EEP 596: LLMs: From Transformers to GPT || Lecture 9

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 6, 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

MAP → Mean Average Precision

- MP1 Part 2 due this weekend
- Kaggle contest consolidates your learnings from MP1 Part 2 to then match up with your peers on a leaderboard over a friendly competition
- Top 5 teams get bonus points

  Groundtruth: 1 2 3 4 5
- Can also add this to your portfolio!

  Pred:- 5 4 3 2 1
- Office Hours/Review Hours/Grading Hours
- Anything else?

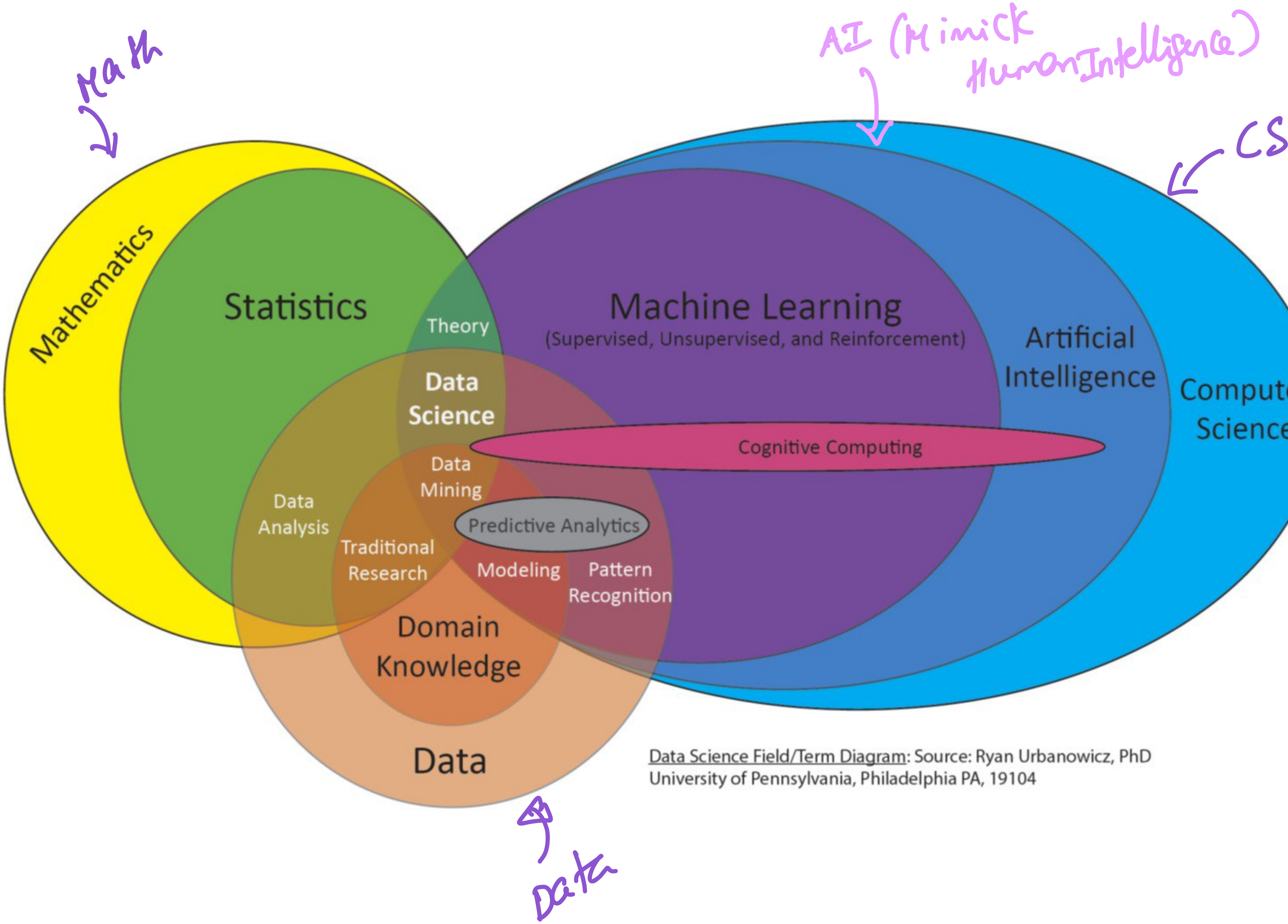
Previous Lecture

- Multi-Head Attention
- Fine-tuning BERT and SBERT

Today's lecture

- Prompt Engineering Principles and In-class coding exercise
- Toggle between Architectures, some Math, ML Modeling, Business use-cases and coding/demos

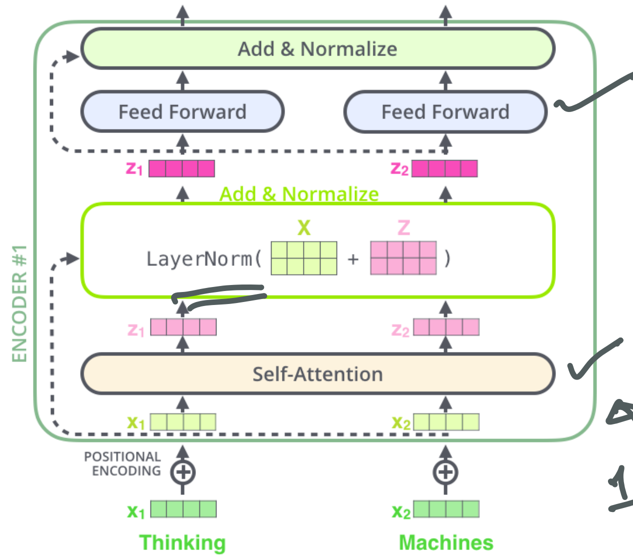
It's an ocean for a reason!



Recap from Last Lecture

- Focus on two types of transformers: Encoding and Decoding Transformers
- Encoding Transformers are good for discriminative tasks (classification, embeddings, semantic search, intent detection, etc)
- Decoding Transformers are needed for text generation (like GPT) but can also be used for discriminative tasks (since GPT-3 and all recent open models like Llama-2/3)
- Encoding Transformer consists of Multi-Head Attention (MHA) and Feed Forward NN block

Recap from Last Lecture



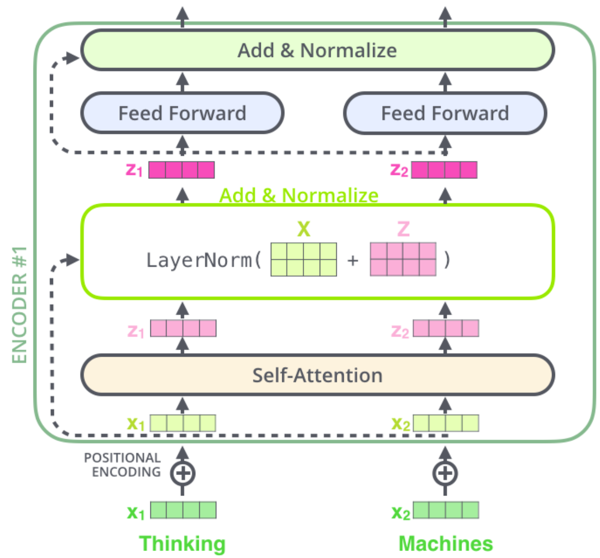
Encoding 1 Transformer Block

- Encoding Transformer consists of **Multi-Head Attention (MHA) and Feed Forward NN block**
- MHA serves the purpose of putting the token in the context of all other tokens (e.g. "I own a dog. It is out playing right now." - It refers to a dog).

Attention

- Feed Forward - Ensures the tokens are "non-linear" transformed and non-linearity can help with complex understanding of the sentences.

Recap from Last Lecture



- We looked at the Math behind FFN and MHA
- A lot of it is essentially **matrix algebra** of layers of abstraction (e.g. Query, Keys, Values, Projection matrices and Relu functions).
- Transfomers use MHA, FFN as **building blocks** and **data** as source of truth through **loss function** optimization to then arrive at parameters that can then mimick a functional language model

ICE #1

CLS (Start token)
768 d

Last Layer | Last Hidden Layer → T1 T2 T3 . . . . T20] 20 tokens

BERT Embeddings and Emotion Detection

Let's say you want to do emotion detection by fine-tuning BERT (Encoding Transformer) on a data set. One of the outputs of the *BERT pre-trained model* for a given input is the *last-hidden-state*. This includes an embedding for every token that was passed into BERT.
Let's say you are going to start with the last hidden layer and use that as input for your *fine-tuned* model. This ICE is about the dimensionality of the inputs and outputs. Let's say you have sentences of the kind: "I am looking forward to today! It's going to be a big day" This sentence conveys excitement. There are 13 words in this input and using *word-piece tokenization*, you arrive at 20 sub-tokens as input into the BERT model. The last hidden layer includes an embedding for every single token. Let's say the embedding dimension for a token is 768.

ICE #1 continued

BERT Embeddings and Emotion Detection

There are 13 words in this input and using *word-piece tokenization*, you arrive at 20 sub-tokens as input into the BERT model. The last hidden layer includes an embedding for every single token. Let's say the embedding dimension for a token is 768. For the purpose of emotion detection - You can either use the *CLS* token (Start token) embedding (also called the pooled embedding) or you can take the average of the embeddings of the tokens in the last hidden layer of BERT. a) What's the dimension of the pooled BERT embedding of this particular input example b) What's the dimension of the CLS/Start token embedding in this example? c) what's the total dimension of the last hidden layer?

1. 768, 15630 and 768
2. 768, 768 and 768
3. 15360, 768 and 15630
4. 768, 768 and 15360

ICE #2

Why does pooling of the output need to be done for sequence classification (e.g. emotion detection)?

1. Reduces the dimensionality
2. Averages context from all the tokens
3. Computational concerns for training the fine-tuned model
4. All of the above

CLS

Prompt Engineering Principles Walkthrough

Instacart Recommendations

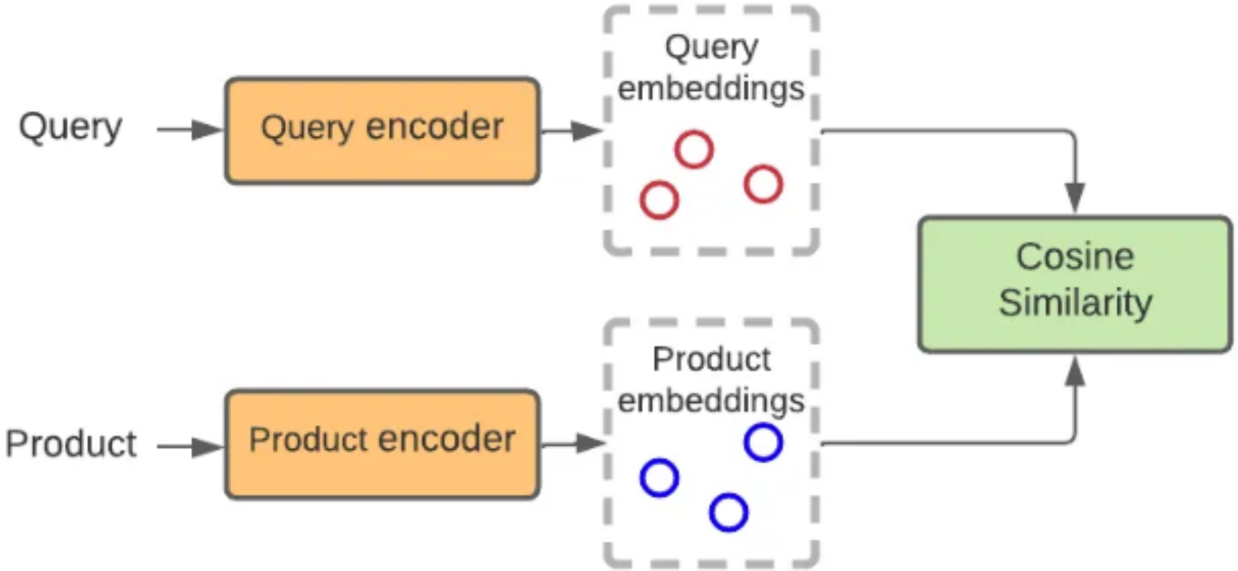


Figure 1. Conceptual diagram of a two-tower model

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).

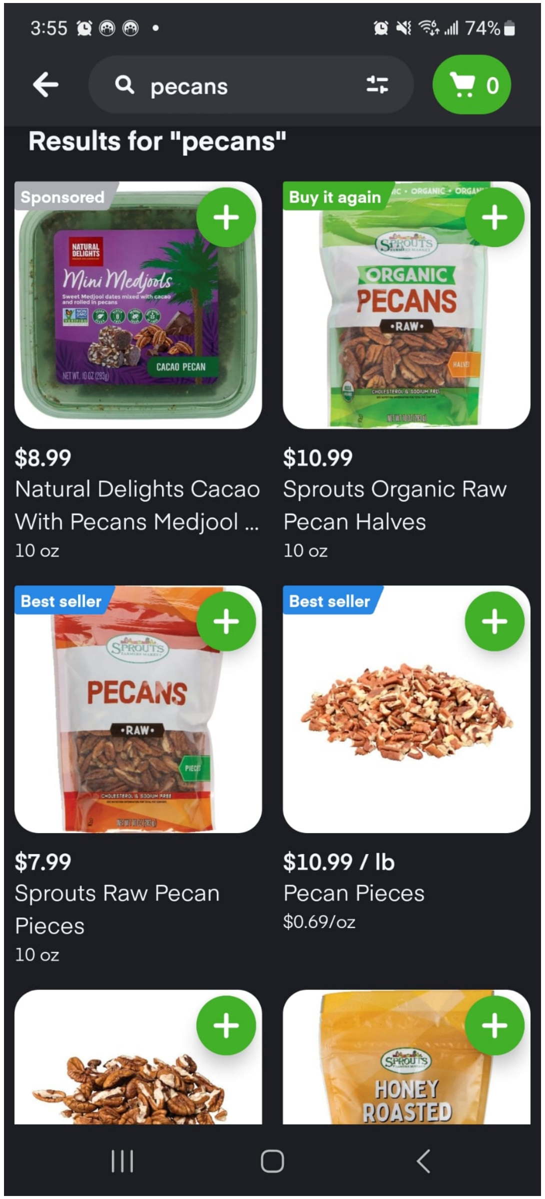
SBERT Two Tower

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

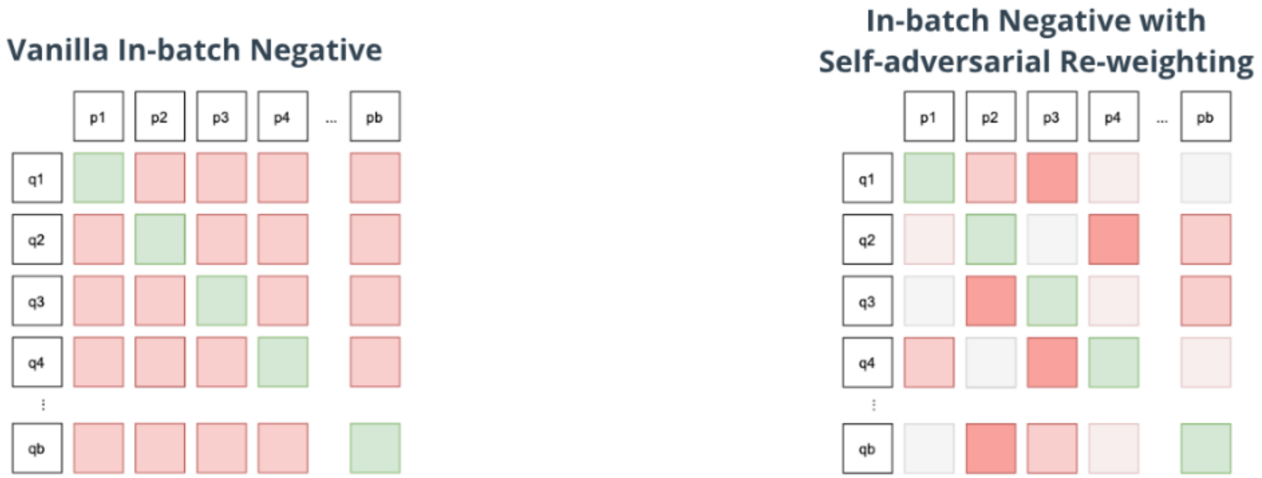


Figure 3. (Left) In the vanilla implementation of in-batch negative, all off-diagonal negative samples are given the same weight. (Right) In our implementation with self-adversarial re-weighting, harder examples are given more weight (darker color), making the task more challenging for the model.

Negative Examples

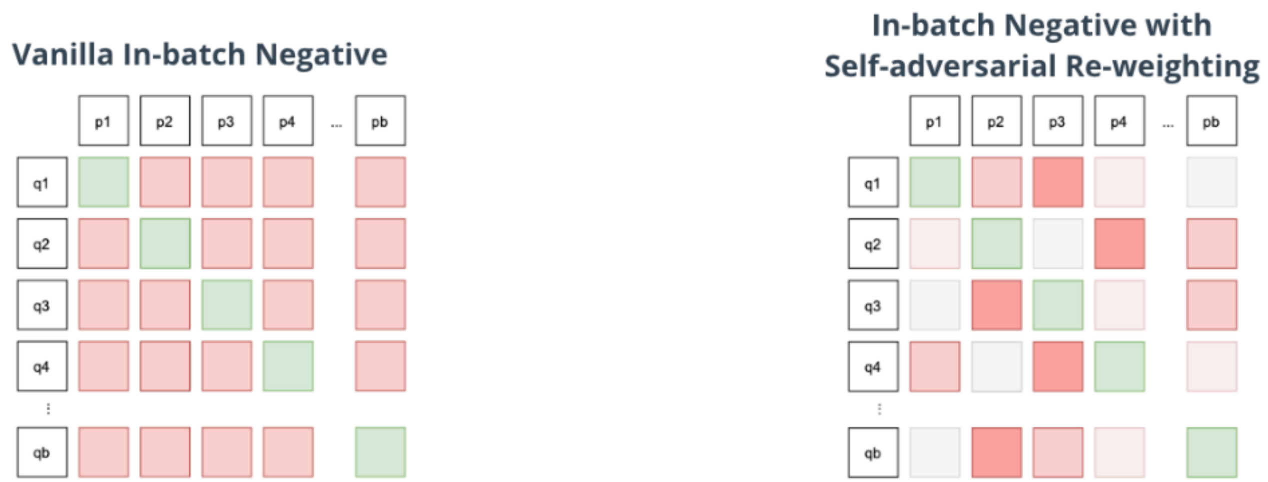


Figure 3. (Left) In the vanilla implementation of in-batch negative, all off-diagonal negative samples are given the same weight. (Right) In our implementation with self-adversarial re-weighting, harder examples are given more weight (darker color), making the task more challenging for the model.

Self-adverserial 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 bettery *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 exapansion

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 exapansion

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 paraphrasers

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

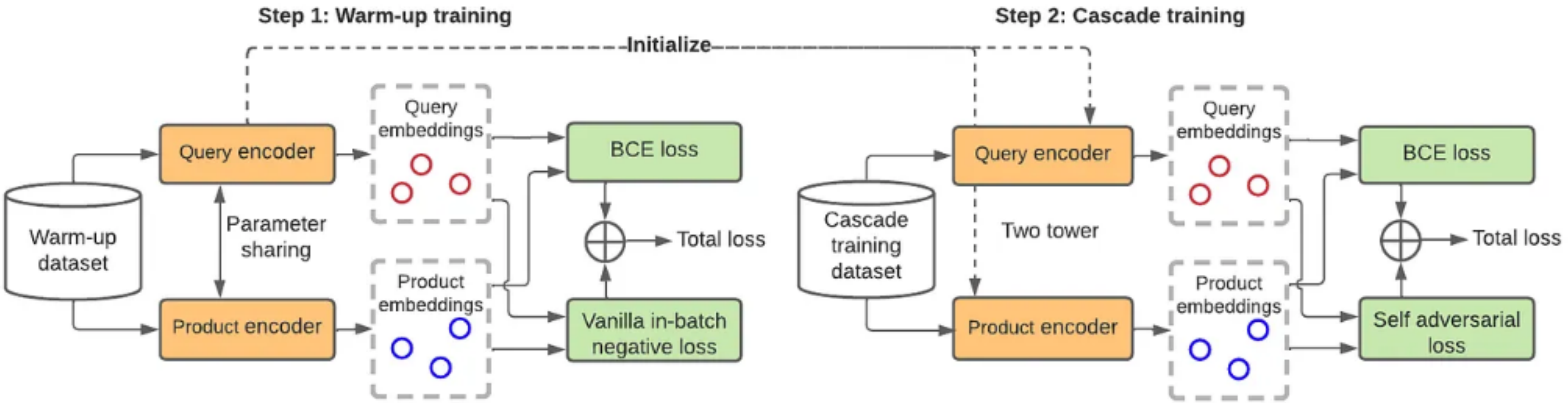


Figure 4. Two-step cascade training for ITEMS.

System Design

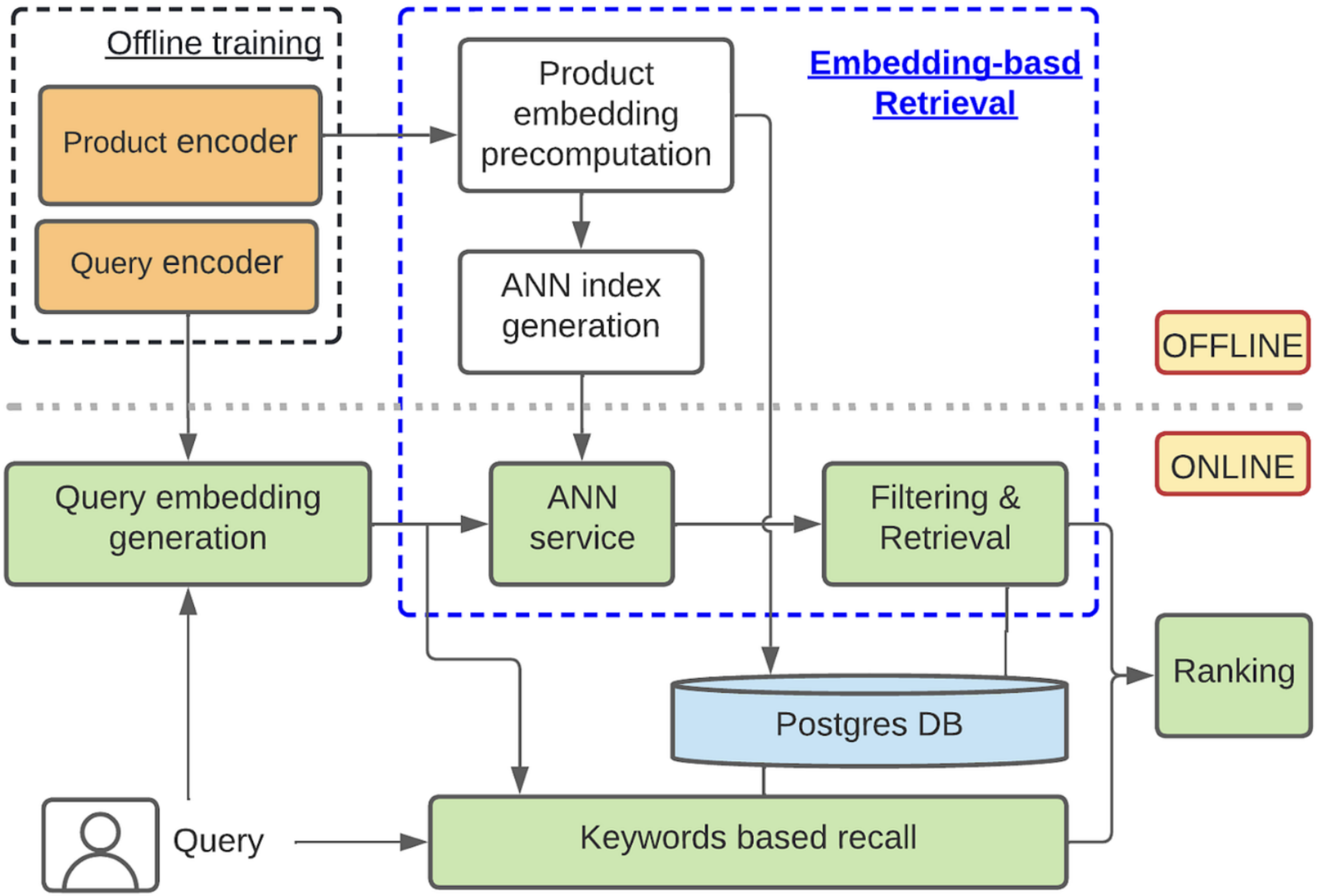


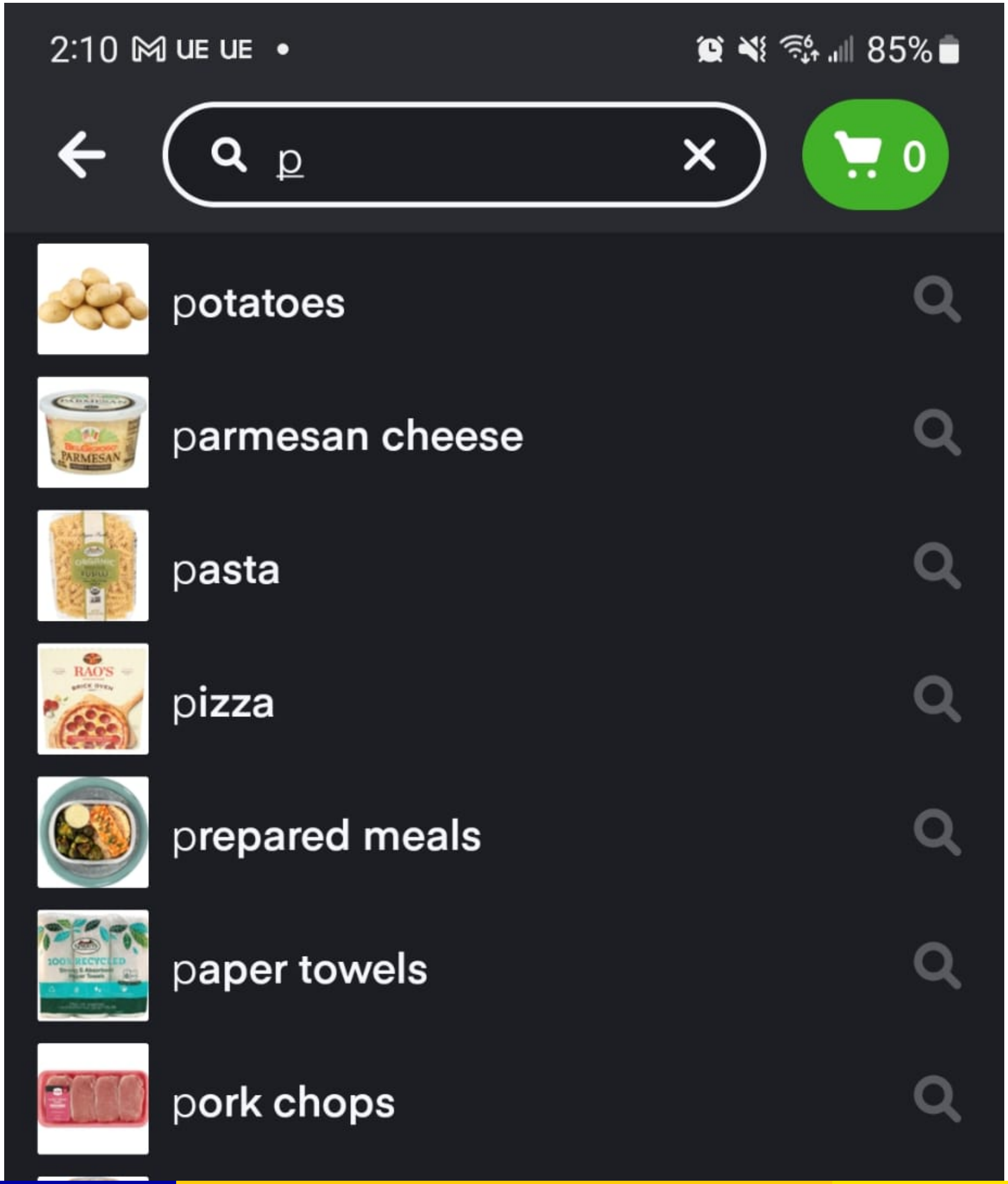
Figure 7. ITEMS system architecture.

Breakouts Time #2

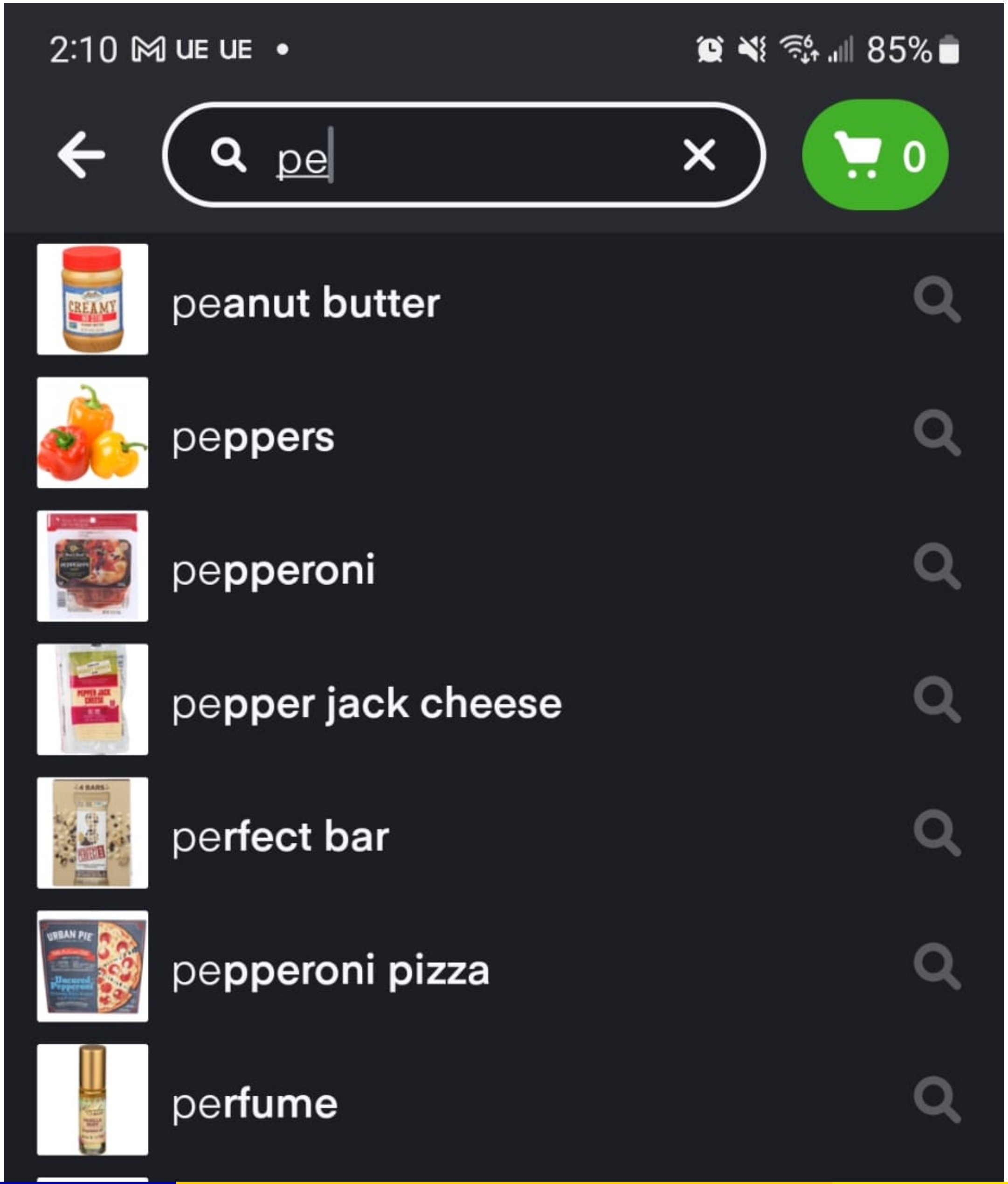
Auto-complete — 5 mins

Let's say you are tasked with building an in-email auto-completion application, which can help complete partial sentences into full 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?

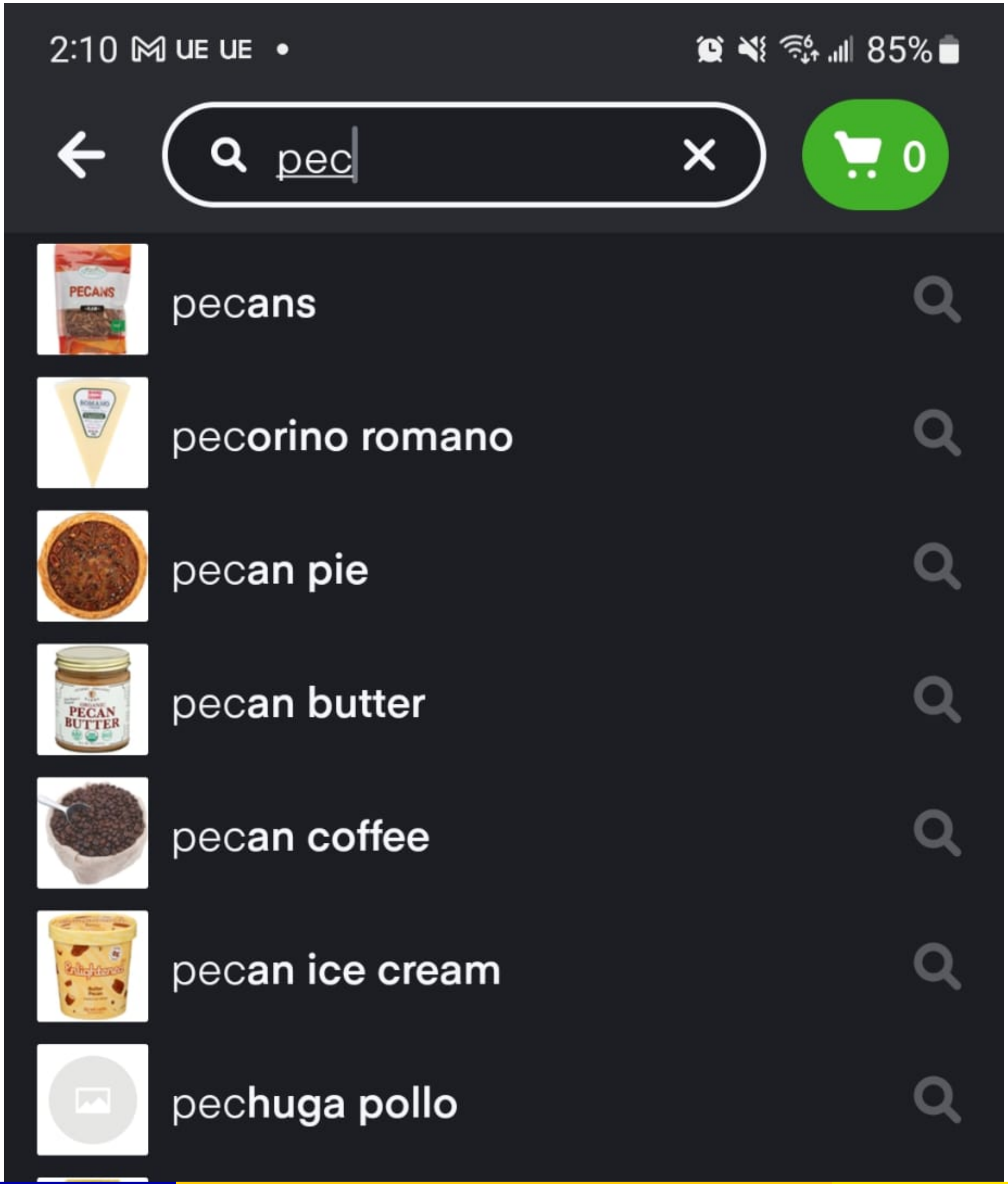
Instacart Auto-Complete and Search Relevance



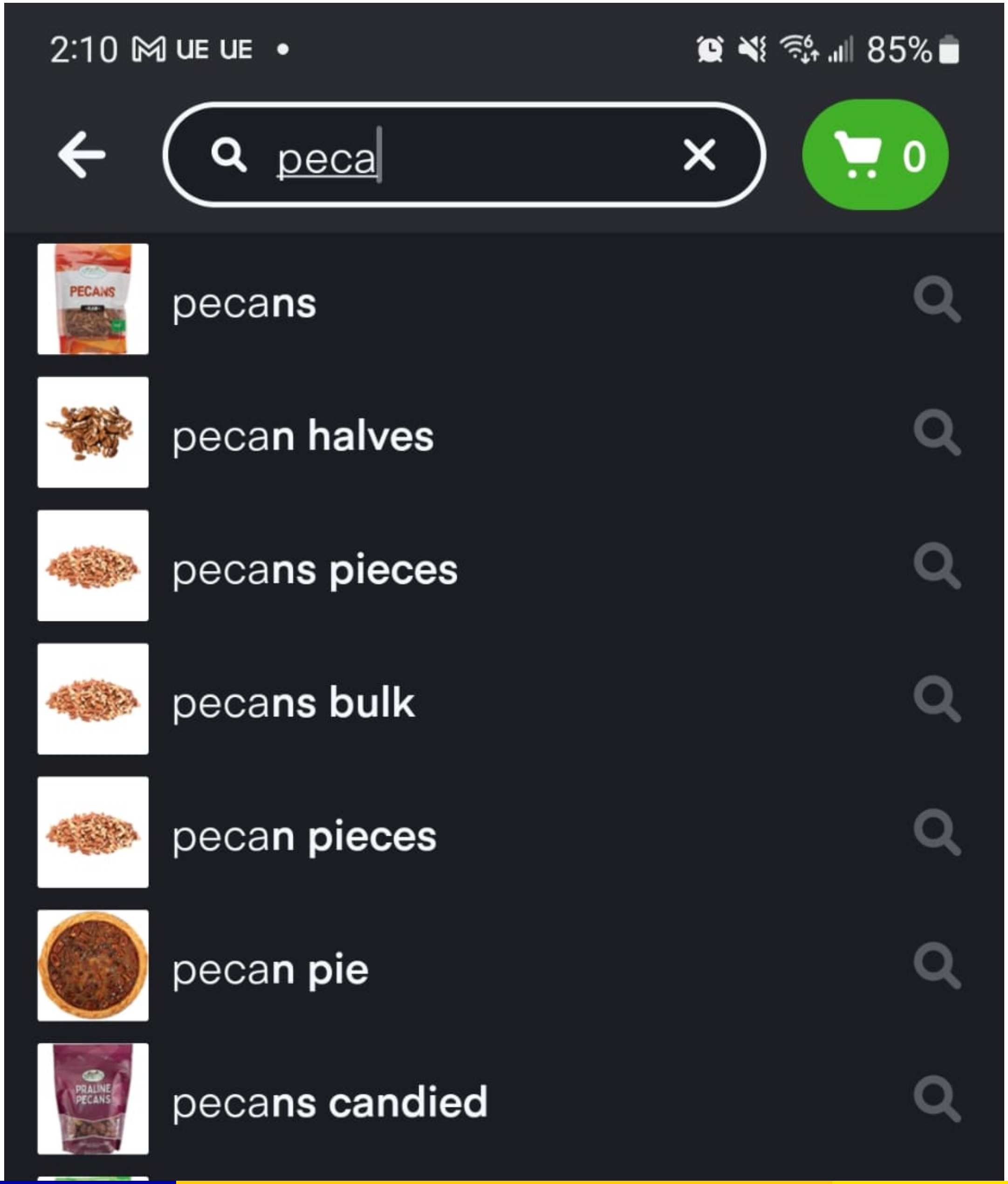
Instacart Auto-Complete



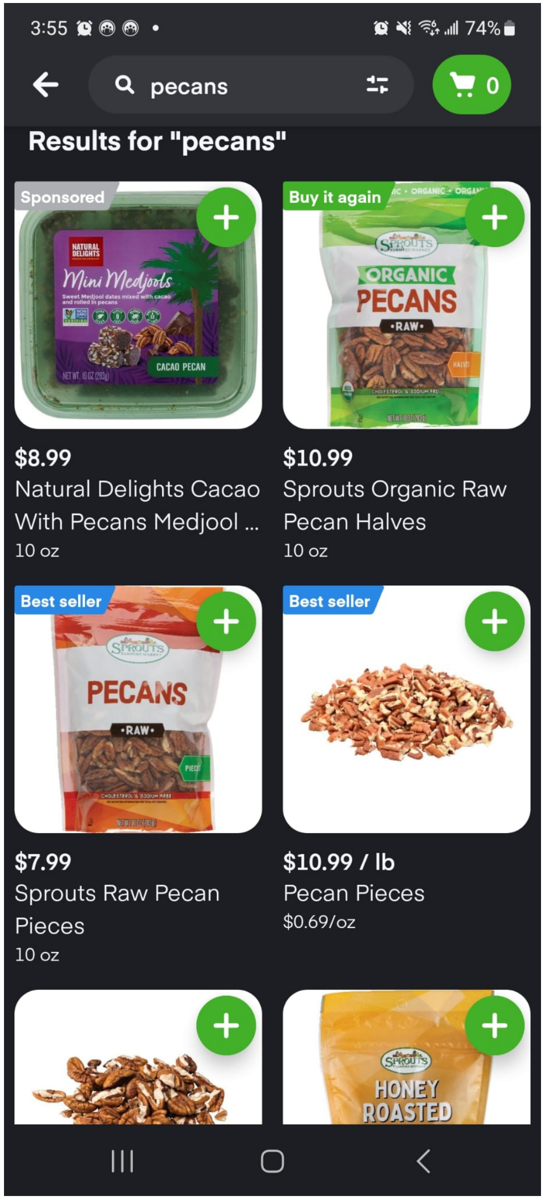
Instacart Auto-Complete



Instacart Auto-Complete



Instacart Auto-Complete and Search Results



Instacart Diversifying Auto-Complete

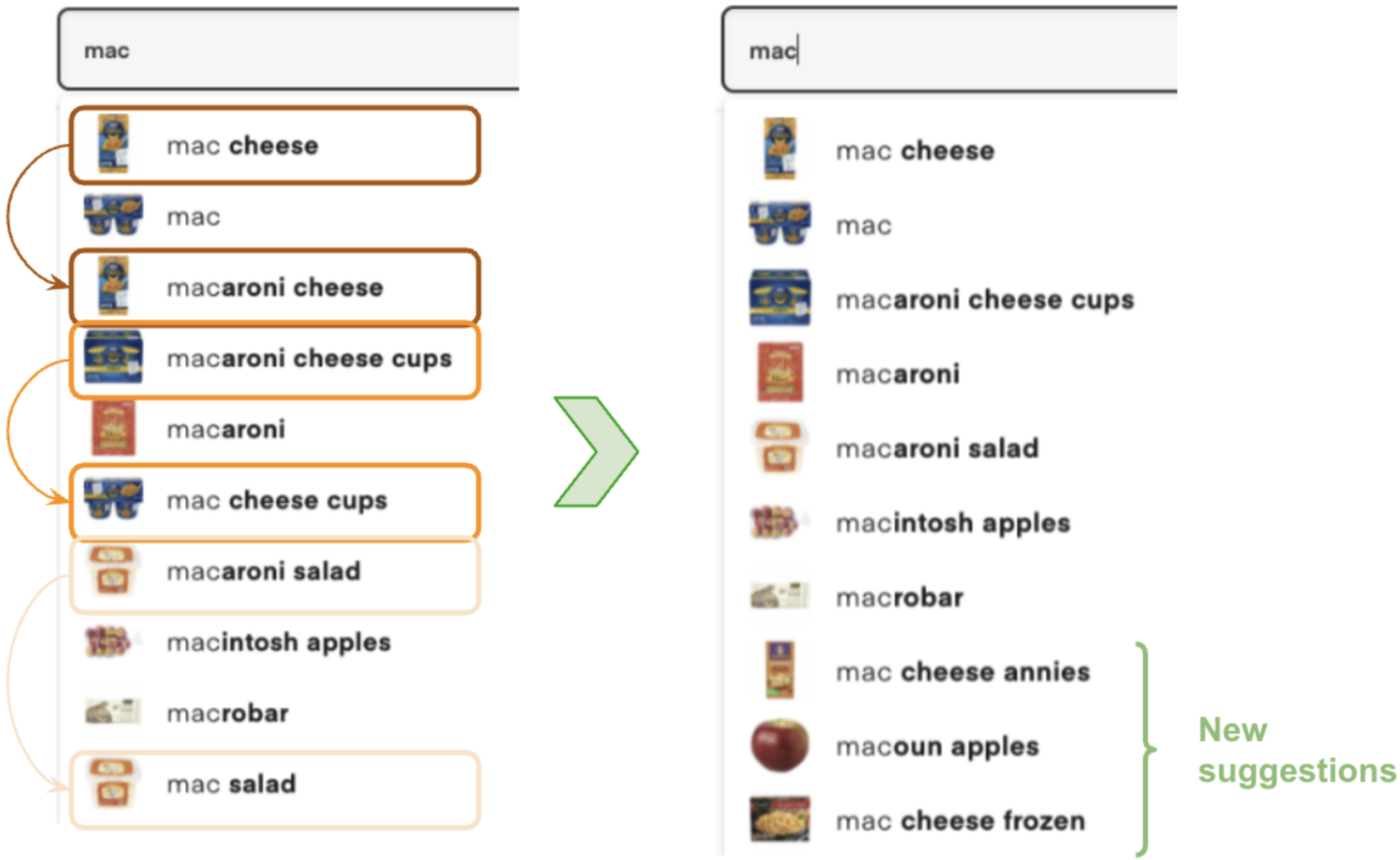


Figure 9. Autocomplete when a customer searches for "mac", before (left) and after (right) semantic deduplication.