogous to the previous Phi model example is annotating inputs with targets using an accurate GPT model or generating more training data through paraphrase of the input.

Breakouts Time #1: Product Review Classification (12 mins)

Classifying product reviews

Let's say that you are a data scientist at Sambazon! Sambazon is an online retailer selling millions of products under tens of thousands of product categories. You work in the **Review Moderation and Insights** team that is responsible for deriving actionable insights from customer reviews data. Your team's charter includes understanding the **intent** of the reviews - esp. if its useful or obnoxious. Your team's product manager (PM) suggests that as part of this years roadmap, the product team would like to understand reviews from the lens of the following categories: highly useful, highly passionate, obnoxious and balanced. How would you as a scientist a) approach this problem b) What would be your sources of data? c) What would be the ML approach you would use? d) How would you train the model? e) What if you didn't have labels in the data as your PM suggested? f) What if you had labels for training but not enough data?

Model Training Architecture

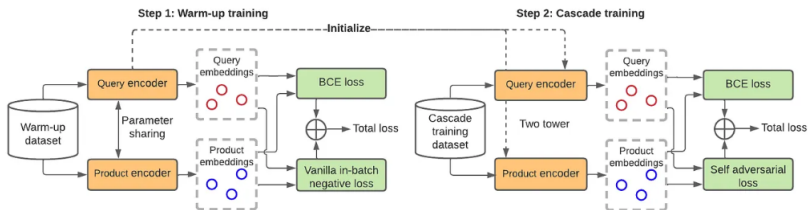


Figure 4. Two-step cascade training for ITEMS.

System Design

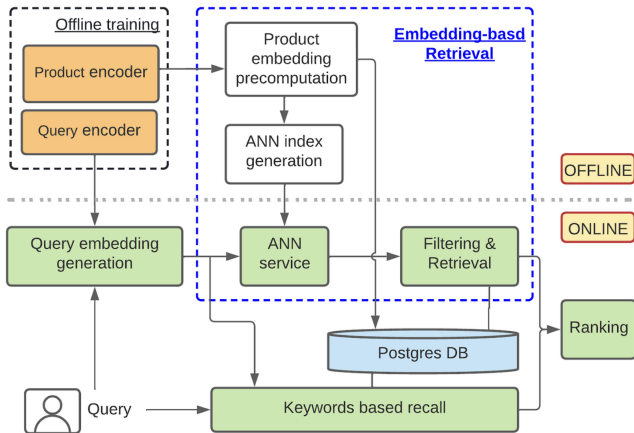
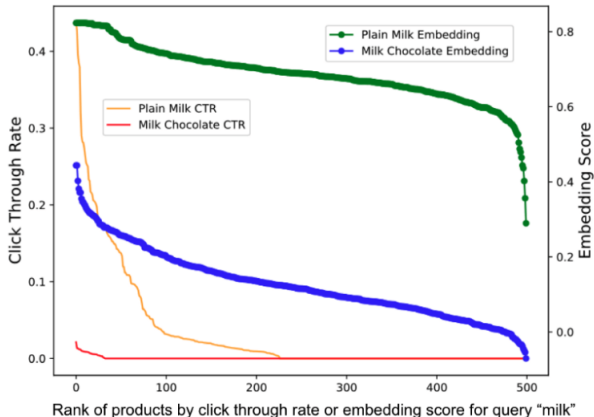


Figure 7. ITEMS system architecture.

CTR vs Embedding Score

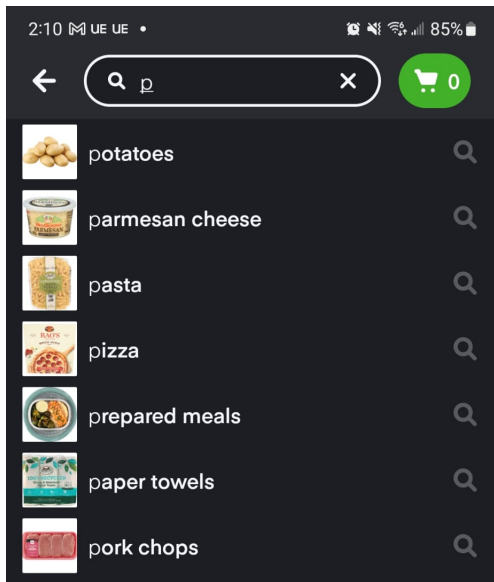


Breakouts Time #2

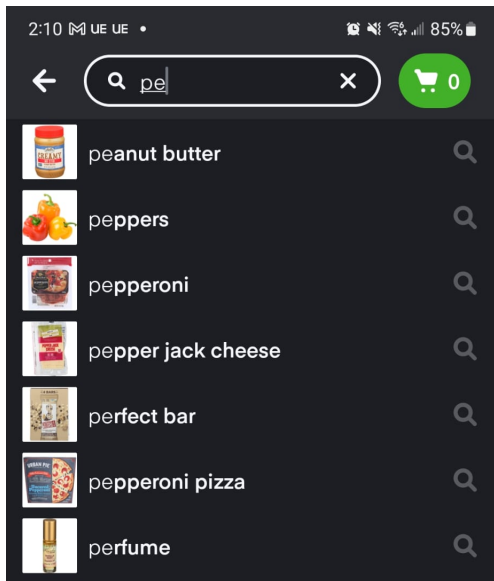
Instacart Auto-complete — 5 mins

Let's say you are tasked with building an auto-complete search for instacart, which can help complete partial words into full words/sentences through suggestions (auto-complete). How would you use what we have learned so far to model this? What architecture would you use? What would be your data? And what are some pitfalls or painpoints your model should address?

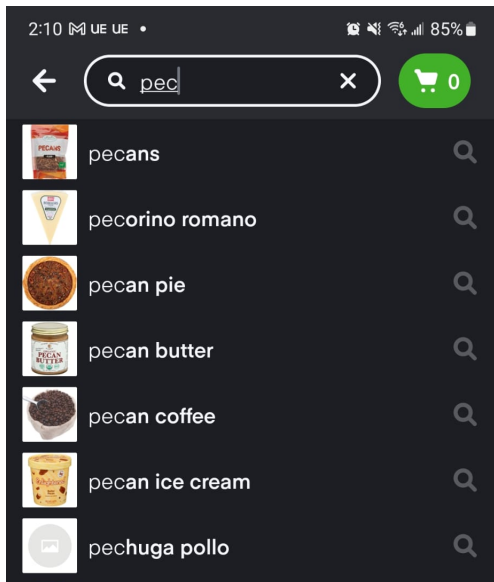
Instacart Auto-Complete and Search Relevance



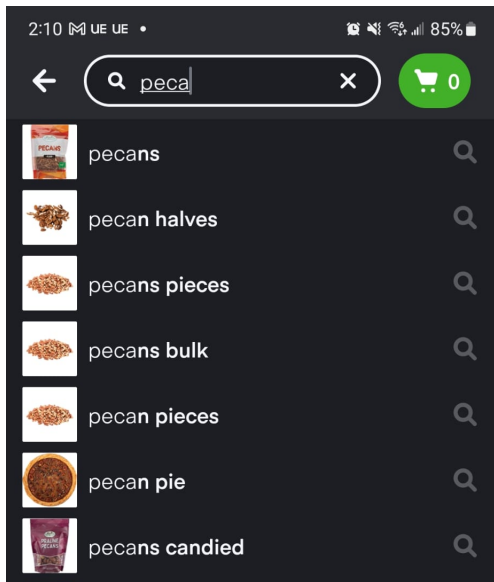
Instacart Auto-Complete



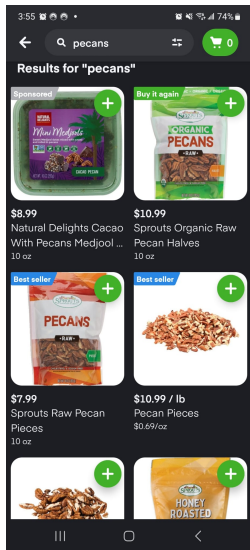
Instacart Auto-Complete



Instacart Auto-Complete



Instacart Auto-Complete and Search Results



Instacart Diversifying Auto-Complete

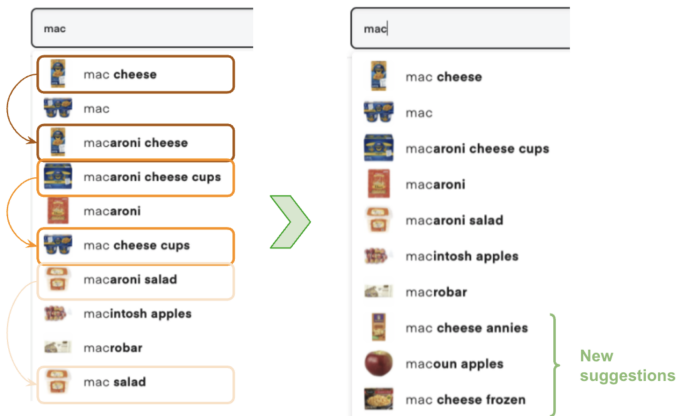


Figure 9. Autocomplete when a customer searches for "mac", before (left) and after (right) semantic deduplication.

Prompt Engineering Best Practices