Rethinking e-Commerce Search

Haixun Wang / Instacart

Taesik Na

Tejaswi Tenneti

Saurav Manchanda

Min Xie

Chuan Lei

The grocery industry is **massive**

$5.7T

Global Grocery Market

$1.3T

US & Canada
Annual Sales (15%)

By comparison...

$400B

Consumer Electronics

$25B

Books & Magazines

$1.3T

Grocery

We're seeing a strong shift to online

In 2019...

47%

Consumer Electronics

32%

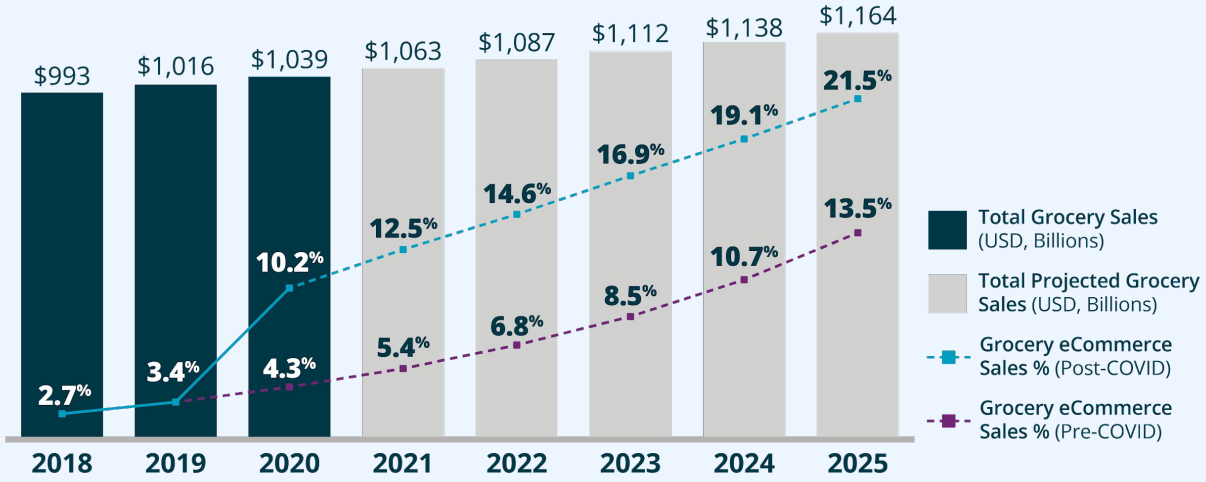
Books & Magazines

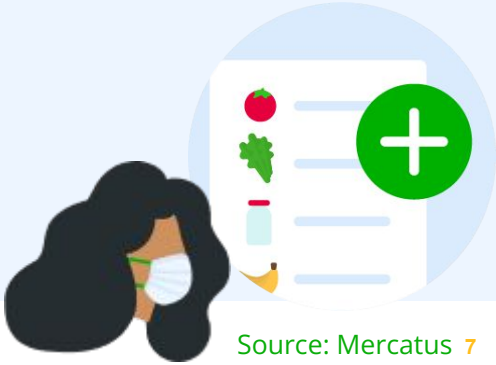
3%

Grocery

...were purchased online

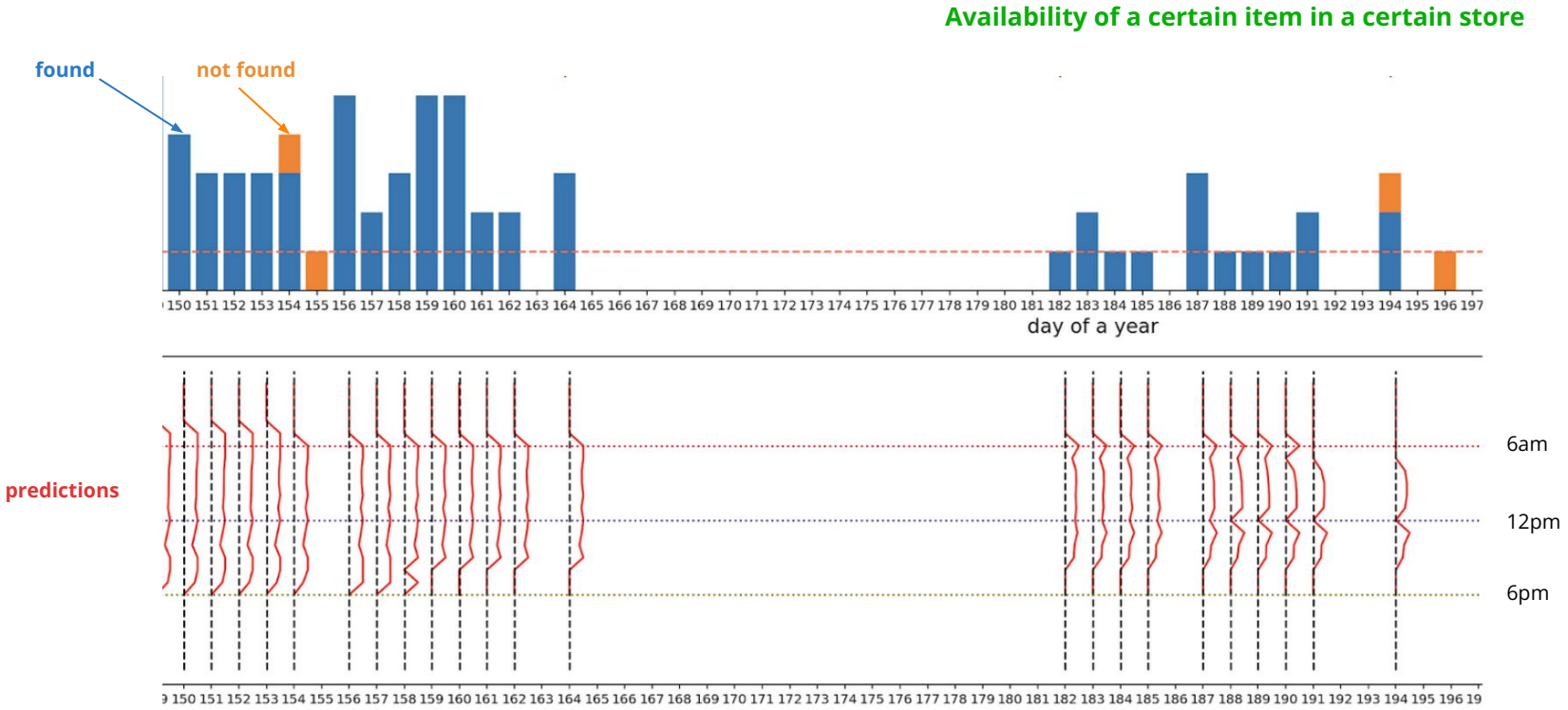
COVID-19 has changed grocery shopping forever



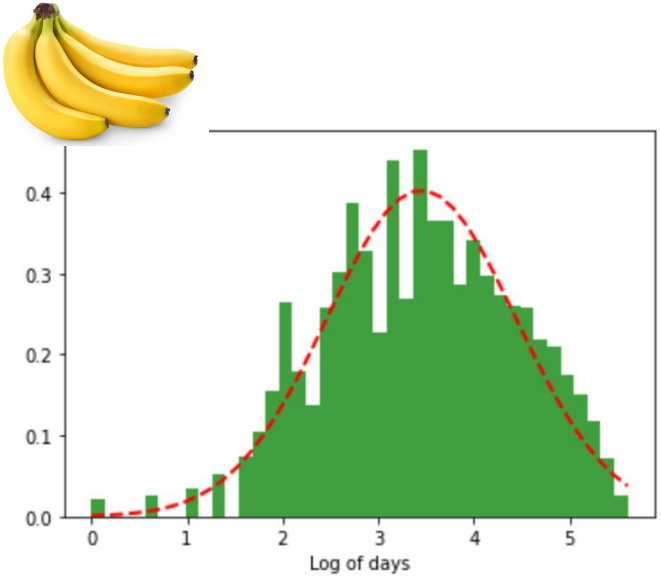


Source: Mercatus

An Example of Technical Challenges

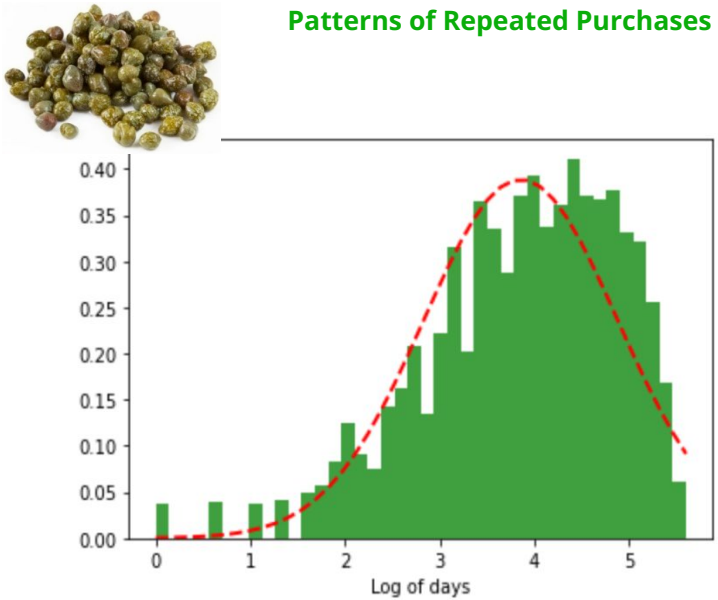


Opportunities



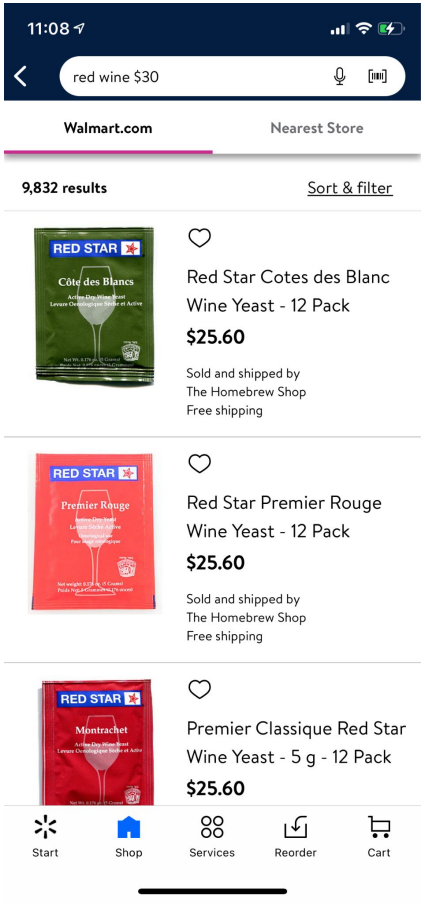
bananas

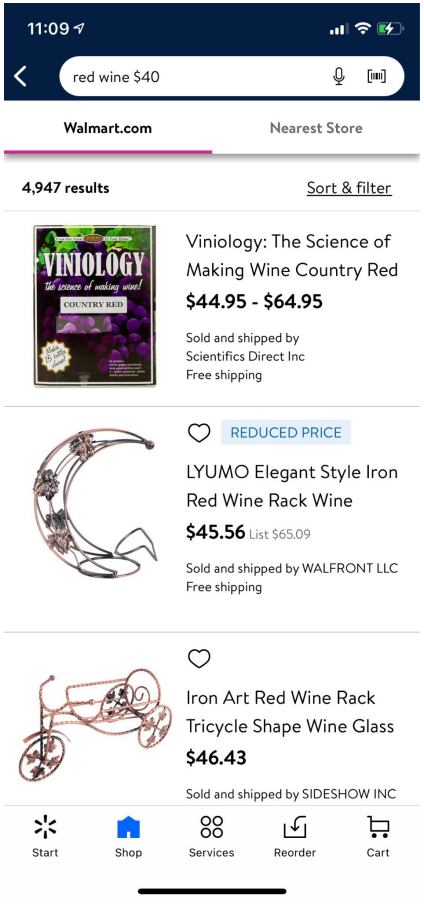
Patterns of Repeated Purchases



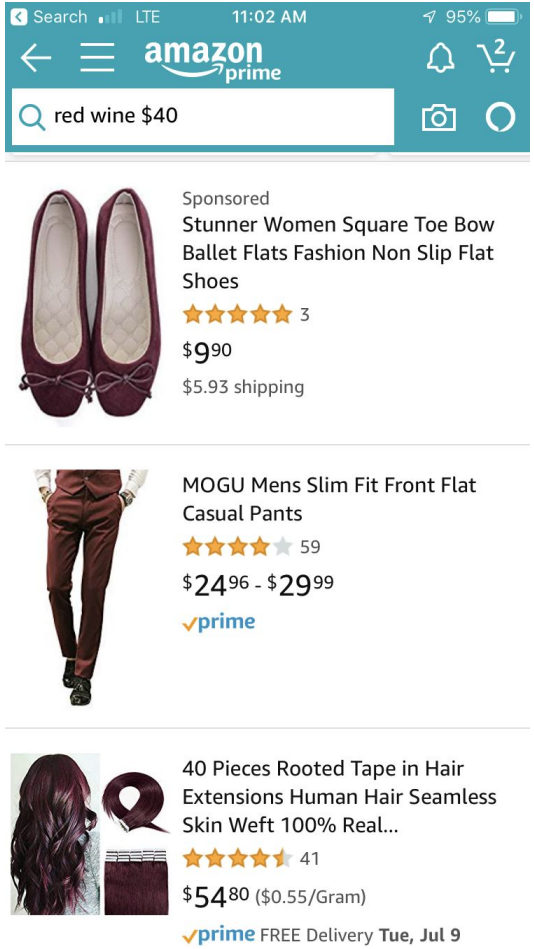
capers

e-Commerce Search @ Walmart

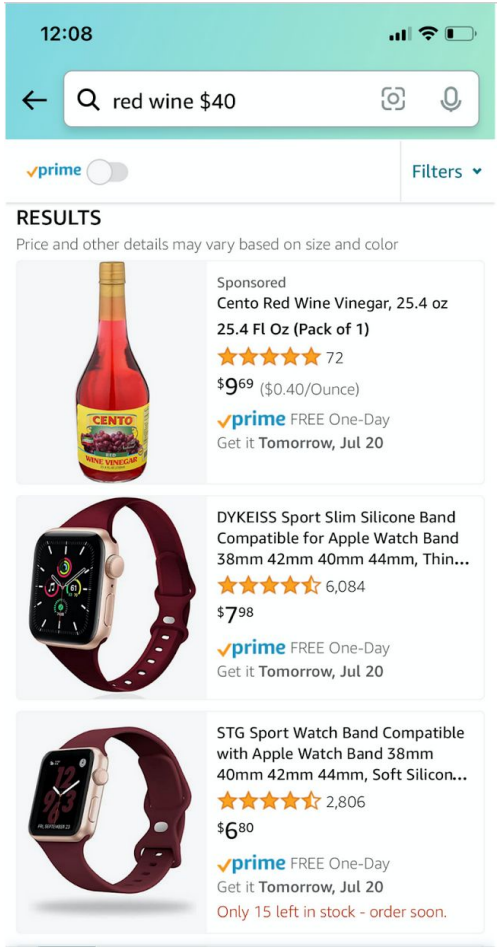




e-Commerce Search @ Amazon

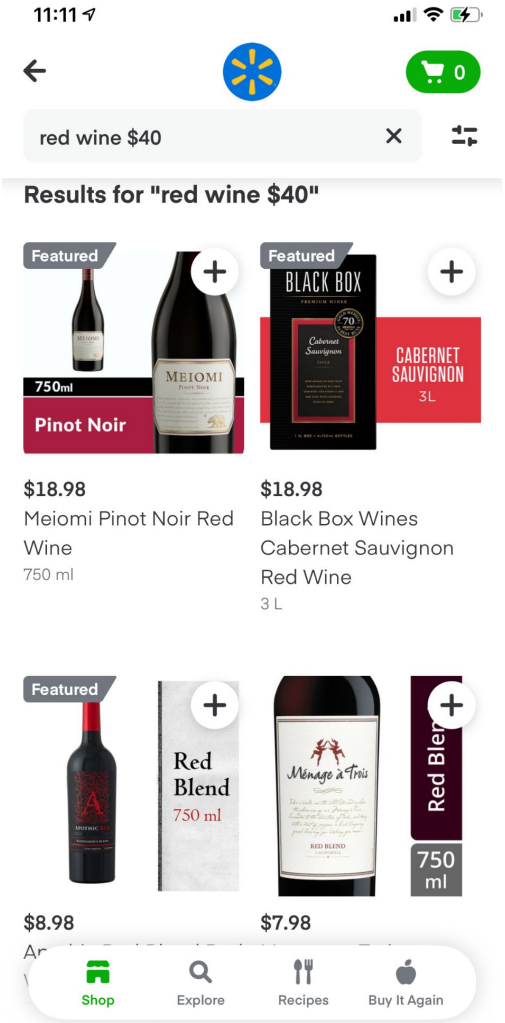


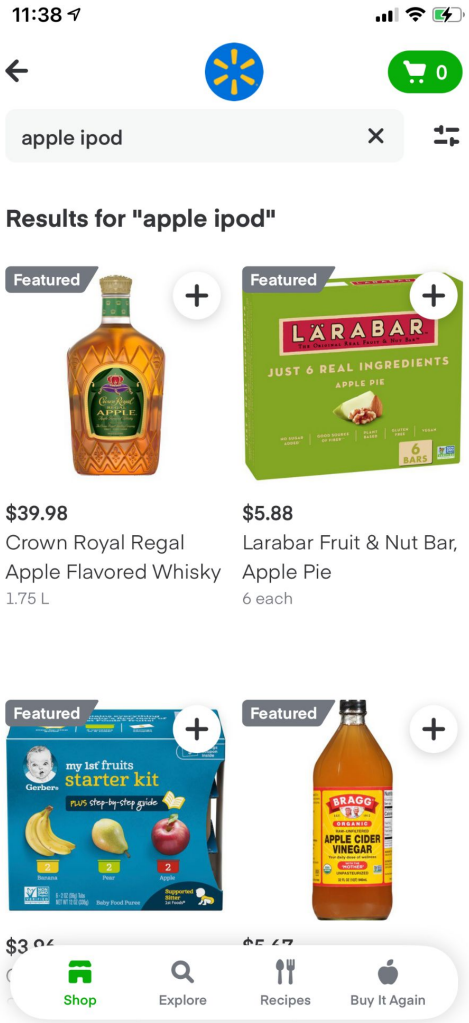
July 7, 2019



July 19, 2021

e-Commerce Search @ Instacart





The future of e-Commerce?

Classical Information Retrieval

Query: blue long sleeve dress

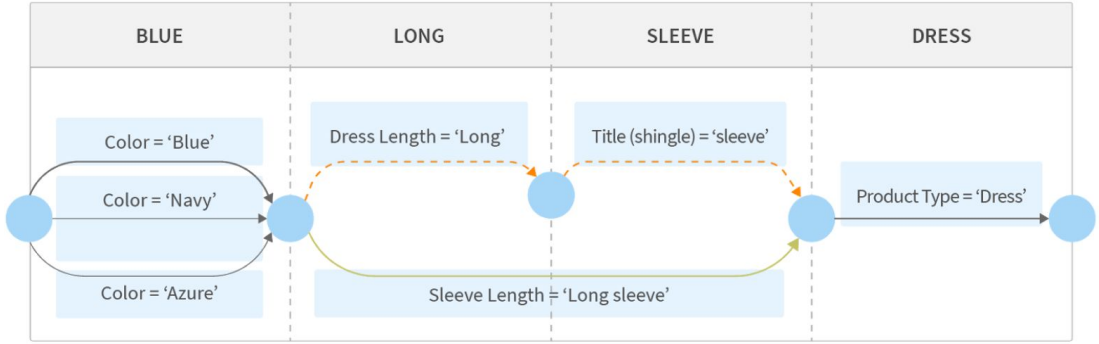
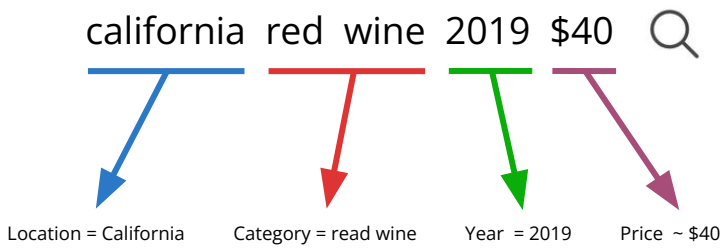


Image from t.ly/qmy0

Two challenges:

- Need many rule-based or ML models to interpret the query
- Need data of heterogeneous types (e.g., catalog, taxonomy, knowledge graph)

Query Understanding



Language Detection (i18n)

Speller

Stemming & Lemmatization

Query Classification

Query Segmentation

Entity Linking

Tagging

Query Rewriting

Query Relaxation

...

Product Catalog & Taxonomy

Retailers adopt different product taxonomies

- Instacart works with 600+ retailers

Google Product Taxonomy

- More than 6,000 categories

Instacart's taxonomy for groceries

- More than 6,000 categories

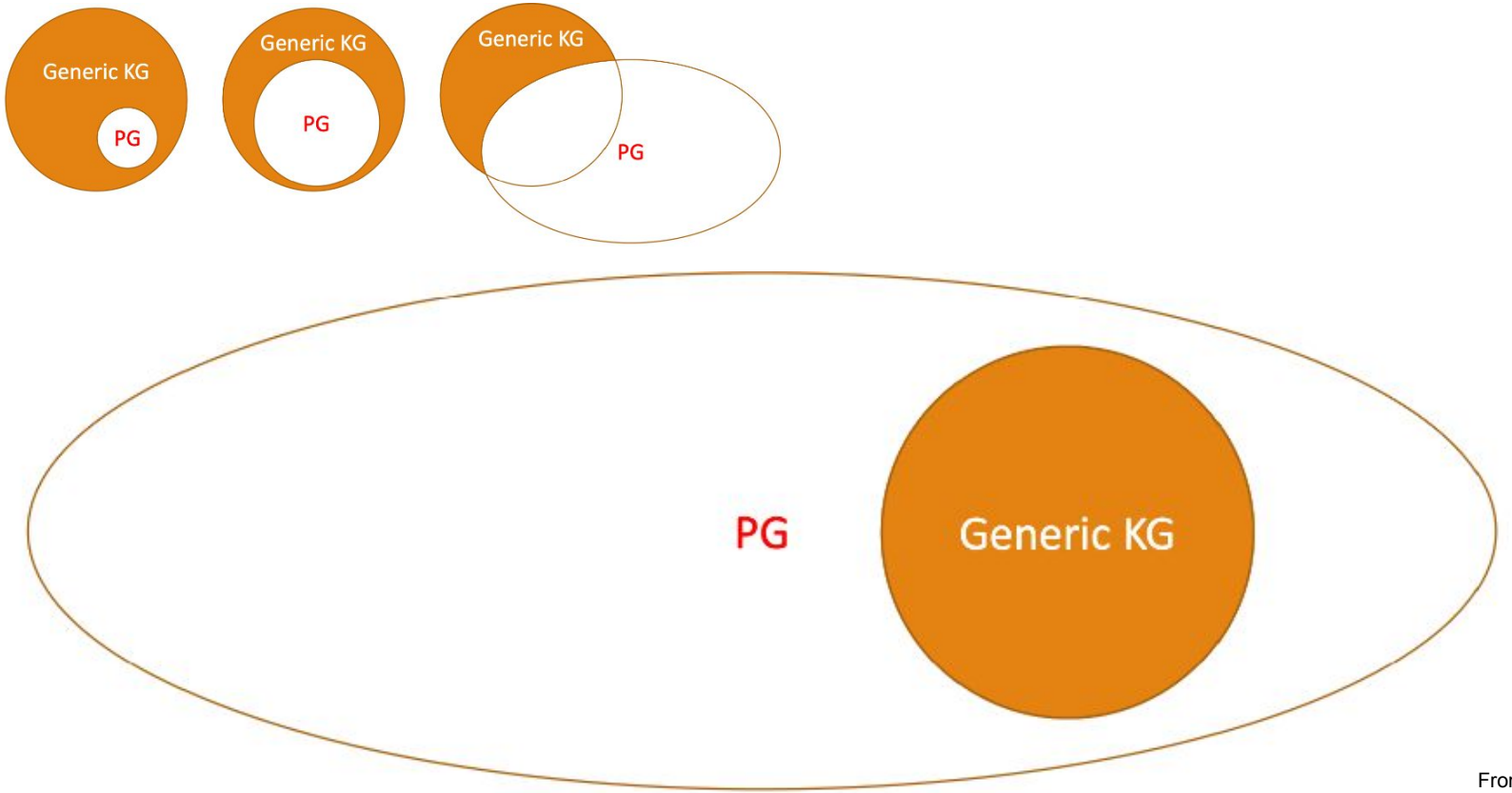
Product Catalog & Taxonomy

When the taxonomy contains thousands of nodes …

… > Floral > Indoor plants

… > Floral > Potted plants

Product Knowledge Graph vs Generic Knowledge Graph



From Luna Dong, Amazon

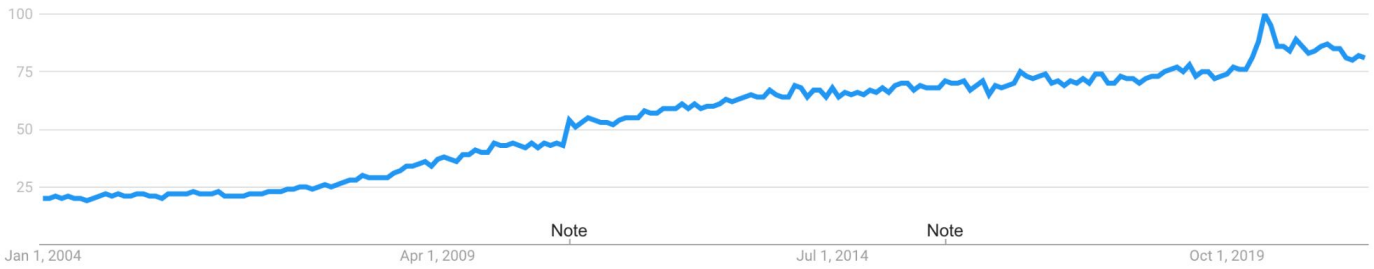
Product Knowledge Graph

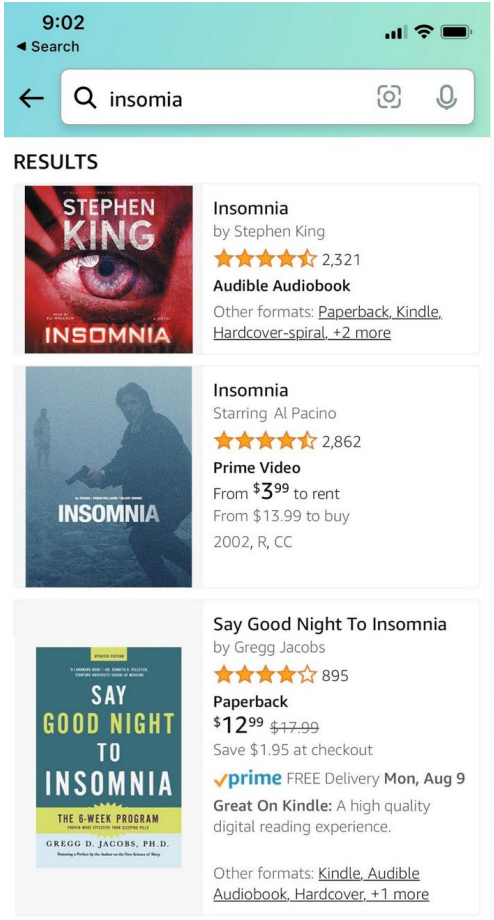
Consider queries:

insomnia, *heartburn*, *how to get rid of a raccoon*

Google trends from 2014:

queries that contain the term “how”





Product Knowledge Graph

Consider queries:

insomnia, *heartburn*, *how to get rid of a raccoon*,

We need knowledge in the form of:

(**key phrase**, relationship, {objects})

For example:

(**heartburn**, medicine-for, {antacids, h2 receptor blockers, proton pump inhibitors})

(**2017 sci-fi movies**, top-10-of, {okja, blade runner 2049, Thor: Ragnarok, Marjorie Prime, …})

(**depression**, treatment-of, {**stay connected**, **exercise**, **healthy diet**, **get sunlight**})

Knowledge on the Web

Ten Kid Friendly Snacks | Save

savethechildren.org/us/charity-st

Incognito

10 Healthy Snacks for Children

Studies show what most parents already know: when kids are well nourished, they perform better in school and are better equipped to fight off disease. But it sometimes seems that pleasing those picky little taste buds is easier said than done. We picked the brains of our in-house nutrition gurus to come up with this list of healthy snack options for kids. These 10 easy-to-make kid friendly treats are so delicious, even the pickiest of eaters will be asking for seconds.

Let the healthy snacking begin!

1. **Go for the Yo (Low-fat Yogurt)**
   Low-fat yogurt is not only high in protein and calcium but also in active cultures that boost the body's immune and digestive systems. Something this good doesn't have to be bland.
   Toss in fresh fruit, add a little low-fat milk, a bit of honey and blend to make a delicious fruit smoothie sure to satisfy any sweet tooth craving. Bonus: freeze your kids' favorite flavors in paper cups and serve as popsicles.
2. **Gain Whole Grains (Whole Grain Snacks)**
   Whole grains are key sources of B vitamins and minerals (iron, magnesium, and selenium), that can keep kids' hearts healthy and reduce the risk of certain cancers and Type-2 diabetes. Replacing even a few refined flour products with whole grains in a child's diet will help provide the dietary fiber necessary to help maintain a healthy body weight.
   A best bet for tummy satisfaction is to pair whole-grain treats with a yummy dip: a whole wheat pretzel with low-fat cheese or yogurt; whole grain crackers with peanut butter or apple sauce; or try whole wheat pita bread with hummus.
3. **Make an Egg-cellent Choice (Eggs)**
   We're bringing breakfast back. Protein-packed eggs are not just a great way to start the day, but also a low-calorie way to refuel in the afternoon.
   Fix them sunny side up or scrambled (go easy on the oil) and serve with whole grain toast and jam. Or opt for a fun, hard-boiled version, slicing eggs in half, adding a cheese flag with a toothpick and sailing your way through the afternoon with an egg boat.
4. **Eat the Rainbow (Fruit)**

11 Best Barbecue Recipes | Po

food.ndtv.com/lists/10-best-barbec...

Incognito

NDTV FOOD Eng

Search Recipes

11 Best Barbecue Recipes | Popular Barbecue Recipes

Barbecue is probably the world's oldest cooking method. We've rounded up our 11 best barbecue recipes that you can try at home on a bonfire night with family and friends.

NDTV Food | Updated: March 17, 2020 13:44 IST

Barbecue recipes you can try at home.

Thinkstock

"It is better to have burnt and lost, than never to have barbecued at all" - William Shakespeare

Barbecue Recipes-Barbecue is probably the world's oldest cooking method. It has come a long way from the traditional pit BBQ that originated in the Caribbean to the great Indian tandoor. Australians have taken to the 'barbie' with great gusto. It is a fun and fiery way to eat hearty and stay snug, perfect on a nippy night or for a breezy brunch. For your next BBQ party, we show you how to do it right.

Data Integration

Structured Data

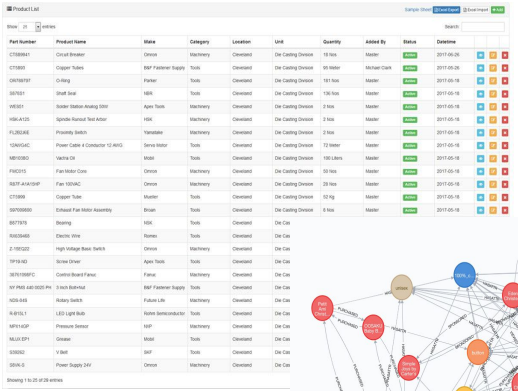
- Catalog (inventory) data
- Transaction data
- …

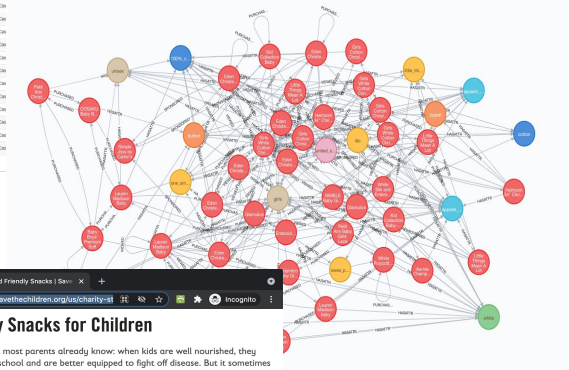
Semi-Structured Data (trees and graphs)

- Taxonomy, Ontology, Knowledge Graph
- …

Unstructured Data

- Customer review
- Web page
- …





Ten Kid Friendly Snacks | Save

savethechildren.org/us/charity-st

Incognito

10 Healthy Snacks for Children

Studies show what most parents already know: when kids are well nourished, they perform better in school and are better equipped to fight off disease. But it sometimes seems that pleasing those picky little taste buds is easier said than done. We picked the brains of our in-house nutrition gurus to come up with this list of healthy snack options for kids. These 10 easy-to-make kid friendly treats are so delicious, even the pickiest of eaters will be asking for seconds.

Let the healthy snacking begin!

1. **Go for the Yo (Low-fat Yogurt)**
   Low-fat yogurt is not only high in protein and calcium but also in active cultures that boost the body's immune and digestive systems. Something this good doesn't have to be bland.
   Toss in fresh fruit, add a little low-fat milk, a bit of honey and blend to make a delicious fruit smoothie sure to satisfy any sweet tooth craving. Bonus: freeze your kids' favorite flavors in paper cups and serve as popsicles.
2. **Gain Whole Grains (Whole Grain Snacks)**
   Whole grains are key sources of B vitamins and minerals (iron, magnesium, and selenium), that can keep kids' hearts healthy and reduce the risk of certain cancers and Type-2 diabetes. Replacing even a few refined flour products with whole grains in a child's diet will help provide the dietary fiber necessary to help maintain a healthy body weight.
   A best bet for tummy satisfaction is to pair whole-grain treats with a yummy dip: a whole wheat pretzel with low-fat cheese or yogurt; whole grain crackers with peanut butter or apple sauce; or try whole wheat pita bread with hummus.
3. **Make an Egg-cellent Choice (Eggs)**
   We're bringing breakfast back. Protein-packed eggs are not just a great way to start the day, but also a low-calorie way to refuel in the afternoon.
   Fix them sunny side up or scrambled (go easy on the oil) and serve with whole grain toast and jam. Or opt for a fun, hard-boiled version, slicing eggs in half, adding a cheese flag with a toothpick and sailing your way through the afternoon with an egg boat.
4. **Eat the Rainbow (Fruit)**

Summary: Issues of e-Commerce Search

- Classical IR uses inverted index that is “term” based. No semantics.
- To support semantic matching, we perform query rewriting at many levels.
- To support tasks such as query rewriting, we develop many individual ML models.
- To do a better job in understanding queries, we must incorporate heterogeneous type of data, such as web pages.

Neural IR

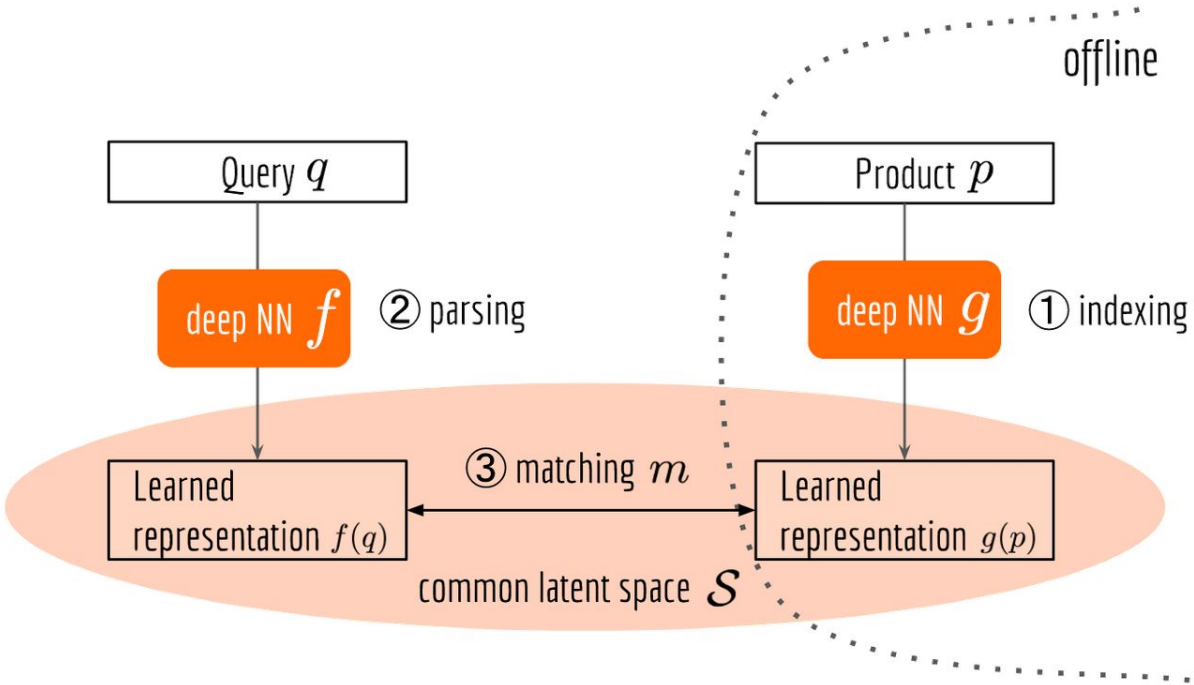
Neural Re-ranking Models

- An extension to the Learning to Rank mechanism.
- Use neural network-based models to score or rank documents.

Representation Learning

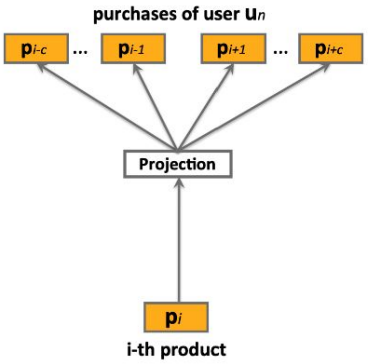
- Encode queries and documents into distributional representations.
- Use k-nearest neighbor search (ANN) to find relevant documents.

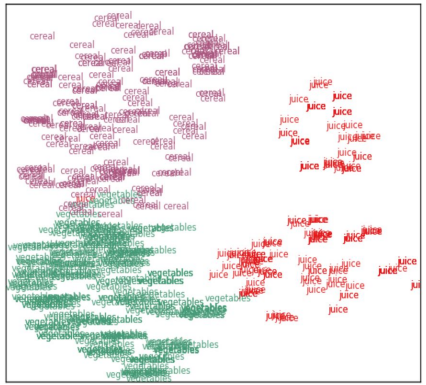
Neural Information Retrieval

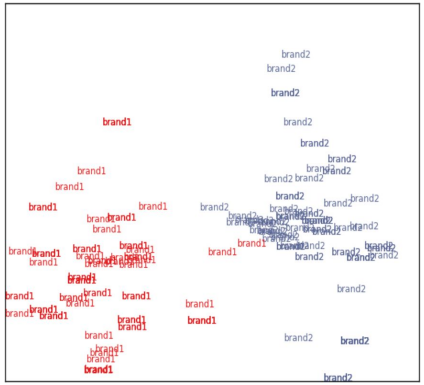


Han Xiao: BERT as a Service

Representations of Products







Prod2vec (Grbovic et al), Prod2BERT (Bianchi et al), E-BERT (Zhang et al)

Applications beyond Search: Recommendation, Intent Prediction, etc.

Neural e-Commerce Search

End-to-End, Personalized

Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-commerce Search via Embedding Learning

Han Zhang[1†], Songlin Wang[1†], Kang Zhang[1], Zhiling Tang[1], Yunjiang Jiang[2], Yun Xiao[2], Weipeng Yan[1,2], Wen-Yun Yang[2*]
[1]JD.com, Beijing, China
[2]JD.com Silicon Valley Research Center, Mountain View, CA, United States
{zhanghan33, wangsonglin3, zhangkang1, tangzhiling, yunjiang.jiang, xiaoyun1, paul.yan, wemyun.yang}@jd.com

ABSTRACT

Nowadays e-commerce search has become an integral part of many people's shopping routines. Two critical challenges stay in today's e-commerce search: how to retrieve items that are semantically relevant but not exact matching to query terms, and how to retrieve items that are more personalized to different users for the same search query. In this paper, we present a *novel* approach called DPSR, which stands for Deep Personalized and Semantic Retrieval, to tackle this problem. Explicitly, we share *our* design decisions on how to architect a retrieval system so as to serve industry-scale traffic efficiently and how to train a model so as to learn query and item semantics accurately. *Based on offline evaluations and online* A/B test with live traffics, we show that DPSR model outperforms existing models, and DPSR system can retrieve more personalized and semantically relevant items to significantly improve users' search experience by +1.29% conversion rate, especially for long tail queries by +18.03%. As a result, our DPSR system has been successfully deployed into JD.com's search production since 2019.

CCS CONCEPTS

•**Computing methodologies** → **Neural networks;**•**Information systems** → **Information retrieval;**

KEYWORDS

Search; Semantic matching; Neural networks

1 INTRODUCTION

Over the recent decades, online shopping platforms (e.g., Ebay, Walmart, Amazon, Tmall, Taobao and JD) have become increasingly popular in people's daily life. E-commerce search, which helps users to find what they need from billions of products, is an essential part of those platforms, contributing to the largest percentage of transactions among all channels [18, 27, 28]. For instance, the top e-commerce platforms in China, e.g., Tmall, Taobao and JD, serve hundreds of million active users with gross merchandise volume of hundreds of billion US dollar. In this paper, we will focus on the immense impact that deep learning has recently had on the e-commerce search system. At a glance, Figure 1 illustrates the user interface for searching on JD's mobile app.

† Both authors contributed equally.
* Corresponding author.

Figure 1: Search interface on JD's e-commerce mobile app.

1.1 Three Components of Search System

Figure 2 illustrates a typical e-commerce search system with three components, query processing, candidate retrieval, and ranking.

Query Processing rewrites a query (e.g., "cellphone for grandpa") into a term based presentation (e.g., [TERM cellphone] AND [TERM grandpa]) that can be processed by downstream components. This stage typically includes tokenization, spelling correction, query expansion and rewriting.

Candidate Retrieval uses offline built inverted indexes, to efficiently retrieve candidate items based on term matching. This step greatly reduces the number of items from billions to hundreds of thousands, in order to make the fine ranking feasible.

Ranking orders the retrieved candidates based on factors, such as relevance, predicted conversion ratio, etc. A production system may have cascading ranking steps, which sequentially apply simpler to more complex ranking functions from upstream to downstream.

In this paper, we focus solely on the *candidate retrieval* stage to achieve more personalized and semantic search results, since this stage contributes the most bad cases in our search production. Based on our analysis, around 20% dissatisfaction cases of search traffic of JD.com, one of the largest e-commerce search engine in the world, can be attributed to the failure of this stage. How to deal

JD.com

End-to-End, Session based

End-to-End Neural Ranking for eCommerce Product Search

An application of task models and textual embeddings

Eliot P. Brenner*
Jet.com/Walmart Labs
Hoboken, NJ
eliot.brenner@jet.com

Jun (Raymond) Zhao
Jet.com/Walmart Labs
Hoboken, NJ
raymond@jet.com

Aliasgar Kutiyanawala
Jet.com/Walmart Labs
Hoboken, NJ
aliasgar@jet.com

Zheng (John) Yan
Jet.com/Walmart Labs
Hoboken, NJ
john@jet.com

ABSTRACT

We consider the problem of retrieving and ranking items in an eCommerce catalog, often called *SKUs*, in order of relevance to a user-issued query. The input data for the ranking are the texts of the queries and textual fields of the SKUs indexed in the catalog. We review the ways in which this problem both resembles and differs from the problems of information retrieval (IR) in the context of web search, which is the context typically assumed in the IR literature. The differences between the product-search problem and the IR problem of web search necessitate a different approach in terms of both models and datasets. We first review the recent state-of-the-art models for web search IR, focusing on the CLSM of [20] as a representative of one type, which we call the *distributed* type, and the kernel pooling model of [26], as a representative of another type, which we call the *local-interaction* type. The different types of relevance models developed for IR have complementary advantages and disadvantages when applied to eCommerce product search. Further, we explain why the conventional methods for dataset construction employed in the IR literature fail to produce data which suffices for training or evaluation of models for eCommerce product search. We explain how *our own* approach, applying *task modeling* techniques to the click-through logs of an eCommerce site, enables the construction of a large-scale dataset for training and robust benchmarking of relevance models. Our experiments consist of applying several of the models from the IR literature to our own dataset. Empirically, we have established that, when applied to our dataset, certain models of *local-interaction* type reduce ranking errors by one-third compared to the baseline system (tf–idf). Applied to our dataset, the *distributed* models fail to outperform the baseline. As a basis for a deployed system, the *distributed* models have several advantages, computationally, over the *local-interaction* models. This motivates an ongoing program of work, which we outline at the conclusion of the paper.

*Corresponding Author.

CCS CONCEPTS

• **Information systems** → **Query representation; Probabilistic retrieval models; Relevance assessment;** *Task models; Enterprise search;* • **Computing methodologies** → Neural networks; Bayesian network models;

KEYWORDS

Ranking, Neural IR, Kernel Pooling, Relevance Model, Embedding, eCommerce, Product Search, Click Models, Task Models

ACM Reference Format:
Eliot P. Brenner, Jun (Raymond) Zhao, Aliasgar Kutiyanawala, and Zheng (John) Yan. 2018. End-to-End Neural Ranking for eCommerce Product Search: An application of task models and textual embeddings. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2018 eCom)*, ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Currently deployed systems for eCommerce product search tend to use inverted-index based retrieval, as implemented in Elasticsearch [4] or Solr [21]. For ranking, these systems typically use *legacy* relevance functions such as tf–idf [25] or Okpai BM25 [18], as implemented in these search systems. Such relevance functions are based on exact ("hard") matches of tokens, rather than semantic ("soft") matches, are insensitive to word order, and have hard-coded, rather than learned weights. On the one hand, their simplicity makes legacy relevance functions scalable and easy-to-implement. One the other hand, they are found to be inadequate in practice for fine-grained ranking of search results. Typically, in order to achieve rankings of search results that are acceptable for presentation to the user, eCommerce sites overlay on top of the legacy relevance function score, a variety of handcrafted filters (using structured data fields) as well as hard-coded rules for specific queries. In some cases, eCommerce sites are able to develop intricate and specialized proprietary NLP systems, referred to as Query-SKU Understanding (QSU) Systems, for analyzing and matching relevant SKUs to queries. QSU systems, while potentially very effective at addressing the shortcomings of legacy relevance scores, require a some degree of domain-specific knowledge to engineer [8]. Because of concept drift, the maintenance of QSU systems demands a long-term commitment of analyst and programmer labor. As a result of

Walmart

Addressing any need of a customer?

Consider queries:

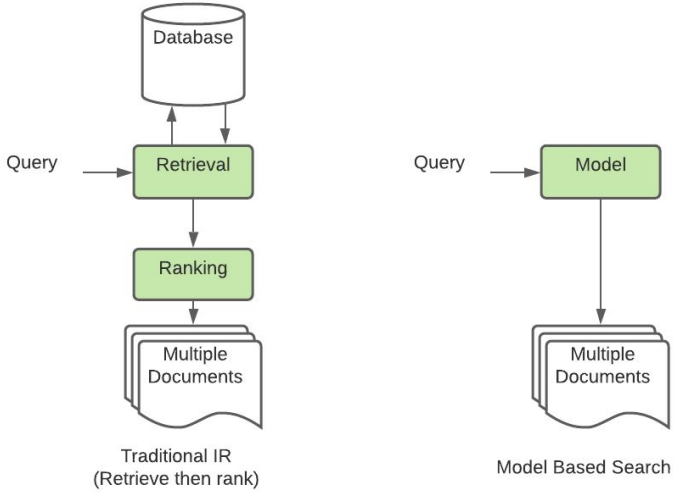
- *insomnia*
- *heartburn*
- *how to get rid of a raccoon*

It's more a data integration problem than a query-product semantic matching problem.

End-to-End IR

Question:

Can we train one ML model that returns related documents/products for a give query?



Metzler et al, Rethinking Search

Model based Search

OPINION PAPER

Rethinking Search: Making Domain Experts out of Dilettantes*

Donald Metzler, Google Research, *metzler@google.com*
Yi Tay, Google Research, *yitay@google.com*
Dara Bahri, Google Research, *dbahri@google.com*
Marc Najork, Google Research, *najork@google.com*

Abstract

When experiencing an information need, users want to engage with a domain expert, but often turn to an information retrieval system, such as a search engine, instead. Classical information retrieval systems do not answer information needs directly, but instead provide references to (hopefully authoritative) answers. Successful question answering systems offer a limited corpus created on-demand by human experts, which is neither timely nor scalable. Pre-trained language models, by contrast, are capable of directly generating prose that may be responsive to an information need, but at present they are dilettantes rather than domain experts – they do not have a true understanding of the world, they are prone to hallucinating, and crucially they are incapable of justifying their utterances by referring to supporting documents in the corpus they were trained over. This paper examines how ideas from classical information retrieval and pre-trained language models can be synthesized and evolved into systems that truly deliver on the promise of domain expert advice.

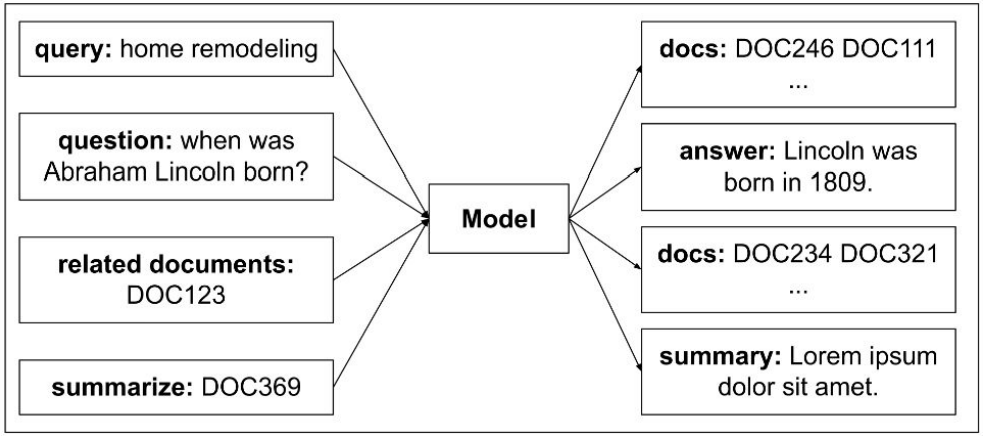
1 Introduction

Given an information need, users often turn to search engines for help. Such systems point them in the direction of one or more relevant items from a corpus. This is appropriate for navigational and transactional intents (e.g. home page finding or online shopping) but typically less ideal for informational needs, where users seek answers to questions they may have [Broder, 2002]. Classical information retrieval (IR) systems do not directly answer information needs, but instead provide references to (hopefully authoritative) content.

The very fact that ranking is a critical component of this paradigm is a symptom of the retrieval system providing users a selection of potential answers, which induces a rather significant cognitive burden on the user. The desire to return *answers* instead of ranked lists of results was one of the motivating factors for developing question answering systems. While there has been a great deal

*Disclaimer: This is a research proposal, not the roadmap for any Google product or service.

ACM SIGIR Forum 1 Vol. 55 No. 1 - June 2021



Model based General Search vs e-Commerce Search

Challenges:

- The document space is huge (Google indexes 30 trillion pages)
- Update is costly (It is estimated that Google crawls 25 billion pages every day)

e-Commerce Search:

- It is more like search against the database
- The product space is much smaller (prod2vec considers ~2 million products)
- For ads, the featured product space is even smaller (usually << 1 million)

Neural Databases

Neural Databases

James Thorne
University of Cambridge
Facebook AI
jt719@cam.ac.uk

Majid Yazdani
Facebook AI
myazdani@fb.com

Marzieh Saeidi
Facebook AI
marzieh@fb.com

Fabrizio Silvestri
Facebook AI
fsilvestri@fb.com

Sebastian Riedel
Facebook AI
sriedel@fb.com

Alon Halevy
Facebook AI
ayh@fb.com

ABSTRACT

In recent years, neural networks have shown impressive performance gains on long-standing AI problems, and in particular, answering queries from natural language text. These advances raise the question of whether they can be extended to a point where we can relax the fundamental assumption of database management, namely, that our data is represented as fields of a pre-defined schema.

This paper presents a first step in answering that question. We describe NeuralDB, a database system with no pre-defined schema, in which updates and queries are given in natural language. We develop query processing techniques that build on the primitives offered by the state of the art Natural Language Processing methods.

We begin by demonstrating that at the core, recent NLP transformers, powered by pre-trained language models, can answer select-project-join queries if they are given the exact set of relevant facts. However, they cannot scale to non-trivial databases and cannot perform aggregation queries. Based on these findings, we describe a NeuralDB architecture that runs multiple Neural SPJ operators in parallel, each with a set of database sentences that can produce one of the answers to the query. The result of these operators is fed to an aggregation operator if needed. We describe an algorithm that learns how to create the appropriate sets of facts to be fed into each of the Neural SPJ operators. Importantly, this algorithm can be trained by the Neural SPJ operator itself. We experimentally validate the accuracy of NeuralDB and its components, showing that we can answer queries over thousands of sentences with very high accuracy.

PVLDB Reference Format:
James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. Neural Databases. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Availability Tag:
The source code of this research paper has been made publicly available at http://vldb.org/pvldb/format_vol:14.html.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

1 INTRODUCTION

In recent years, neural networks have shown impressive performance gains on long-standing AI problems, such as natural language understanding, speech recognition, and computer vision. Based on these successes, researchers have considered the application of neural nets to data management problems, including learning indices [21], query optimization and entity matching [25, 29]. In applying neural nets to data management, research has so far assumed that the data was modeled by a database schema.

The success of neural networks in processing unstructured data such as natural language and images raises the question of whether their use can be extended to a point where we can relax the fundamental assumption of database management, which is that the data we process is represented as fields of a pre-defined schema. What if, instead, data and queries can be represented as short natural language sentences, and queries can be answered from these sentences? This paper presents a first step in answering that question. We describe NeuralDB, a database system in which updates and queries are given in natural language. The query processor of a NeuralDB builds on the primitives that are offered by the state of the art Natural Language Processing (NLP) techniques. Figure 1 shows example facts and queries that NeuralDB can answer.

Realizing the vision of NeuralDB will offer several benefits that database systems have struggled to support for decades. The first, and most important benefit is that a NeuralDB, by definition, has no pre-defined schema. Therefore, the scope of the database does not need to be defined in advance and any data that becomes relevant as the application is used can be stored and queried. The second benefit is that updates and queries can be posed in a variety of natural language forms, as is convenient to any user. In contrast, a traditional database query needs to be based on the database schema. A third benefit comes from the fact that the NeuralDB is based on a pre-trained language model that already contains a lot of knowledge. For example, the fact that London is in the UK is already encoded in the language model. Hence, a query asking who lives in the UK can retrieve people who are known to live in London without having to explicitly specify an additional join. Furthermore, using the same paradigm, we can endow the NeuralDB with more domain knowledge by extending the pre-training corpus to that domain.

By nature, a NeuralDB is not meant to provide the same correctness guarantees of a traditional database system, i.e., that the answers returned for a query satisfy the precise binary semantics of the query language. Hence, NeuralDBs should not be considered

Facts: (4 of 50 shown)

Nicholas lives in Washington D.C. with Sheryl.
Sheryl is Nicholas's spouse.
Teuvo was born in 1912 in Ruskala.
In 1978, Sheryl's mother gave birth to her in Huntsville.

Queries:

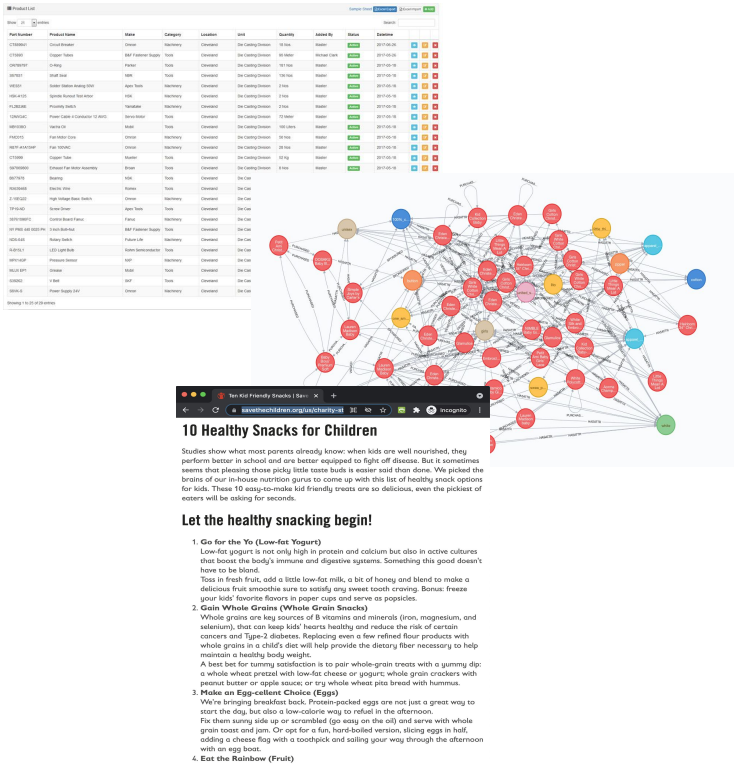
Does Nicholas's spouse live in Washington D.C.?
(Boolean Join) ⟶ TRUE

Who is Sheryl's husband?
(Lookup) ⟶ Nicholas

Who is the oldest person in the database?
(Max) ⟶ Teuvo

Who is Sheryl's mother?
(Lookup) ⟶ NULL

Unifying Heterogeneous Data to Text



Unifying Heterogeneous Data to Text

Special ID token for each Product

- Create a special token for each product, and use the token in text to represent the product
- e.g., [P123] could refer to a specific product

Alternative Approach

- [PStart] Lucerne Milk with Reduced Fat [PEnd]

Unifying Heterogeneous Data to Text

Structure Data are basically relational tables with columns or attributes. We assume each record is about a product.

We create a set of multiple natural language templates for each attribute. These templates allow us to generate descriptions of the product in a natural language.

For example:

- Product name:
  - (ex) [P123] is Lucerne Milk with Reduced Fat.
  - (ex) The product id of Lucerne Milk with Reduced Fat is [P123]
- Product brand name:
  - (ex) The brand name of [P123] is Lucerne.
  - (ex) Lucerne is [P123]'s brand name.
- Product attributes:
  - (ex) The attributes of [P123] include organic, gluten free, and kosher.
  - (ex) The attributes of [P123] include kosher, organic, and gluten free.
- Aggregation:
  - (ex) Lucerne products are dairy you can depend on. It produces milk such as [P123], …

Unifying Heterogeneous Data to Text

Transaction Data

For each type of translation, we create some natural language templates.

- Top 10 converted items for the search query 'milk' at Costco is [P001], [P002], …, and [P010]
- A customer bought [P001], [P022], …, and [P042] together
- …

Unifying Heterogeneous Data to Text

Product Knowledge Graph, Ontology, Taxonomy

We may generate descriptions across multiple tables (e.g., between product and recipe). This helps us answer questions such as complementary products, substitute products, etc.

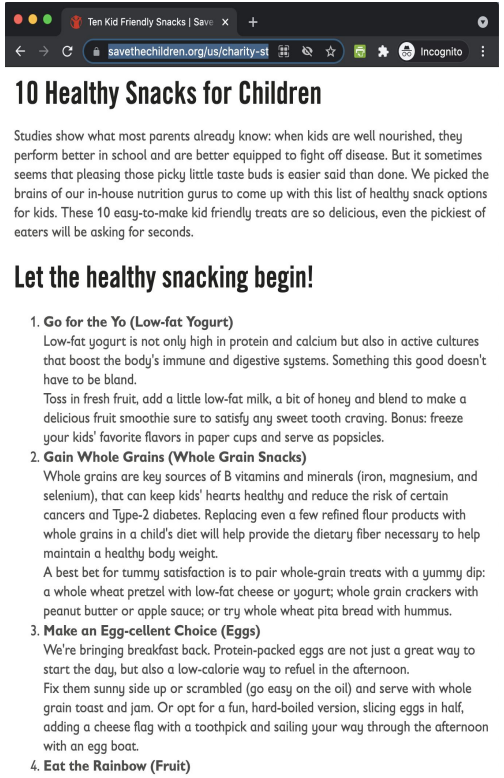
- For example, we have "catalog product class - product serving size -> Volume" in our KG. We can use it as a template and the instances of these two classes to generate something like "Reduced fat Lucerne Milk has 8 serving size of 1 cup per container"

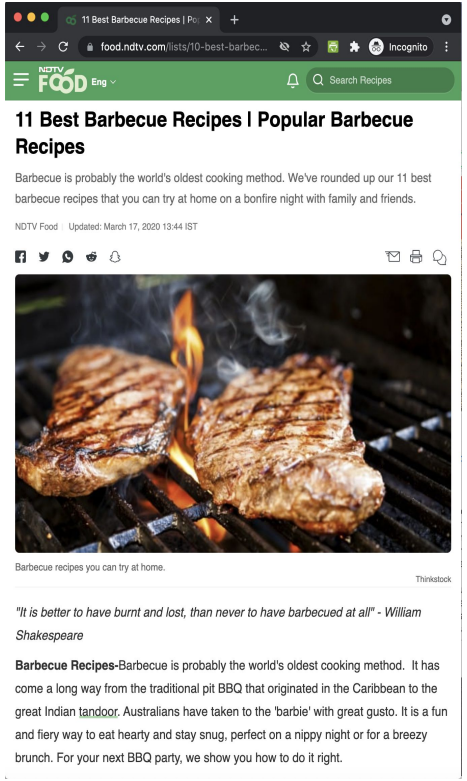
- For example, we can also generate something like "Reduced fat Lucerne Milk has 140 Calories" based on another relation (product fatCaloriesPerServing) between "catalog product class" and "energy".

Unifying Heterogeneous Data to Text

Unstructured Data

- For example: recipes, web articles, …
- Perform **entity linking** and embed product IDs such as [Pxxx] in the doc



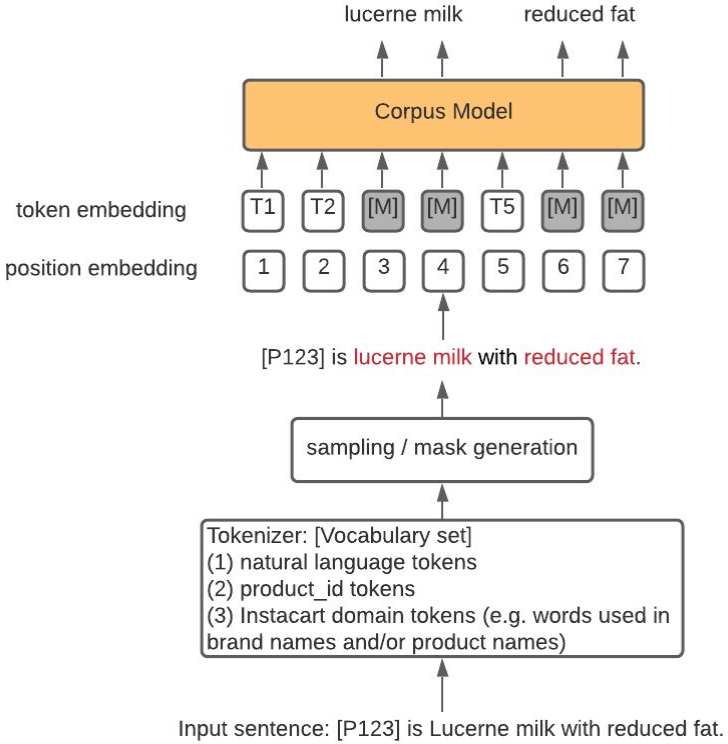


Pre-training a LM

Consider the pre-training for **masked language models**.

We first add product_ids and words used in instacart (e.g. words used in brand names and product names) as additional tokens in the vocabulary set.

- Input sentence: “[P123] is Lucerne Milk with Reduced Fat”
- Masking: “[P123] is [Mask] [Mask] with [Mask] [Mask]
- We want to predict the original input sentence based on the masked sentence



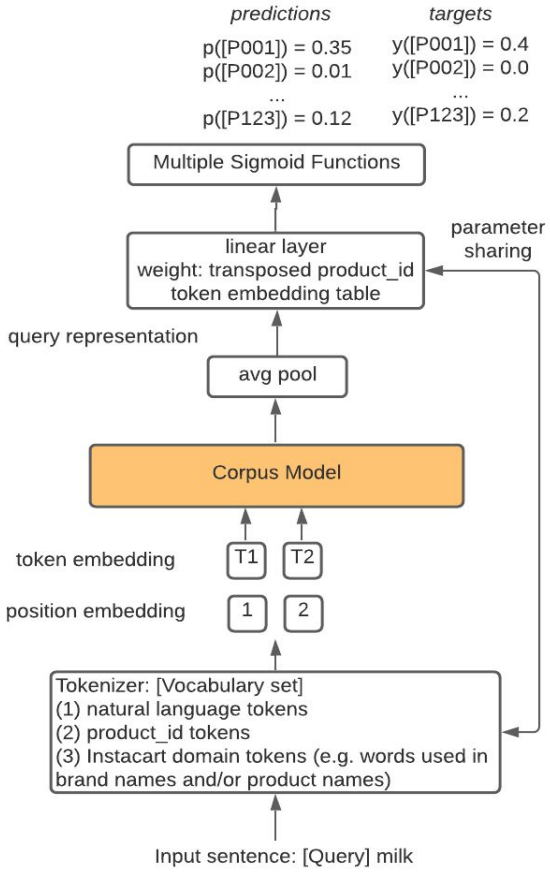
Fine-tuning - Document Retrieval

We treat *document retrieval* as a multiclass-multilabel classification process, where # of classes = # of product_ids.

Training data

<“[Query] milk”: ([P001], 0.4), ([P006], 0.3), ([P123], 0.2), … , (P[234], 0.1)>

Labels can be soft labels or hard labels.



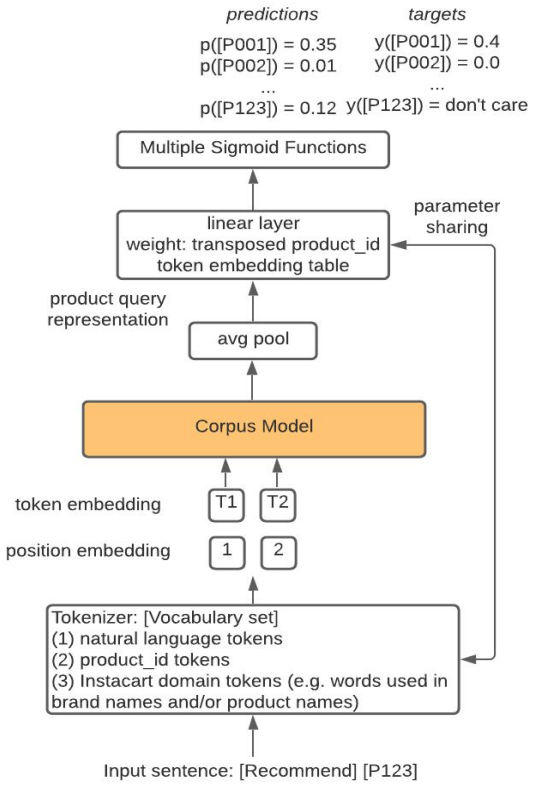
(a) Document Retrieval

Fine-tuning - Document Recommendation

The input is a document identifier and the output is one or more relevant document identifiers. We also treat *Document Recommendation* as a multiclass-multilabel classification process, where # of classes = # of product_ids.

<“[Recommend] [P001]”: ([P006], 0.8), ([P123], 0.9), … , (P[234], 0.7)>

Labels can be soft labels or hard labels.



(b) Related Document Retrieval

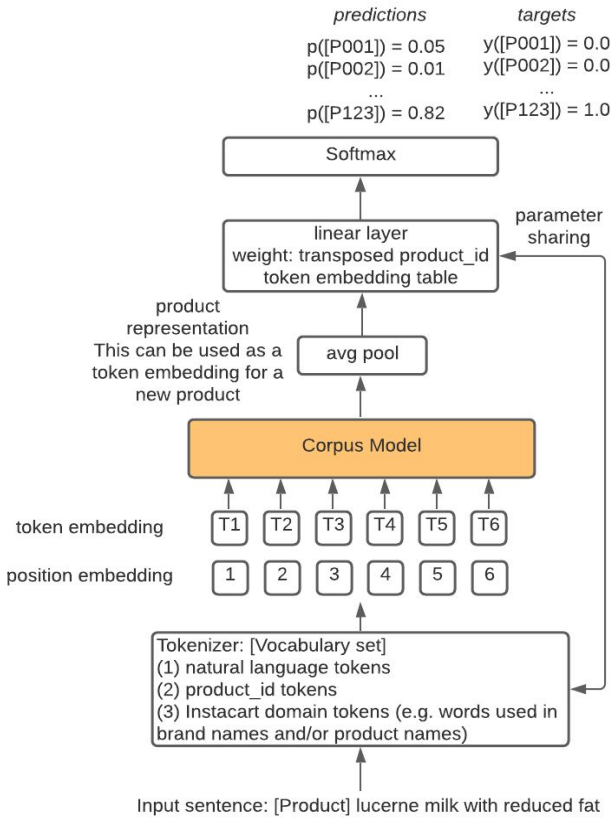
Fine-tuning - Document Encoding

The objective of this task is to encode token embedding for a given product description. The input is a product description (product brand name + product name + attributes) and the output is corresponding product id. Note that this is multi-class single label classification.

<“[Product] lucerne milk with reduced fat”: ([P123], 1.0)>

Label is product_id.

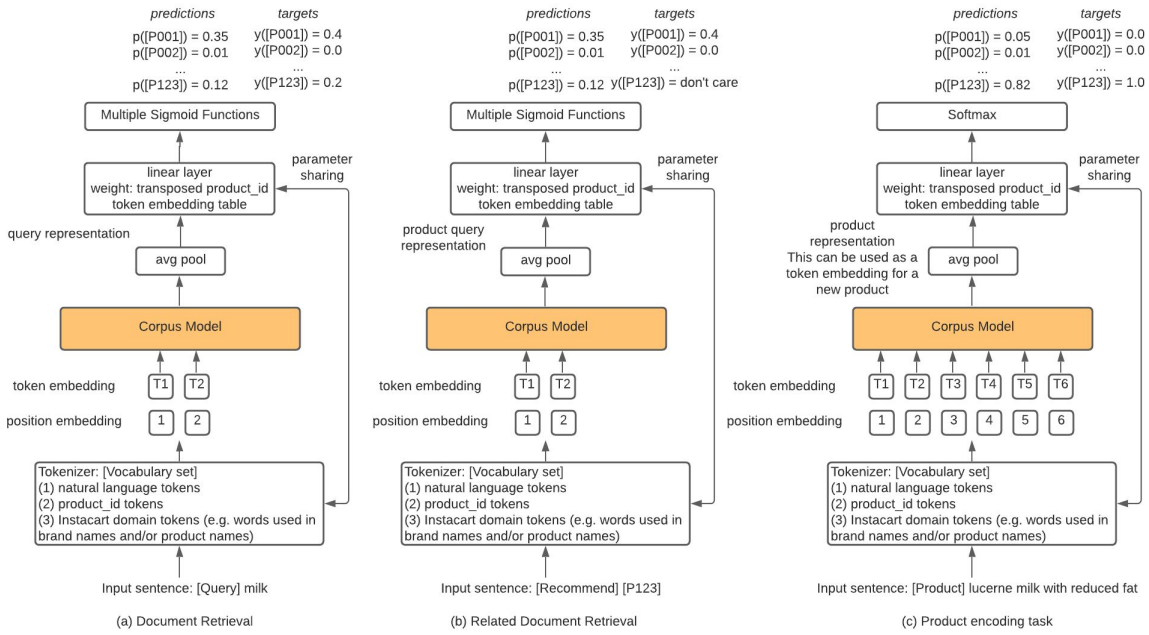
Once the training is done, the model will be able to encode new products. We then append these additional token embeddings to the weight of the final output layer.



(c) Product encoding task

Multi-task Learning

Finally we fine-tune the model from the pre-trained checkpoint with all the objectives we mentioned above. Below diagram shows how we feed all the inputs to the corpus model and how we use output of the model in a multi task learning setting.



Remaining problems? Too many …

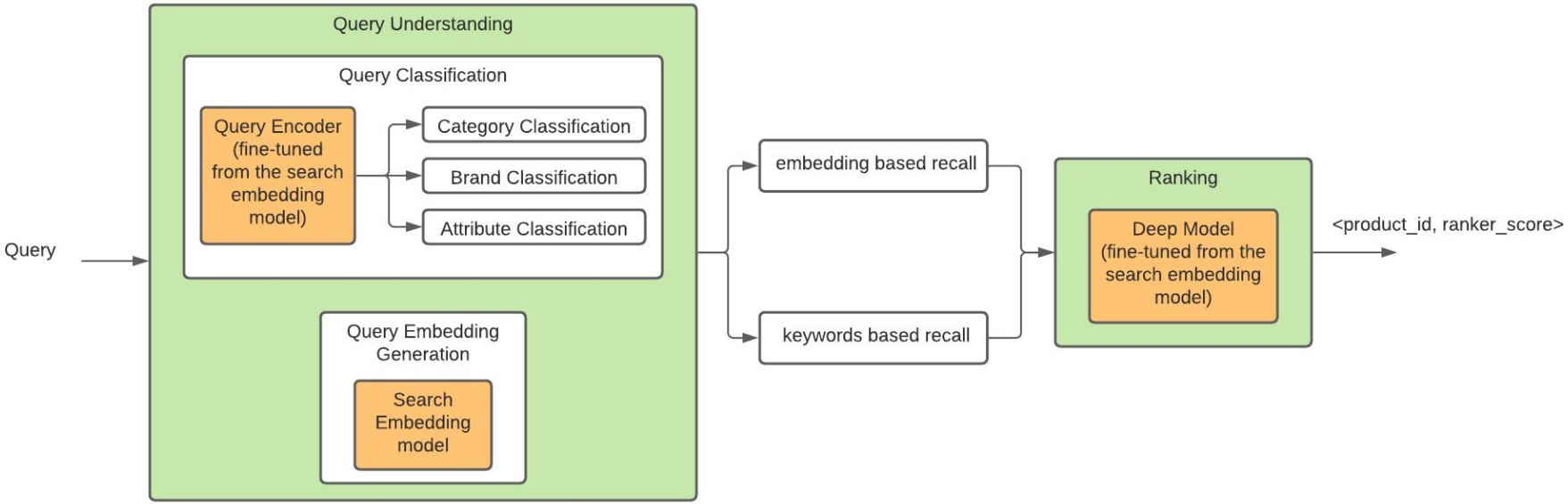
What about personalization?

Templates? Aren’t they biased?

How about new products?

…

A Hybrid Approach



Conclusion

e-Commerce Search is challenging

- Heterogeneous types of data
- Converting the data to structured

A hybrid approach is the most realistic at the moment

- Relying on neural models for high recall
- Relying on classical approaches for precision and scalability

End-to-end Model based IR is becoming increasingly more attractive

- Challenges: vocabulary size, updates, text description generation, etc.

Thanks