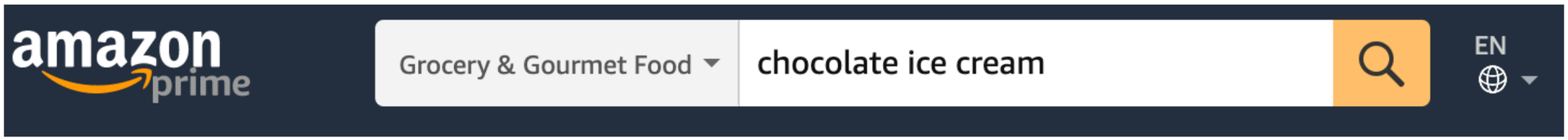
TXtract:
Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories

Giannis Karamanolakis
Columbia University
gkaraman@cs.columbia.edu

Jun Ma, Xin Luna Dong
Amazon.com
{junmaa, lunadong}@amazon.com

Product Understanding
for Search and Question Answering

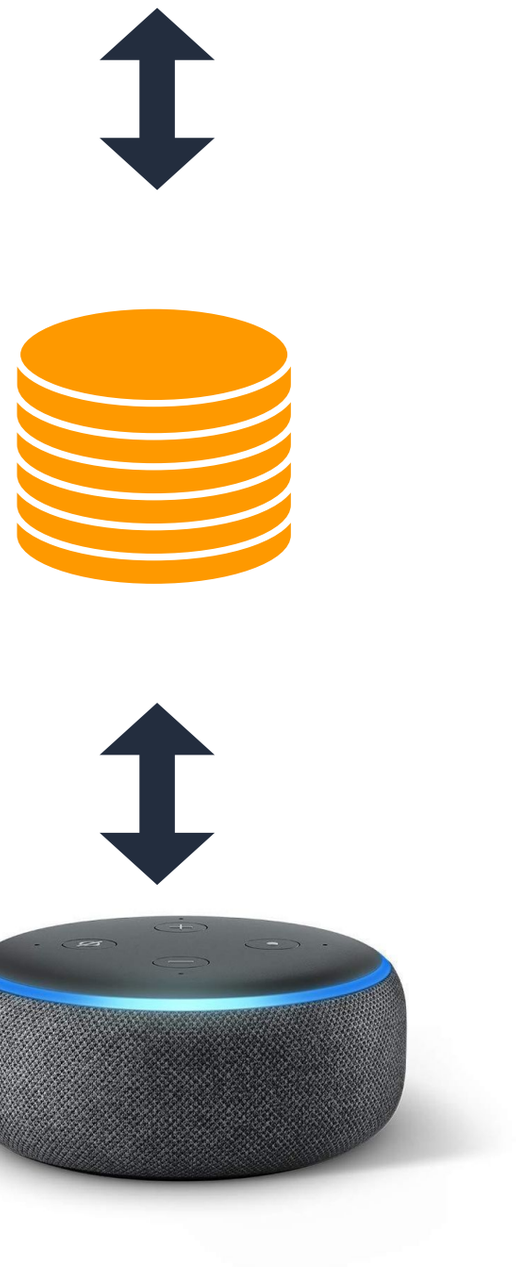


“Alexa, which shampoos contain argan oil?”

Need to Store Structured Knowledge About Products

flavor: “chocolate”

ingredients: “biotin”, “argan oil”, …



“Alexa, which shampoos contain argan oil?”

Understanding Values for Product Attributes

- **Product catalog:**

Product Attributes

Products

Product ID	Brand	Flavor	Size	Ingredients
B00FZHEGGW	Fage	Plain	35.3 oz	…
B0725VRRLP	Ben & Jerry’s			…
…	…	…	…	…

Understanding Values for Product Attributes

- **Product catalog:**

Product Attributes

Products

Product ID	Brand	Flavor	Size	Ingredients
B00FZHEGGW	Fage	Plain	35.3 oz	…
B0725VRRLP	Ben & Jerry's	???	???	…
…	…	…	…	…

(-) Issue: catalog is missing attribute values for many products

Understanding Values for Product Attributes

- **Product catalog:**

Product Attributes

Products

Product ID	Brand	Flavor	Size	Ingredients
B00FZHEGGW	Fage	Plain	35.3 oz	…
B0725VRRLP	Ben & Jerry's	???	???	…
…	…	…	…	…

(-) Issue: catalog is missing attribute values for many products

- **This work:** extract values from product profiles (titles, descriptions)

Brand Flavor Size

Ben & Jerry's Strawberry Cheesecake Ice Cream 16 oz

In Stock.

- Ben & Jerry's Strawberry Cheesecake ice cream pint
- Includes Fairtrade certified sugar

Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions

- **Previous work:** deep neural networks for **sequence tagging**

[Zheng et al., KDD'18]
[Xu et al., ACL'19]
[Rezk et al., ICDE'19]

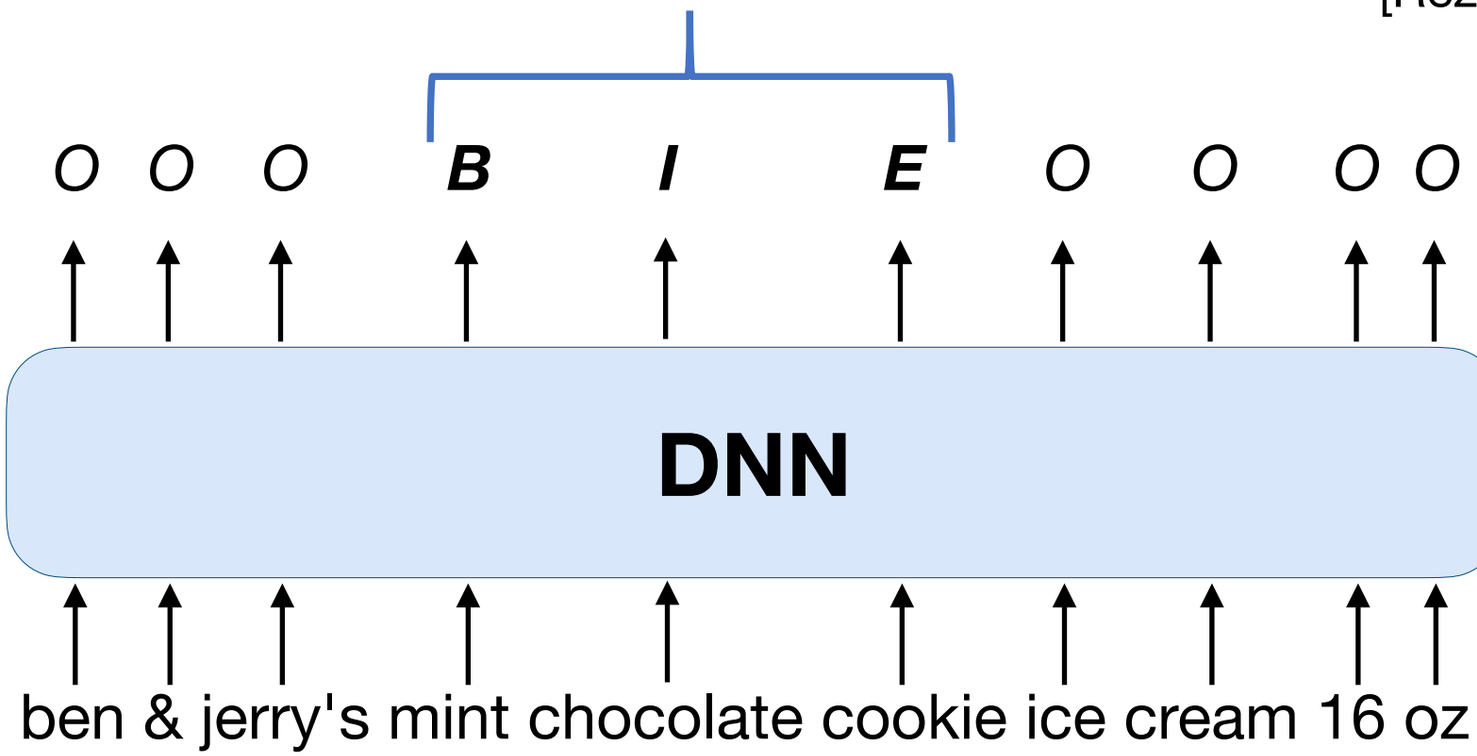
Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**

[Zheng et al., KDD'18]
[Xu et al., ACL'19]
[Rezk et al., ICDE'19]

BIOE Tagging Example

extracted *flavor* value: "mint chocolate cookie"



Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**
- **Limitations of previous work:**

(-) designed for a single category

[Zheng et al., KDD'18]

"Dog Food"
Atribute Values

DNN

"Dog Food"
Product Titles

[Xu et al., ACL'19]

"Sports"
Atribute Values

DNN

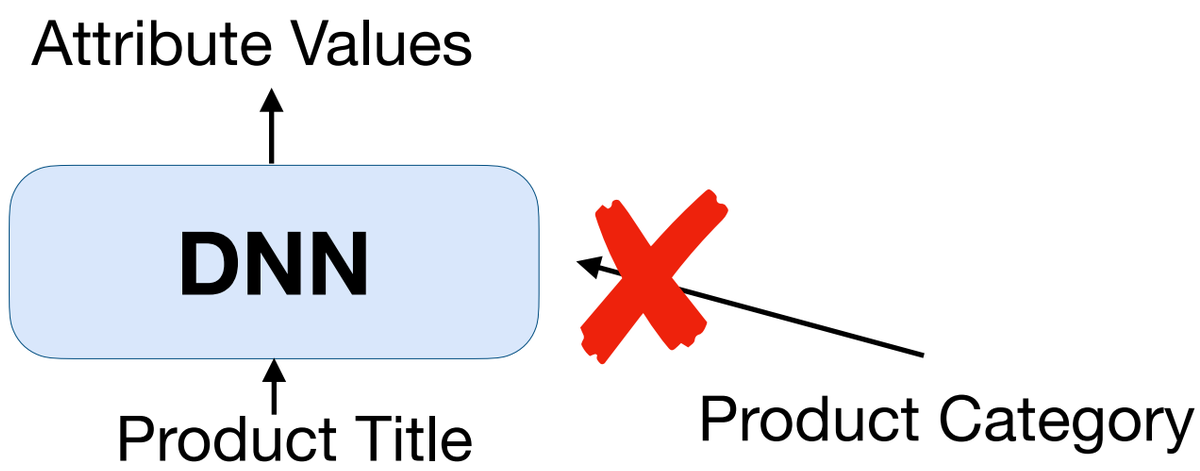
"Sports"
Product Titles

Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**
- **Limitations of previous work:**

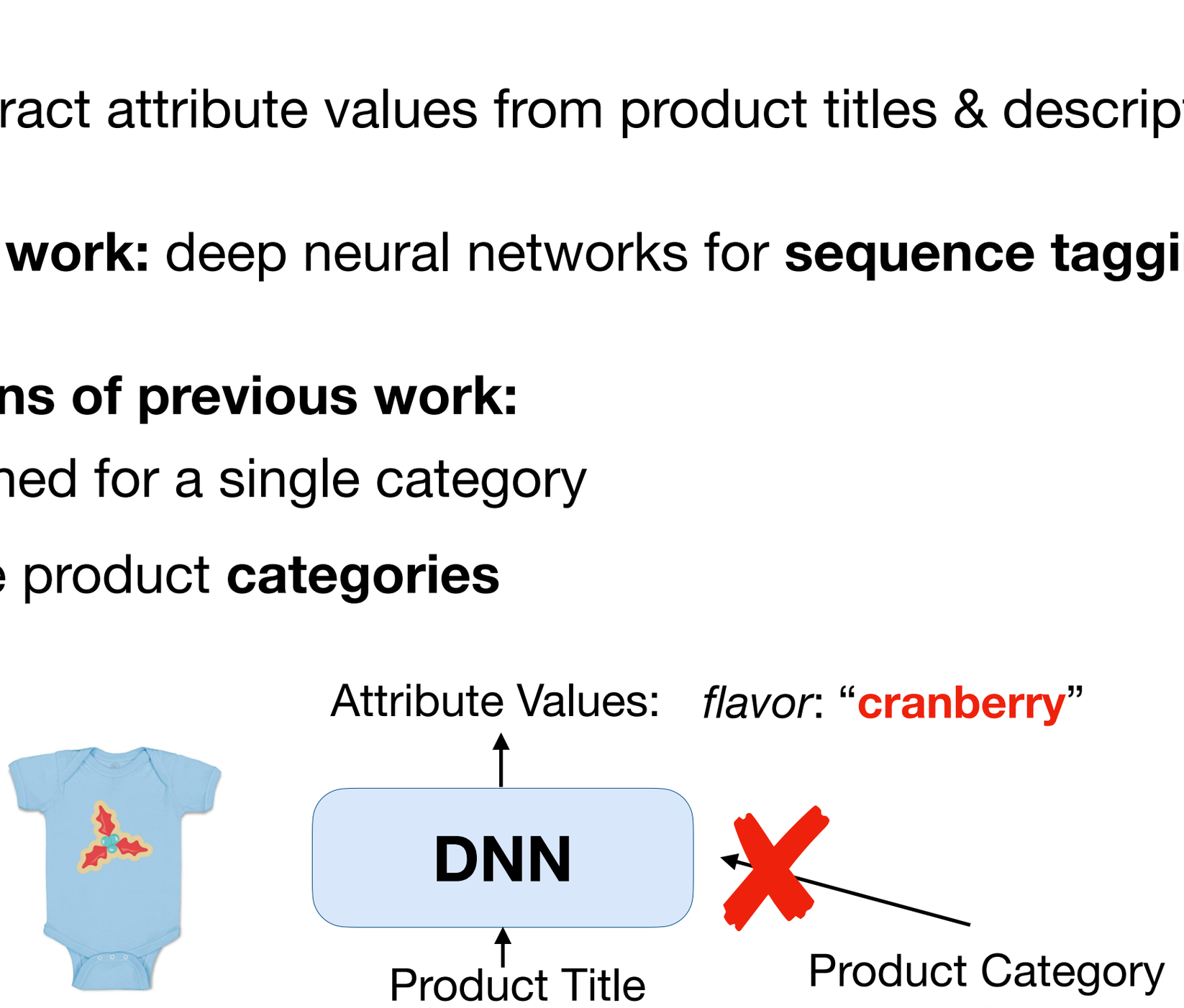
(-) designed for a single category

(-) ignore product **categories**



Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**
- **Limitations of previous work:**
  - **(-)** designed for a single category
  - **(-)** ignore product **categories**



*"Custom Boy & Girl Baby Bodysuit Red Blue **Cranberry** Funny Cotton Baby Clothes"*

Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**
- **Limitations of previous work:**

(-) designed for a single category

(-) ignore product **categories**

(-) hard to capture **diversity** of categories

Digital Camera

flavor?
Not applicable

Vitamin

flavor: “fruit”

Fruit

flavor: ~~“fruit”~~
Not valid

Attribute Value Extraction from Product Profiles

- **Goal:** extract attribute values from product titles & descriptions
- **Previous work:** deep neural networks for **sequence tagging**
- **Limitations of previous work:**

(-) designed for a single category

(-) ignore product **categories**

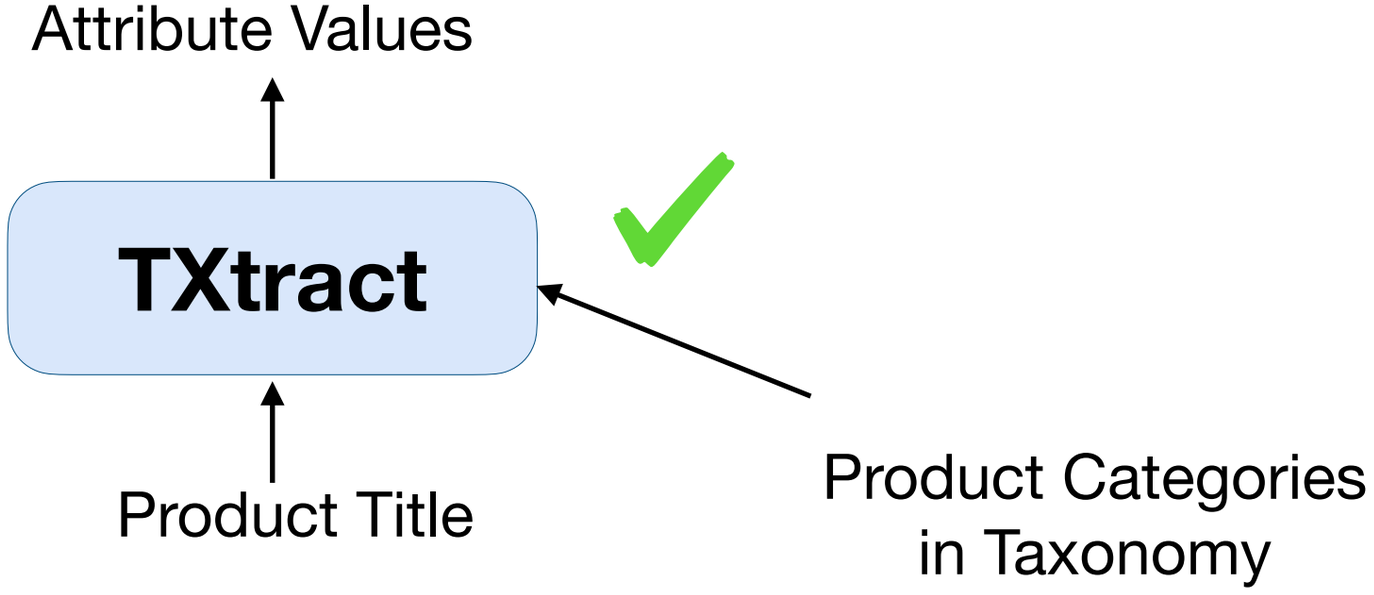
(-) hard to capture **diversity** of categories

(-) hard to scale to **real-world** product taxonomies

- **>100M products**
- **>10K categories**
- **Products/categories continuously added**

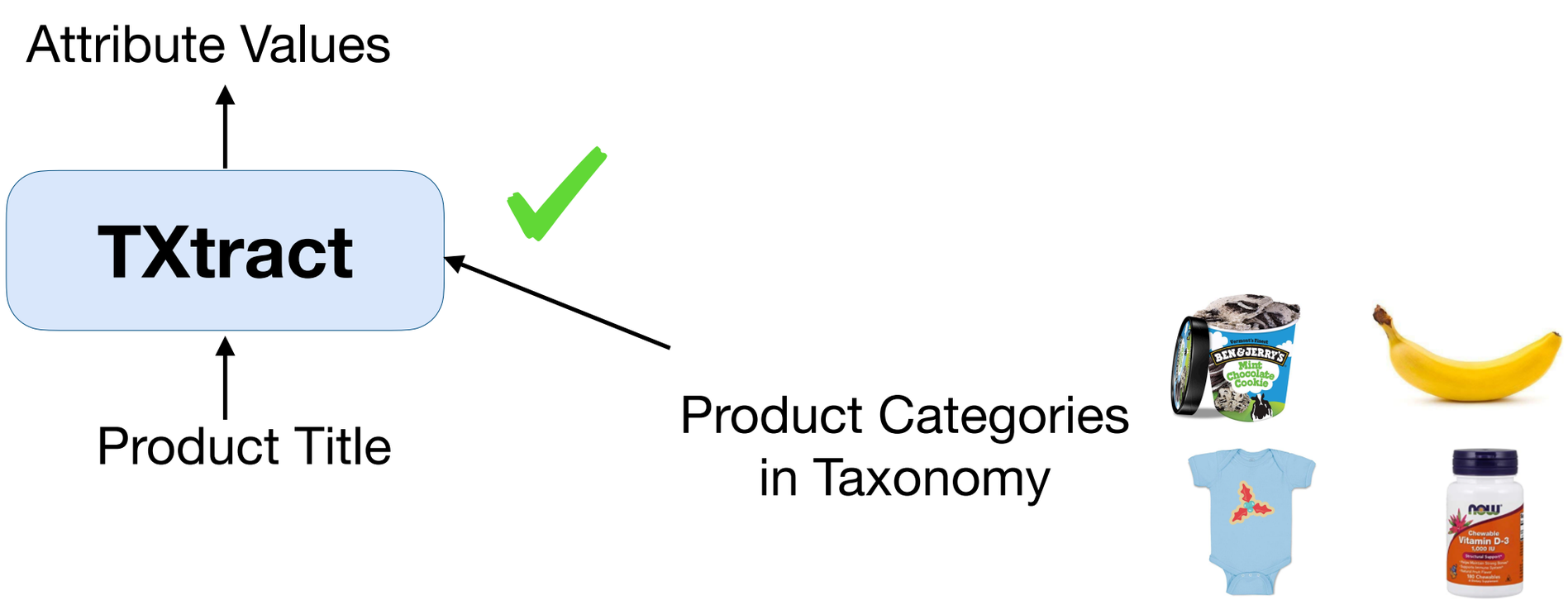
TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction



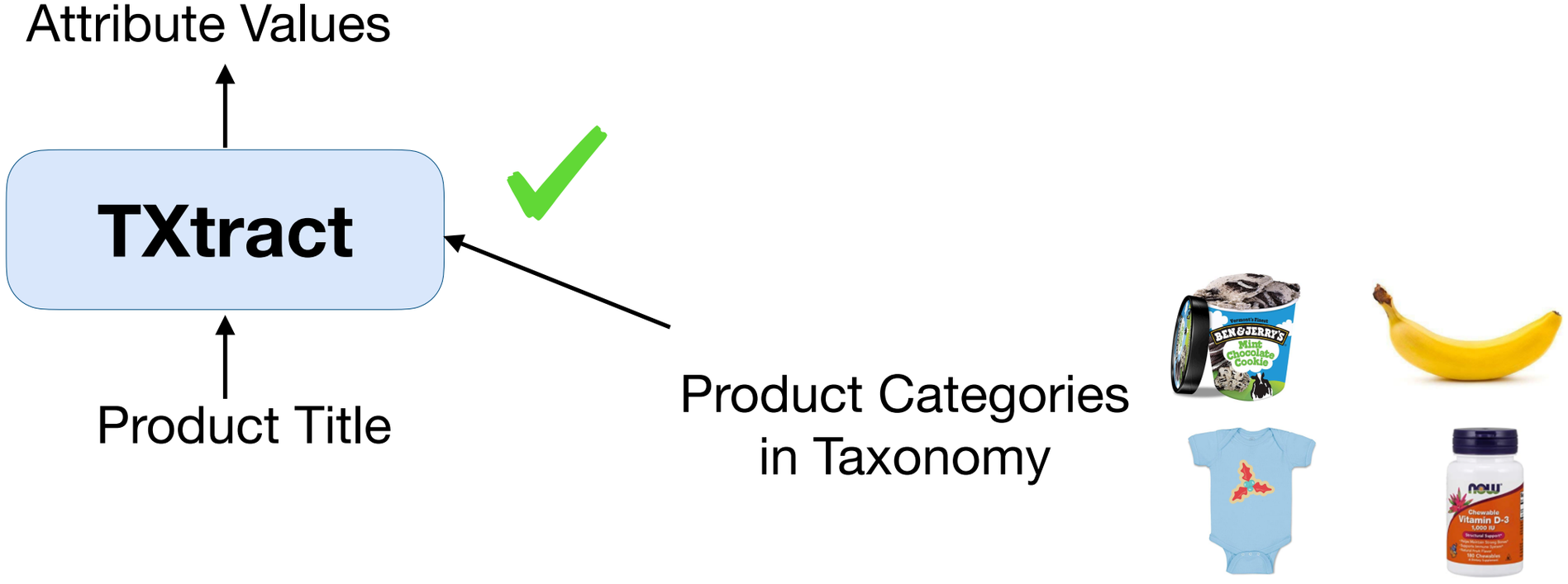
TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction
- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model



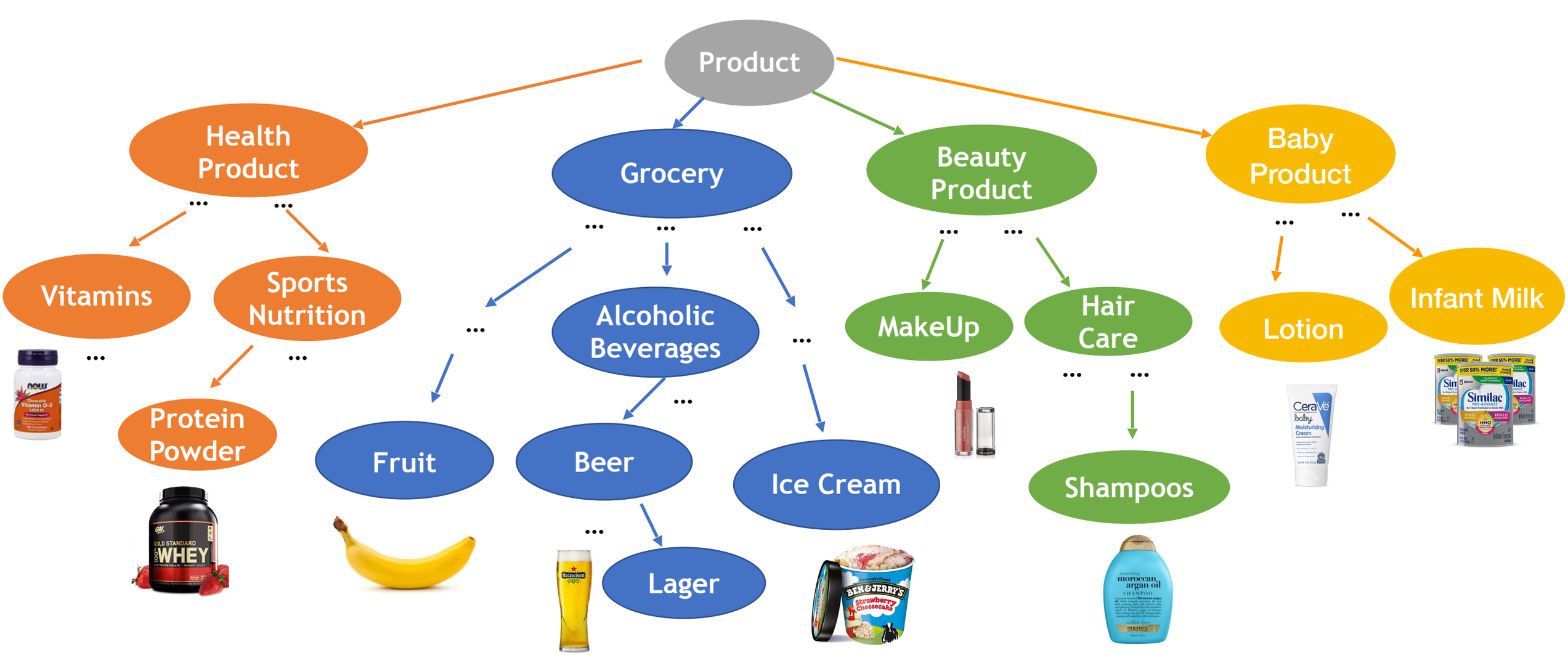
TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction
- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model
  2. Extract **category-specific** attribute values using **conditional** self-attention



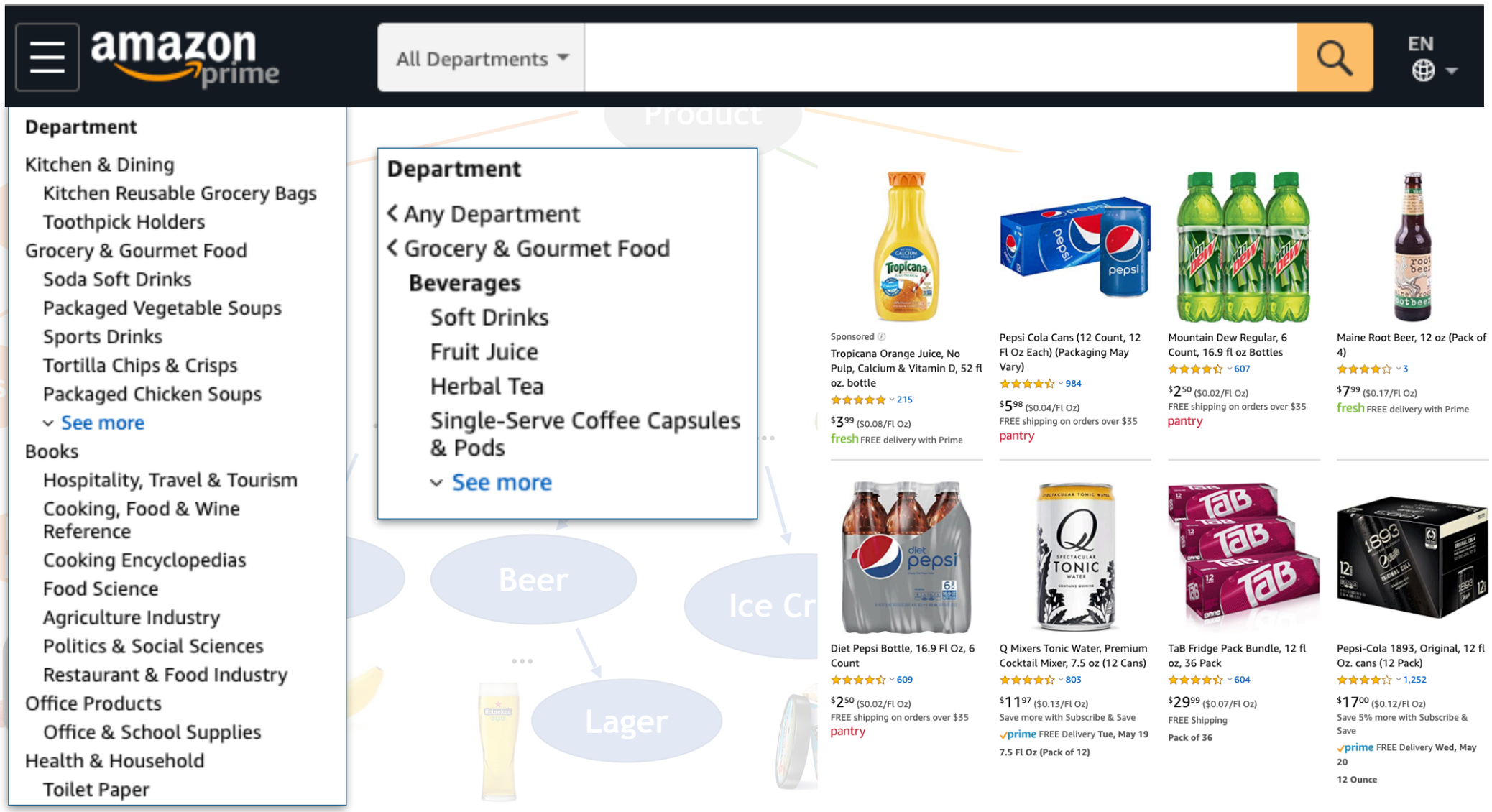
TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction
- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model
  2. Extract **category-specific** attribute values using **conditional** self-attention
  3. **Scale up** extraction to hierarchical taxonomies with **thousands** of categories



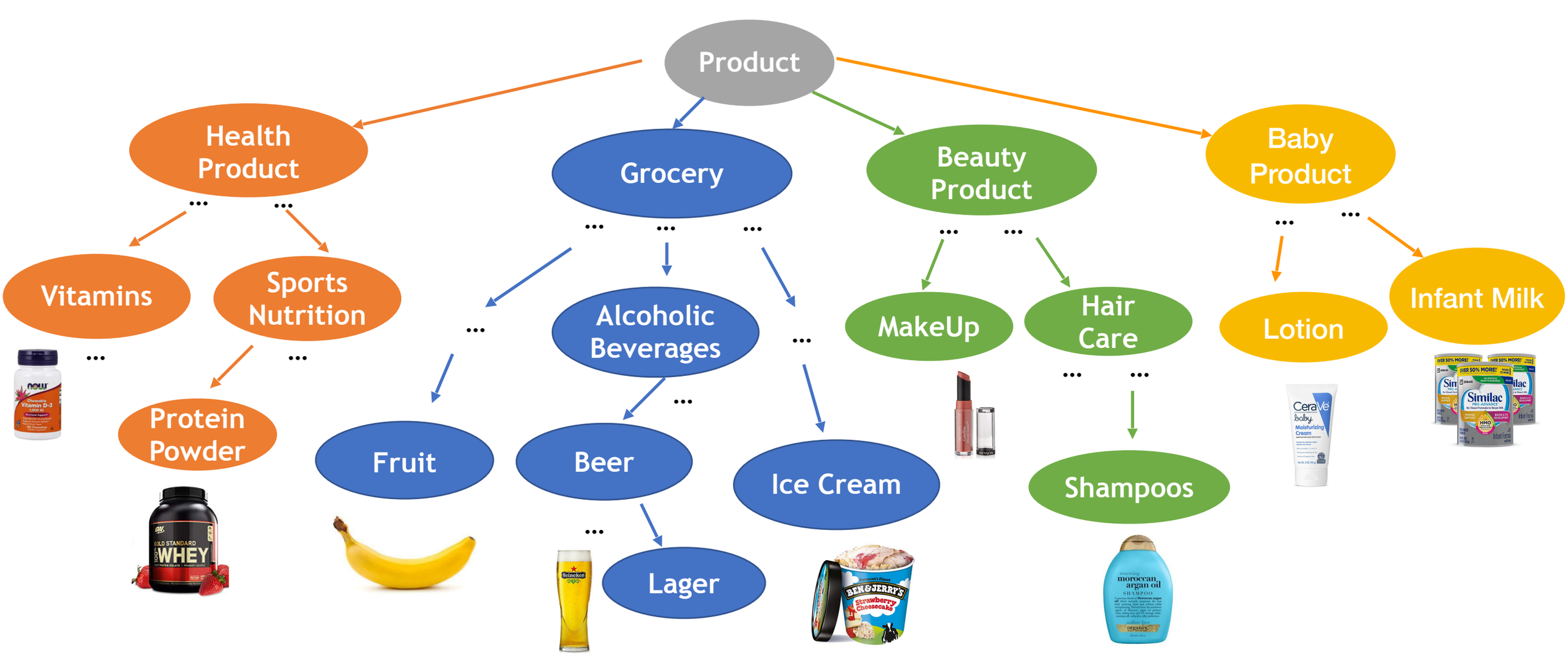
TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction
- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model
  2. Extract **category-specific** attribute values using **conditional** self-attention
  3. **Scale up** extraction to hierarchical taxonomies with **thousands** of categories



TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction
- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model
  2. Extract **category-specific** attribute values using **conditional** self-attention
  3. **Scale up** extraction to hierarchical taxonomies with **thousands** of categories



TXtract: Extraction for Thousands of Product Categories

- **TXtract:** a taxonomy-aware neural network for attribute value extraction

- **Our Contributions:**
  1. Consider **multiple** categories efficiently with a **single** model
  2. Extract **category-specific** attribute values using **conditional** self-attention
  3. **Scale up** extraction to hierarchical taxonomies with **thousands** of categories
  4. Make TXtract **robust** to noisy assignments using **multi-task** training

Outline

1. Attribute Value Extraction from Product Profiles

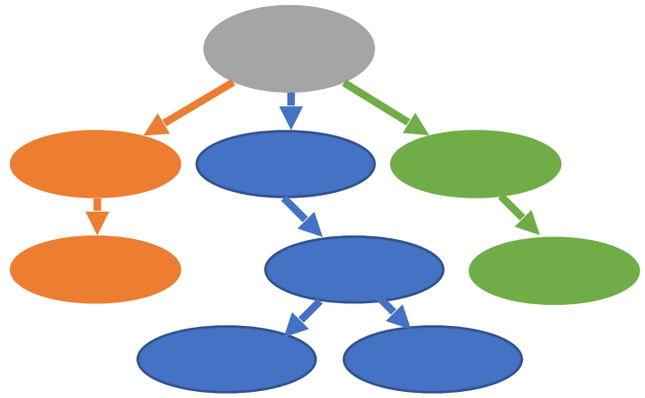
2. TXtract: Taxonomy-Aware Attribute Value Extraction

3. Experiments

4. Conclusions and Ongoing Work

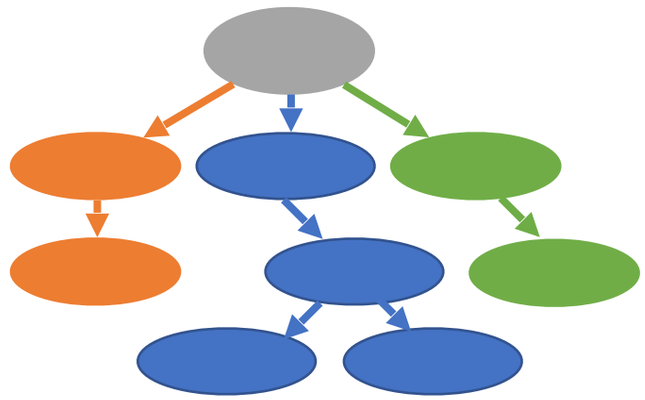
Scaling to Thousands of Product Categories - Challenges

- **Goal:**
  - Extract attribute values for products …
  - … from thousands of **diverse** categories
  - … organized in **hierarchical** taxonomies



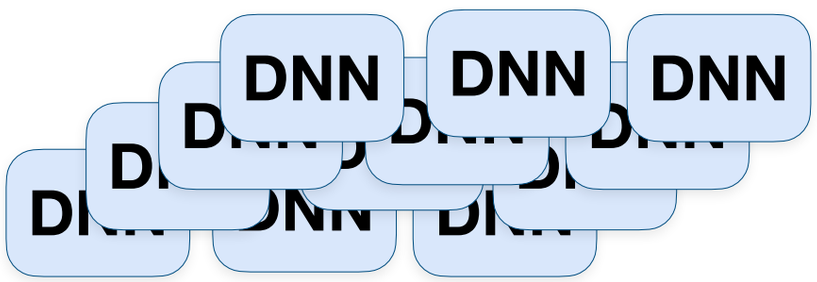
Scaling to Thousands of Product Categories - Challenges

- **Goal:**
  - Extract attribute values for products …
  - … from thousands of **diverse** categories
  - … organized in **hierarchical** taxonomies



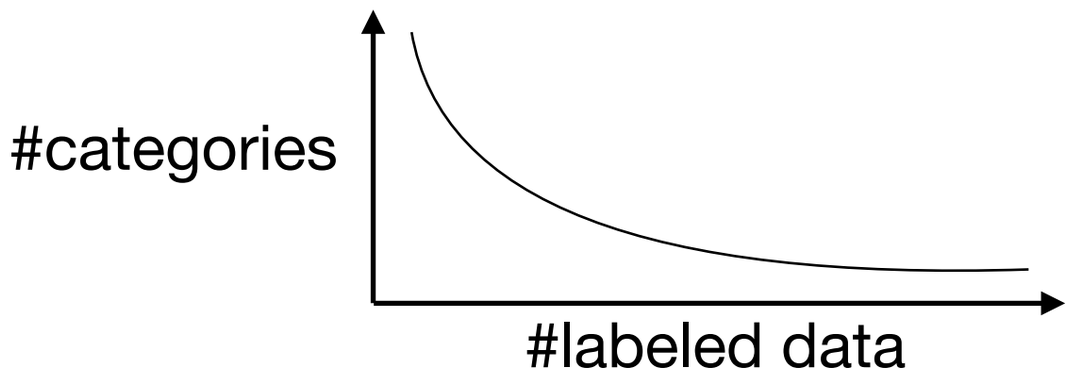
- **Approach 1:** train a **separate** DNN for each category

(-) expensive



store/orchestrate
1000+ models

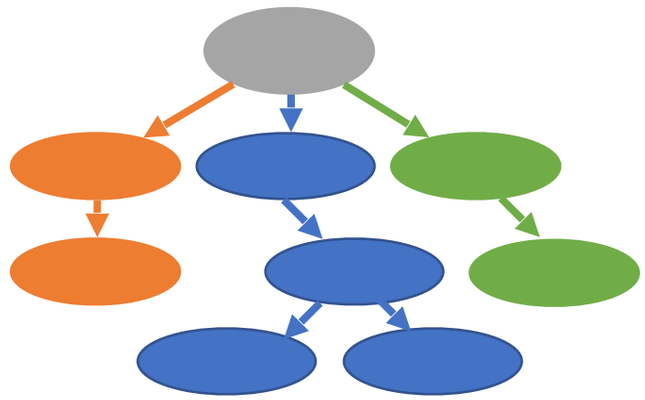
(-) prone to overfitting



most categories have
<<1000 labeled training data

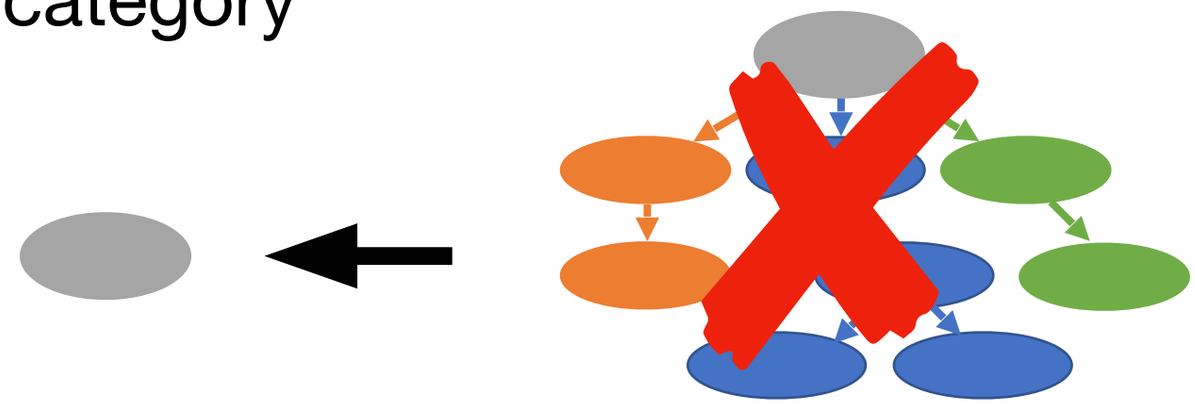
Scaling to Thousands of Product Categories - Challenges

- **Goal:**
  - Extract attribute values for products …
  - … from thousands of **diverse** categories
  - … organized in **hierarchical** taxonomies



- **Approach 1:** train a **separate** DNN for each category

- **Approach 2:** assume a single “flat” category

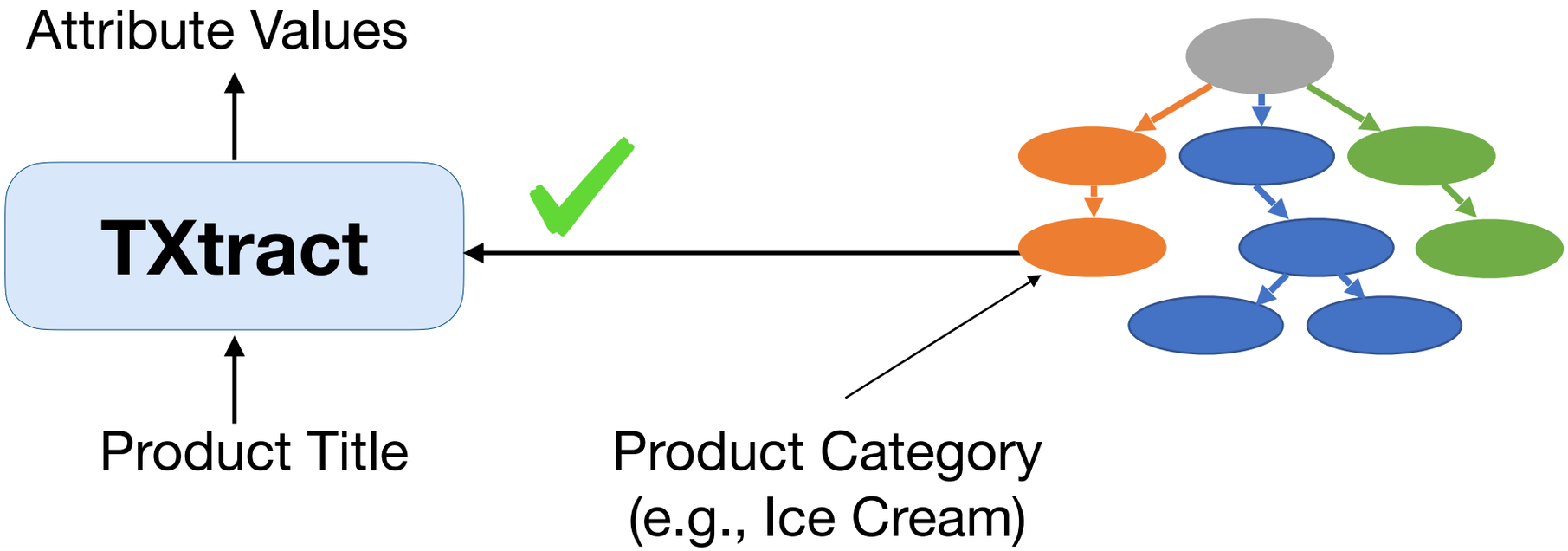


(+) cheaper: train a single DNN

(-) not effective: missing category-specific characteristics

TXtract: Taxonomy-Aware Attribute Value Extraction

- TXtract leverages the **hierarchical** product taxonomy



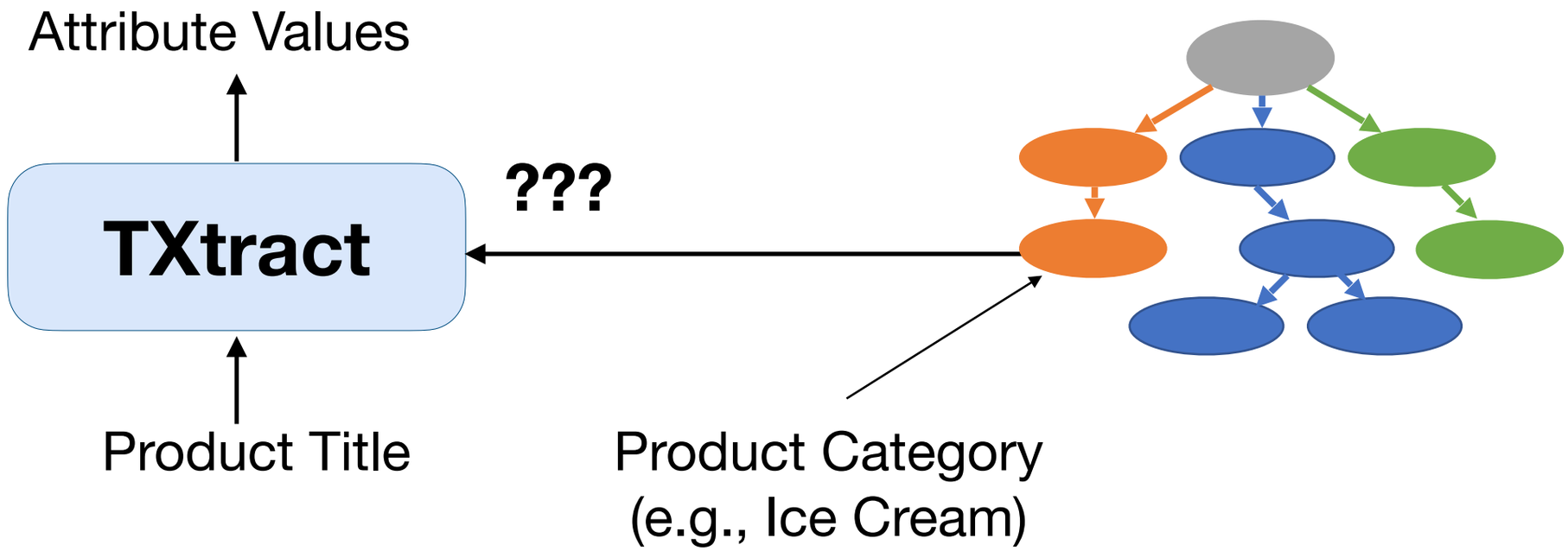
(+) efficient: trained on **all** categories in parallel

(+) effective: extracts **category-specific** attribute values

- product category -> attribute applicability, valid attribute values

TXtract: Taxonomy-Aware Attribute Value Extraction

- TXtract leverages the **hierarchical** product taxonomy!



(+) efficient: trained on **all** categories in parallel

(+) effective: extracts **category-specific** attribute values

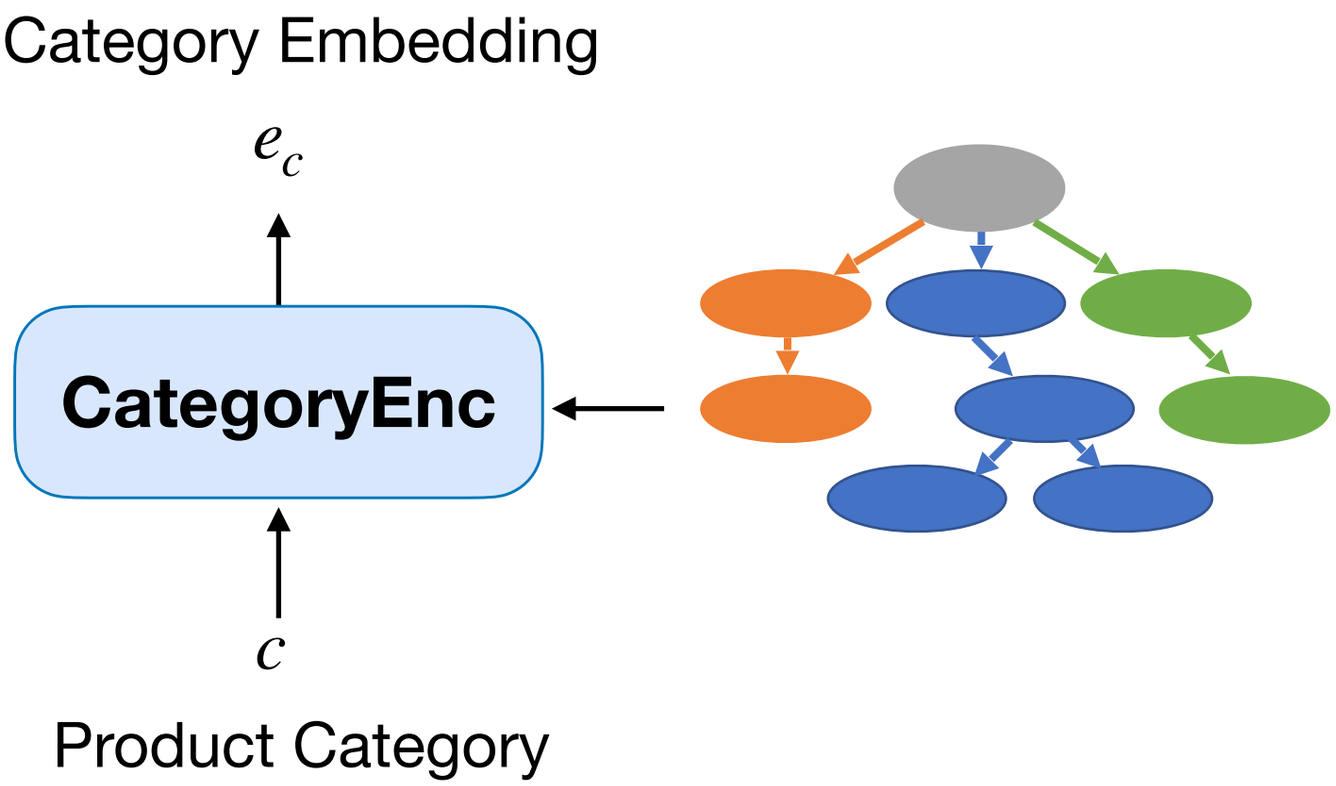
- How to condition on **hierarchical** product categories?

Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
2. Conditional Self-Attention: **conditions** on category embeddings e_c

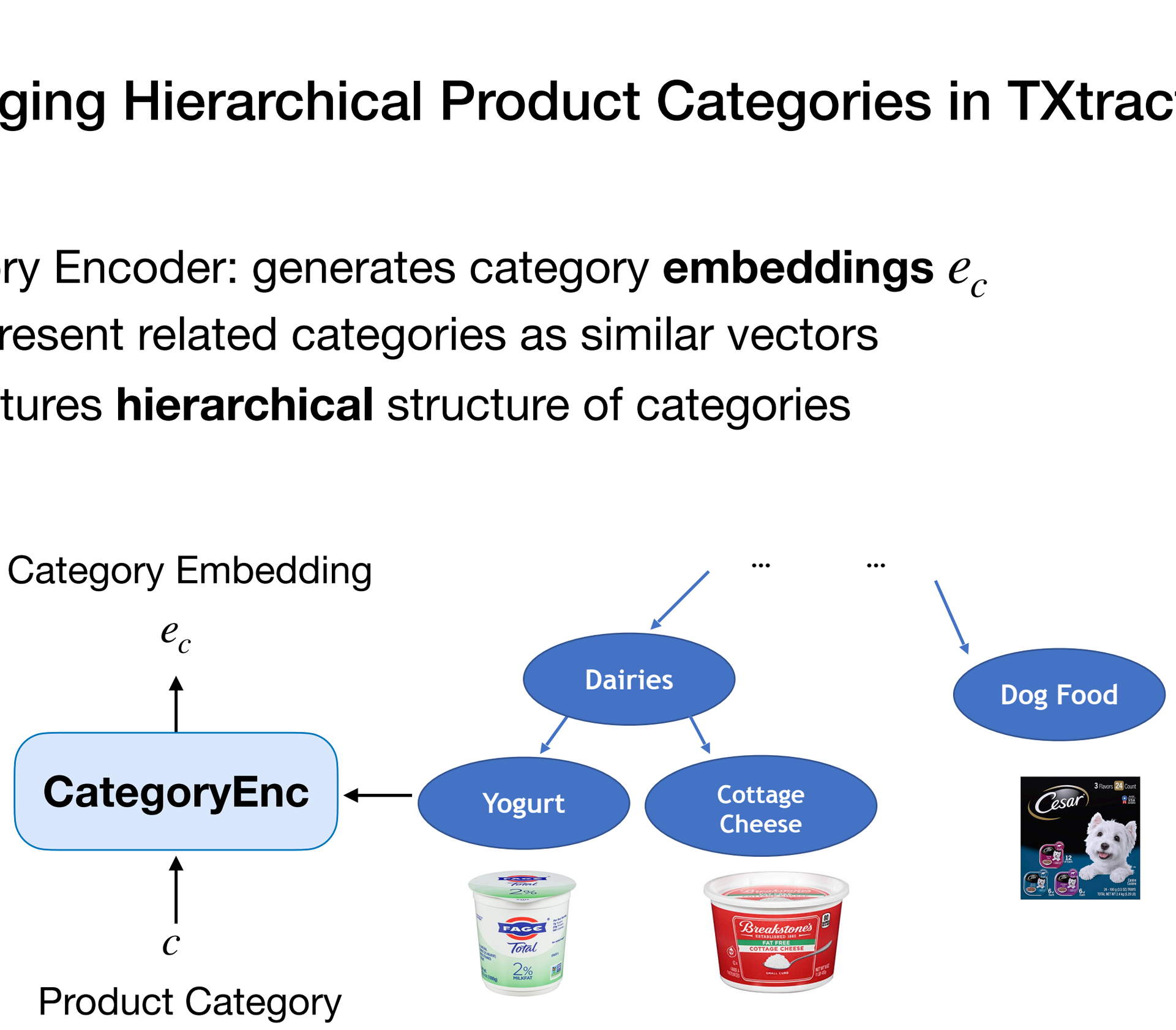
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
   - Represent related categories as similar vectors



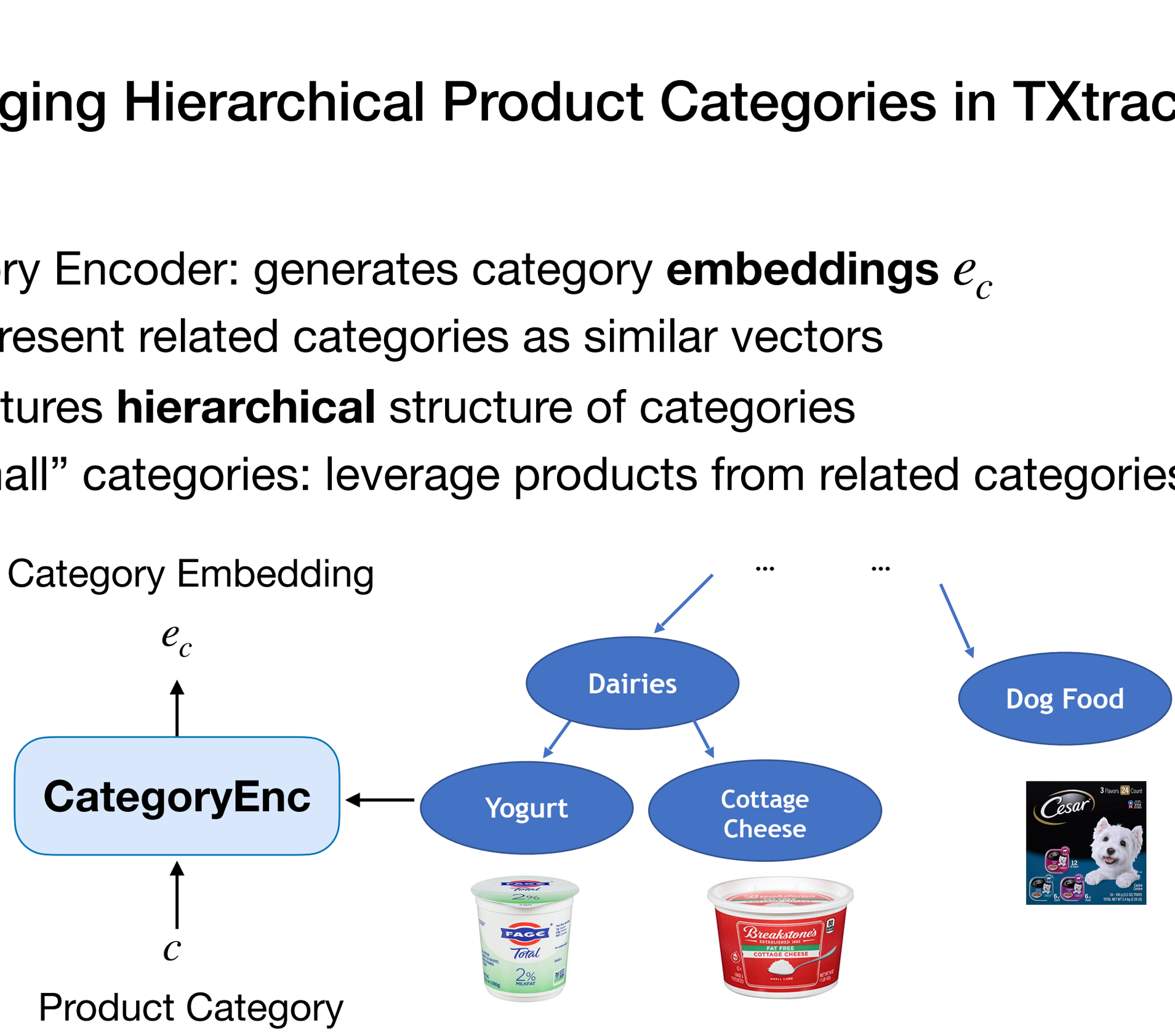
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
   - Represent related categories as similar vectors
   - Captures **hierarchical** structure of categories



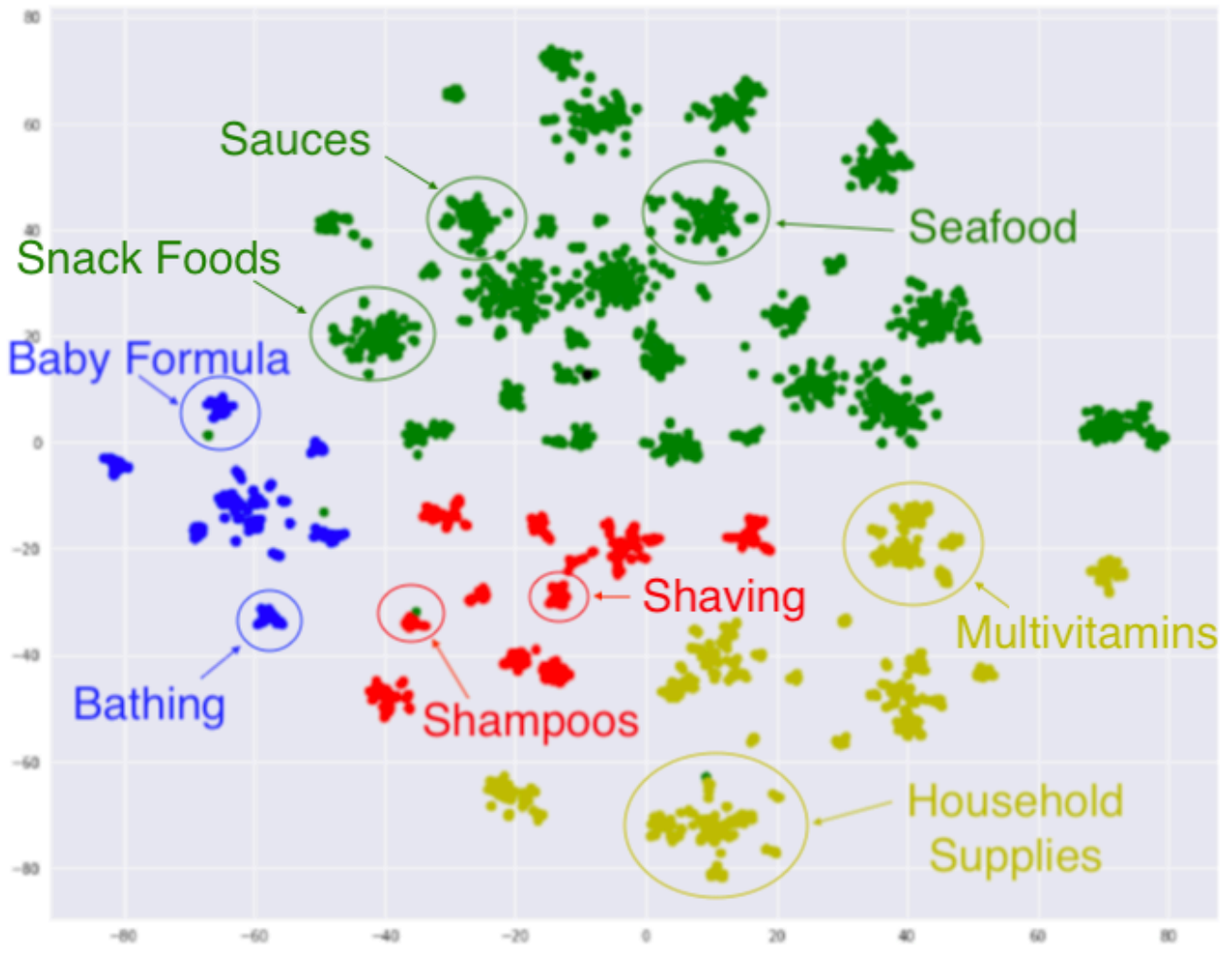
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
   - Represent related categories as similar vectors
   - Captures **hierarchical** structure of categories
   - "Small" categories: leverage products from related categories



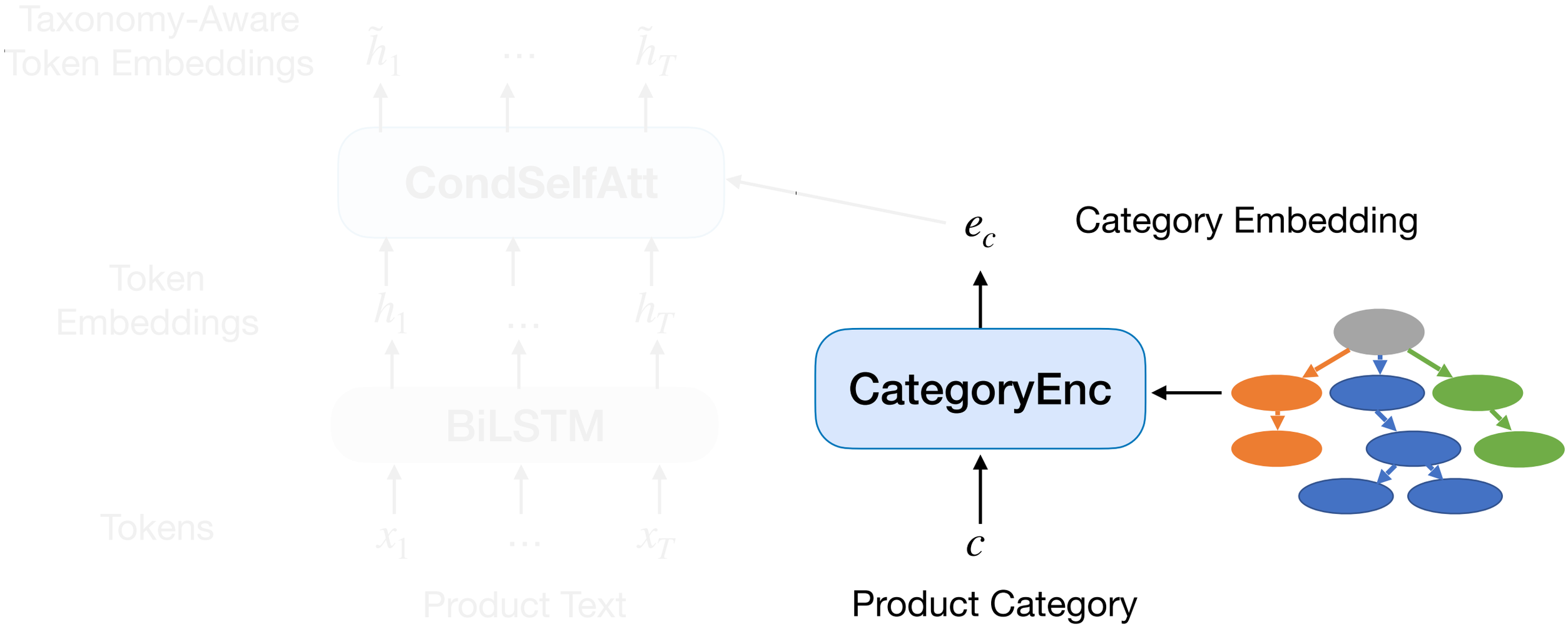
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
   - Represent related categories as similar vectors
   - Captures **hierarchical** structure of categories
   - … using Poincaré embeddings



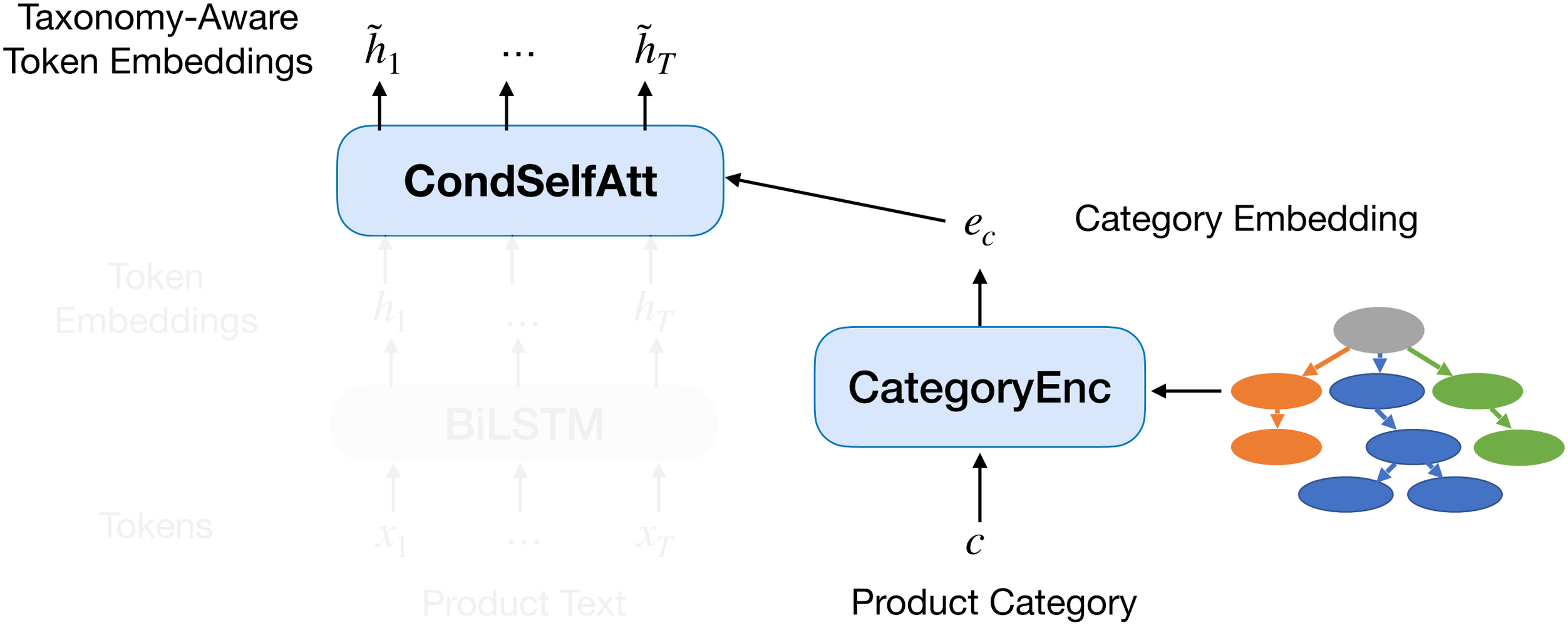
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c



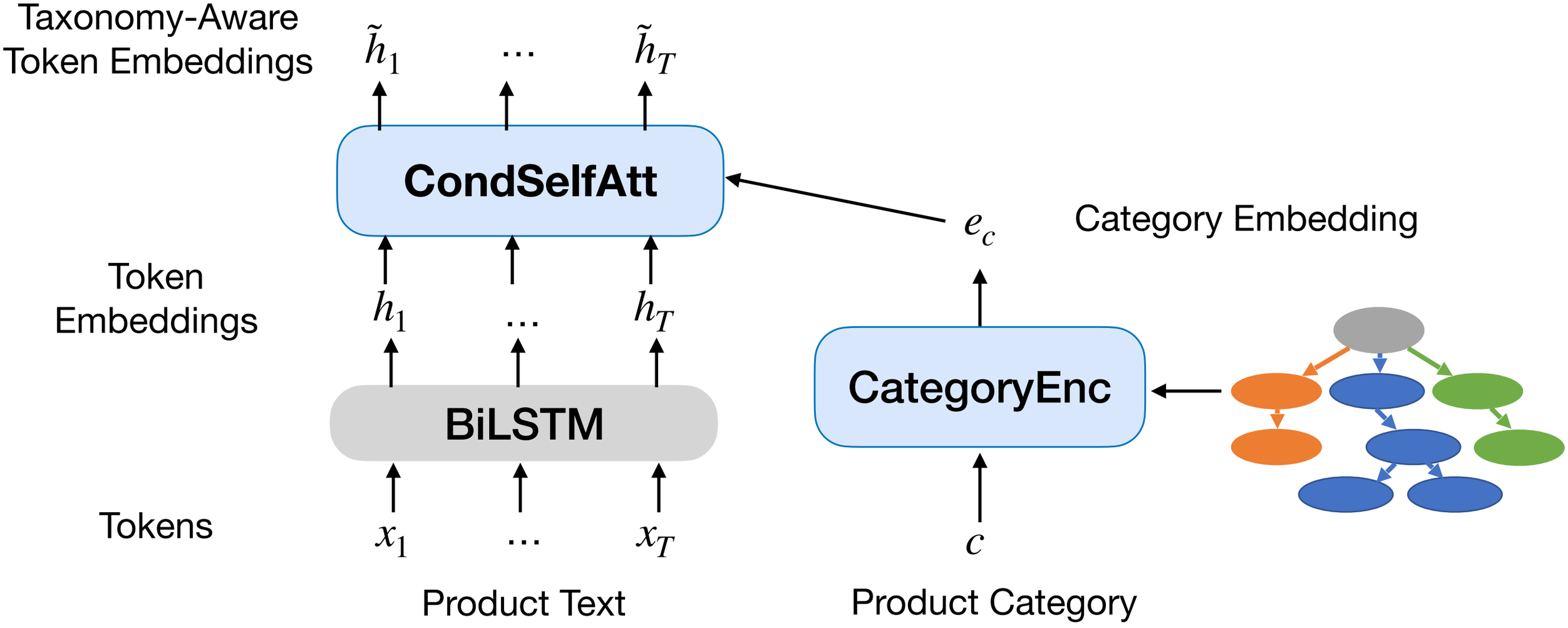
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
2. Conditional Self-Attention: **conditions** on category embeddings e_c



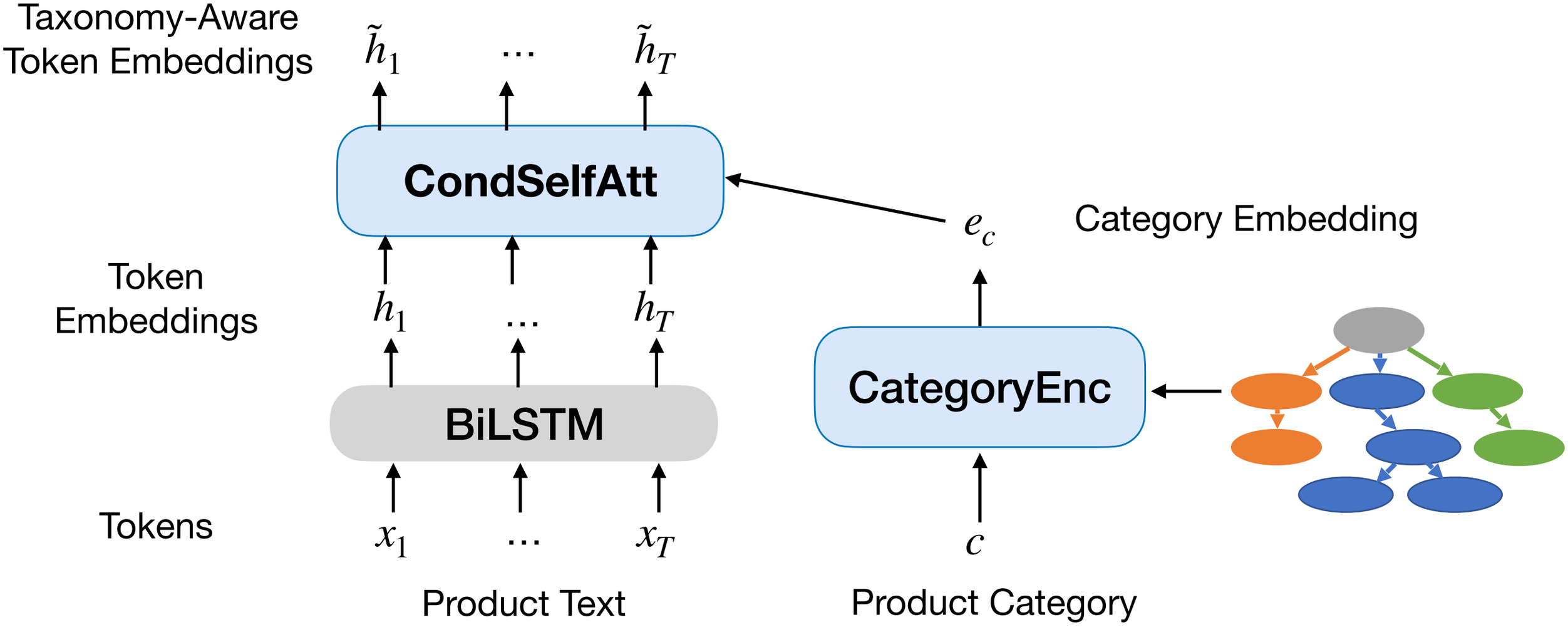
Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
2. Conditional Self-Attention: **conditions** on category embeddings e_c

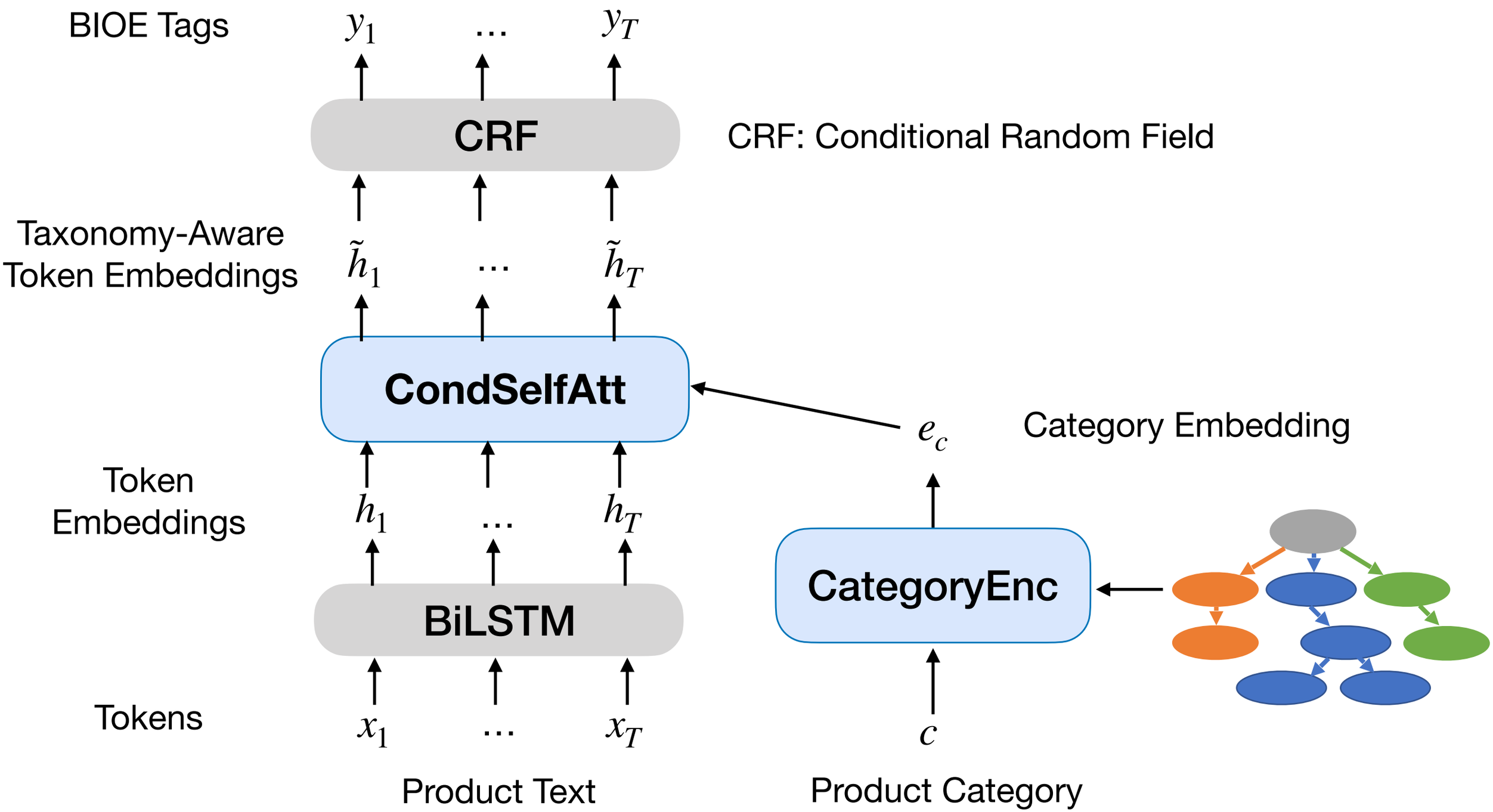


Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
2. Conditional Self-Attention: **conditions** on category embeddings e_c
   - Use e_c as "query vector" in self-attention



TXtract: Taxonomy-Aware Attribute Value Extraction



TXtract: Taxonomy-Aware Attribute Value Extraction

flavor value “mint chocolate cookie”

BIOE Tags y_1 … y_T

CRF

Taxonomy-Aware Token Embeddings $\tilde{h}_1$ … $\tilde{h}_T$

CondSelfAtt

Token Embeddings h_1 … h_T

BiLSTM

Tokens x_1 … x_T

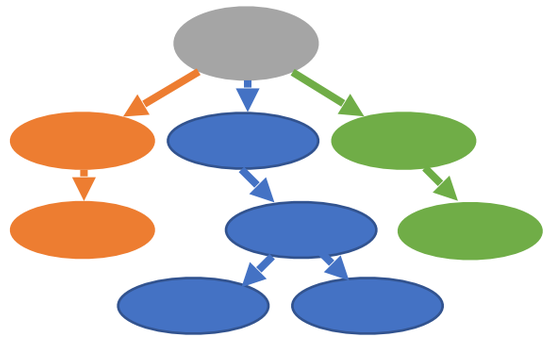
Product Text

“mint chocolate cookie ice cream 16 oz”

e_c

Category Embedding

CategoryEnc



c

Product Category

Improving Robustness Towards Noisy Category Assignments

- TXtract extracts attribute values conditionally on product categories

Improving Robustness Towards Noisy Category Assignments

- TXtract extracts attribute values conditionally on product categories

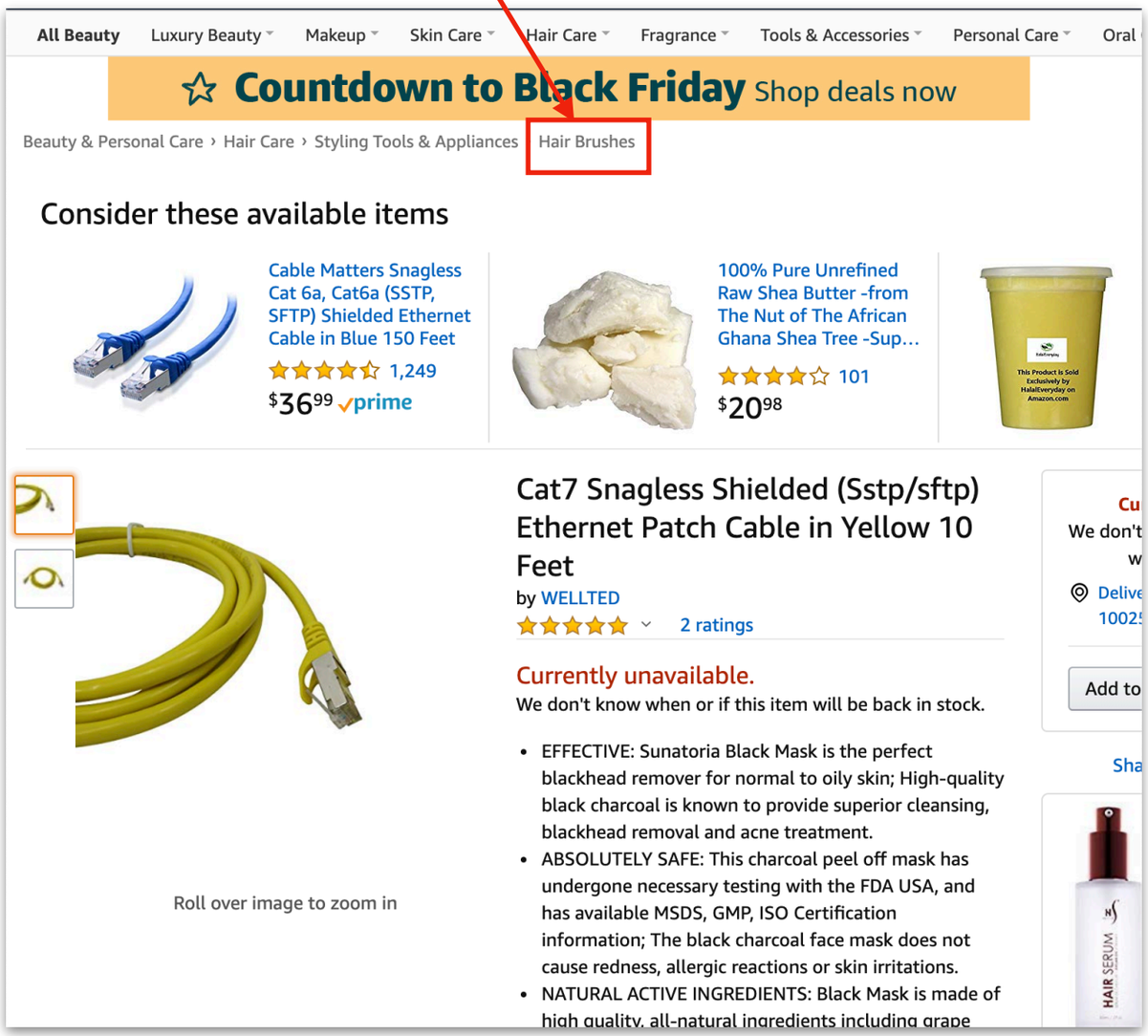
  (-) Issue: products may be assigned to **wrong** taxonomy nodes!

Improving Robustness Towards Noisy Category Assignments

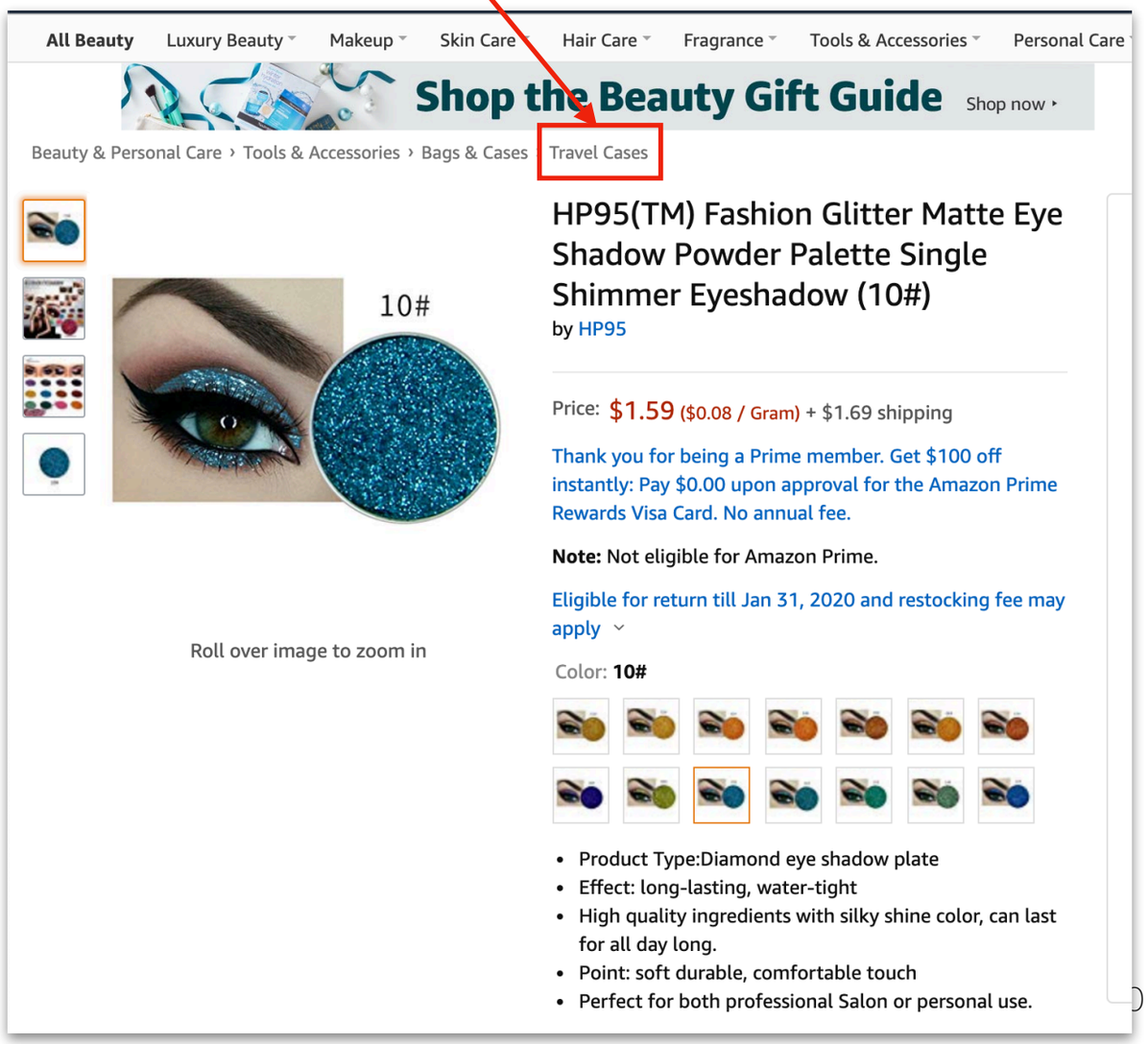
- TXtract extracts attribute values conditionally on product categories

  (-) Issue: products may be assigned to **wrong** taxonomy nodes!

Ethernet cable assigned under
Hair Brushes



Eyeshadow assigned under
Travel Cases



Improving Robustness Towards Noisy Category Assignments

- TXtract extracts attribute values conditionally on product categories
  - **(-) Issue:** products may be assigned to **wrong** taxonomy nodes!
  - **(-)** Conditioning on **wrong** categories -> **wrong** values

Improving Robustness Towards Noisy Category Assignments

- TXtract extracts attribute values conditionally on product categories
  - **(-) Issue:** products may be assigned to **wrong** taxonomy nodes!
  - **(-)** Conditioning on **wrong** categories -> **wrong** values

- How to make TXtract more **robust** towards wrong assignments?

Improving Robustness Towards Noisy Category Assignments

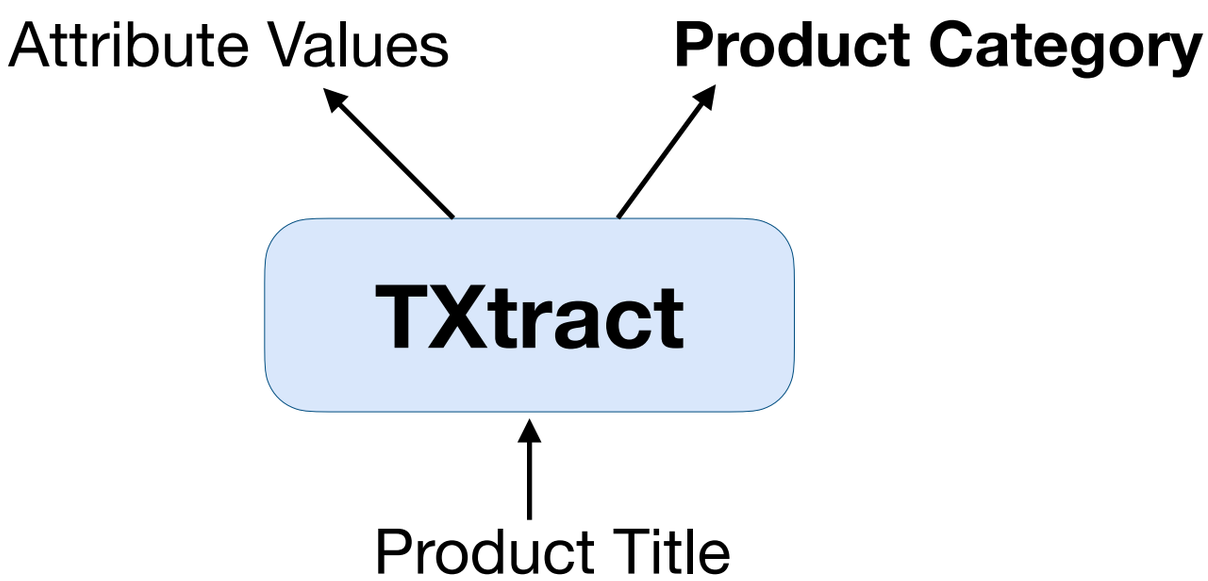
- TXtract extracts attribute values conditionally on product categories
  (-) Issue: products may be assigned to **wrong** taxonomy nodes!
  (-) Conditioning on **wrong** categories -> **wrong** values

- How to make TXtract more **robust** towards wrong assignments?
  - Idea: product **title** is indicative of product categories

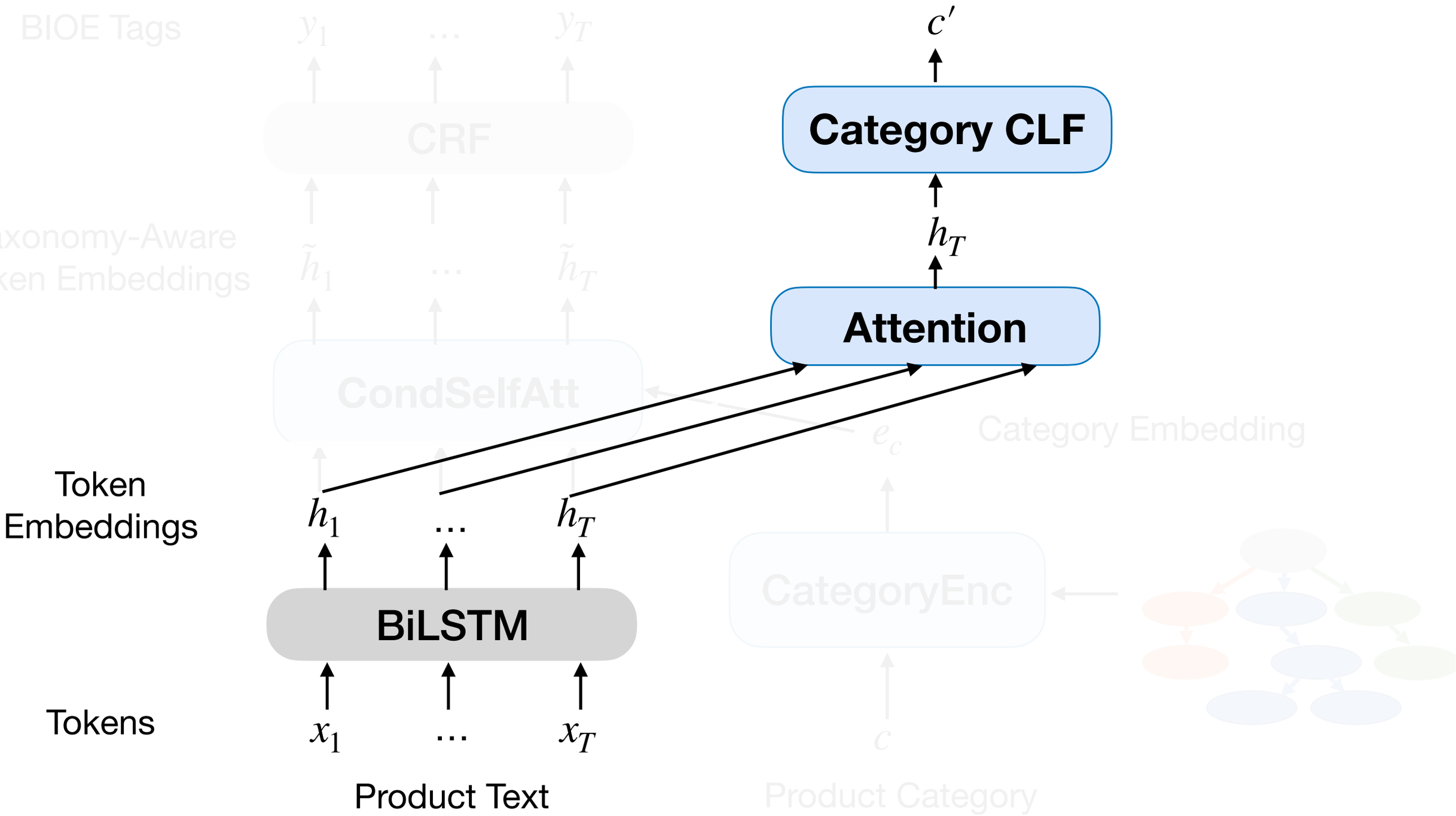
Improving Robustness Towards Noisy Category Assignments

- TXtract extracts attribute values conditionally on product categories
  - **(-) Issue:** products may be assigned to **wrong** taxonomy nodes!
  - **(-)** Conditioning on **wrong** categories -> **wrong** values

- How to make TXtract more **robust** towards wrong assignments?
  - Idea: product **title** is indicative of product categories
  - Auxiliary task: predict product **category** using product **title**



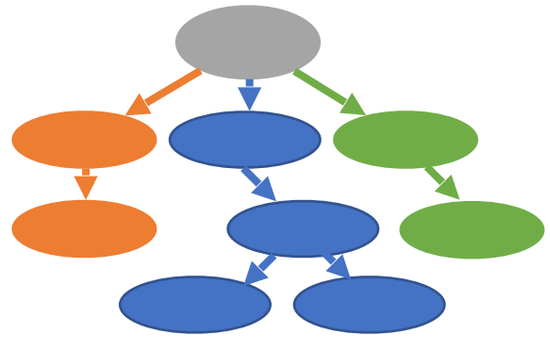
Taxonomy-Aware Product Category Prediction



Taxonomy-Aware Product Category Prediction

"Taxonomy-aware" loss function

"Correctly guess category AND ancestors"



c'

Category CLF

h_T

Attention

Token Embeddings

h_1 … h_T

BiLSTM

Tokens

x_1 … x_T

Product Text

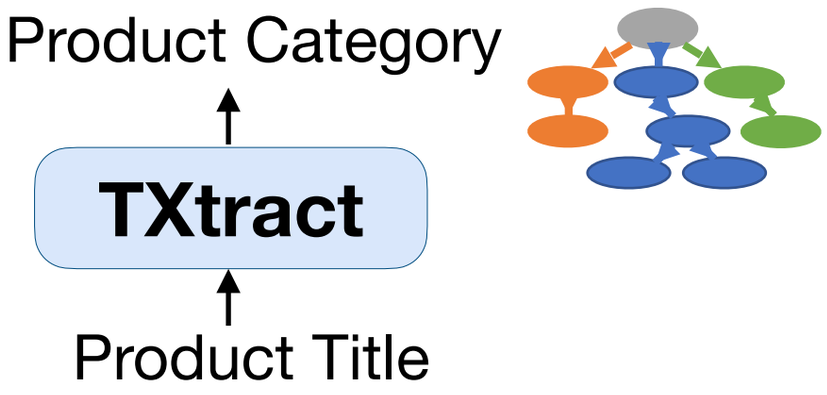
TXtract Summary - Multi-Task Training

- **Primary task:** taxonomy-aware attribute value extraction

TXtract Summary - Multi-Task Training

- **Primary task:** taxonomy-aware attribute value extraction

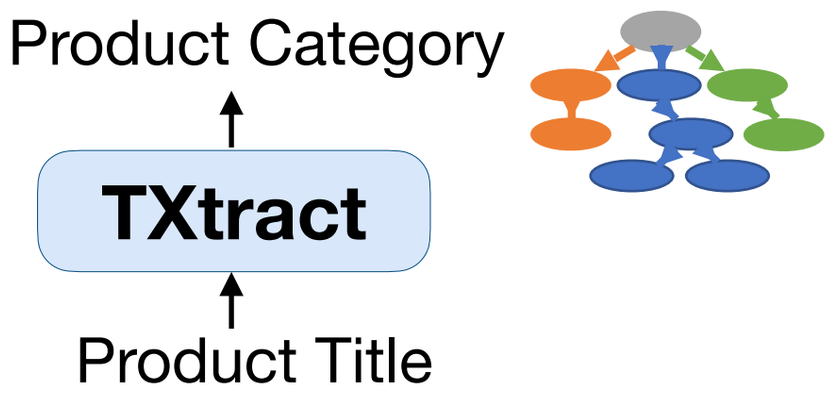
- **Auxiliary task:** taxonomy-aware category prediction



TXtract Summary - Multi-Task Training

- **Primary task:** taxonomy-aware attribute value extraction

- **Auxiliary task:** taxonomy-aware category prediction



- **Multi-task training:** shared text encoder (BiLSTM)
  - Token embeddings discriminative of categories
    (+) robustness to noisy assignments
    (+) more **effective** attribute value extraction

Outline

1. Attribute Value Extraction from Product Profiles

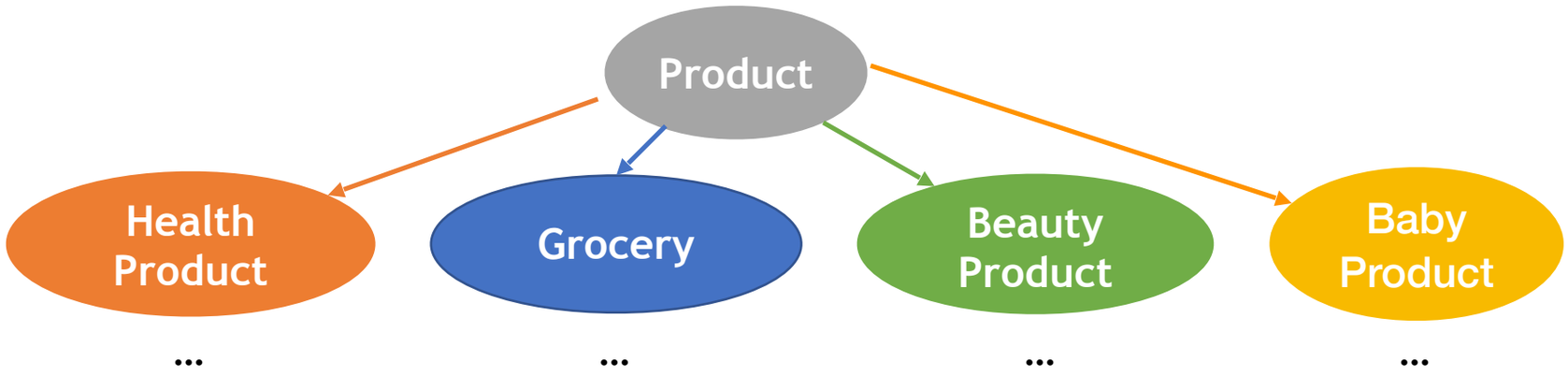
2. TXtract: Taxonomy-Aware Attribute Value Extraction

3. Experiments: Taxonomy with 4,000 Product Categories

4. Conclusions and Ongoing Work

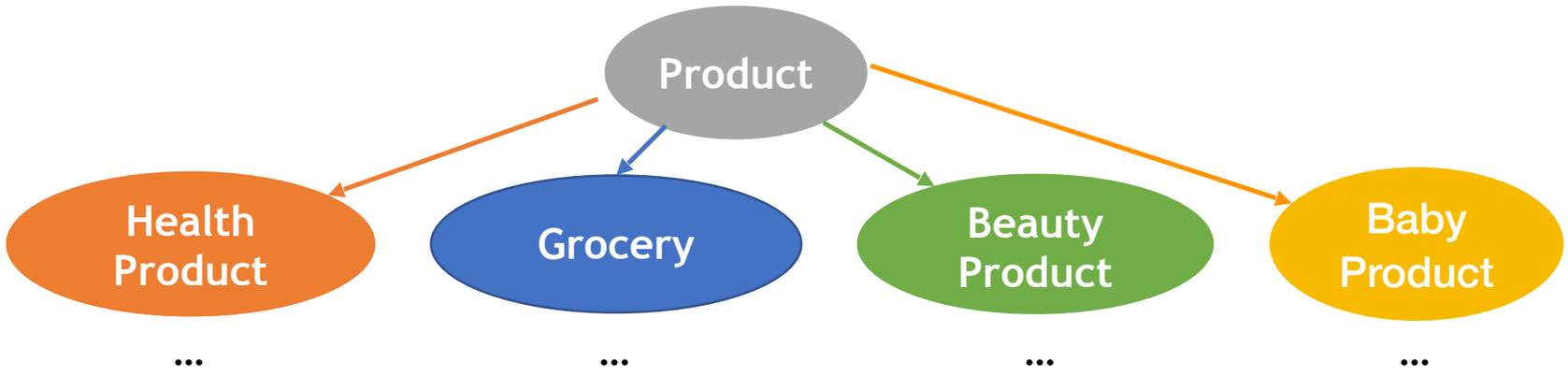
Experiments: Attribute Value Extraction

- **Dataset:**
  - 2 million products sampled from Amazon.com webpages
  - 4,000 categories sampled from Amazon's taxonomy (4 sub-trees)



Experiments: Attribute Value Extraction

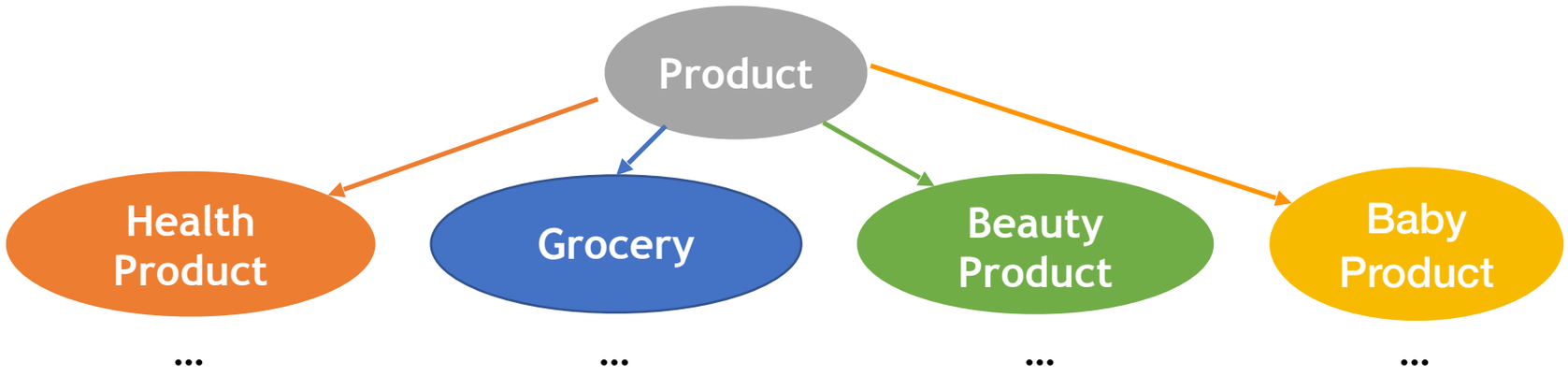
- **Dataset:**
  - 2 million products sampled from Amazon.com webpages
  - 4,000 categories sampled from Amazon's taxonomy (4 sub-trees)



- **Training** (60% of products)**:**
  - Train TXtract separately for 4 attributes: *flavor*, *scent*, *brand*, *ingredients*
  - **Input:** product title & descriptions
  - **Labels:** BIOE tags (generated using catalog values + distant supervision)

Experiments: Attribute Value Extraction

- **Dataset:**
  - 2 million products sampled from Amazon.com webpages
  - 4,000 categories sampled from Amazon's taxonomy (4 sub-trees)



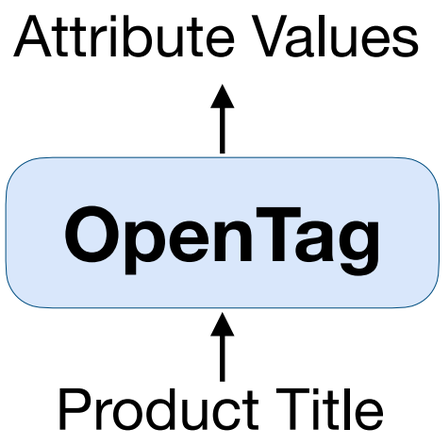
- **Training** (60% of products)**:**
  - Train TXtract separately for 4 attributes: *flavor*, *scent*, *brand*, *ingredients*
  - **Input:** product title & descriptions
  - **Labels:** BIOE tags (generated using catalog values + distant supervision)

- **Evaluation** (20% of held-out products):
  - Value Vocabulary (#distinct extracted values)
  - Coverage (#products with extracted values)
  - Product-level extraction metrics (Micro F1/Precision/Recall)
  - Category-level extraction metrics (Macro F1/Precision/Recall)

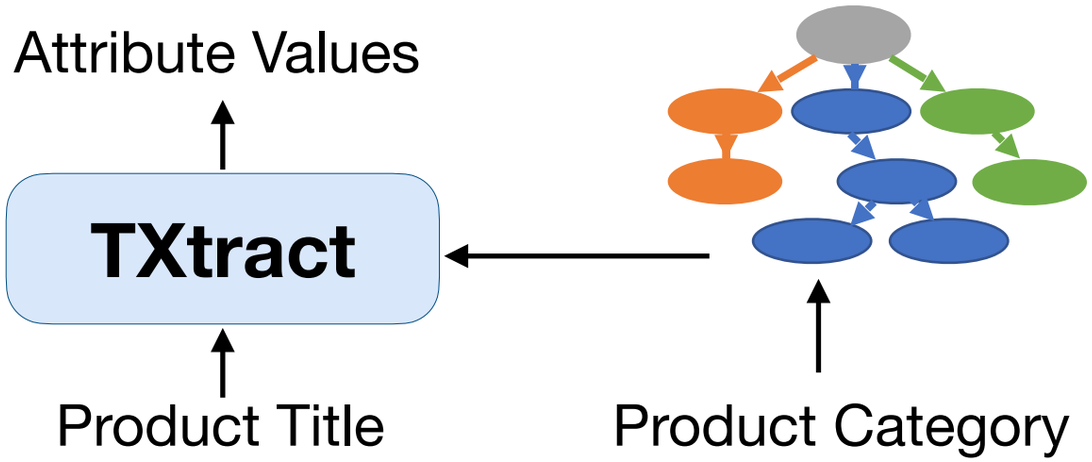
Model Comparison

- TXtract vs OpenTag [Zheng et al., KDD'18]
  - **Same** model (BiLSTM, CRF)

OpenTag **ignores** categories



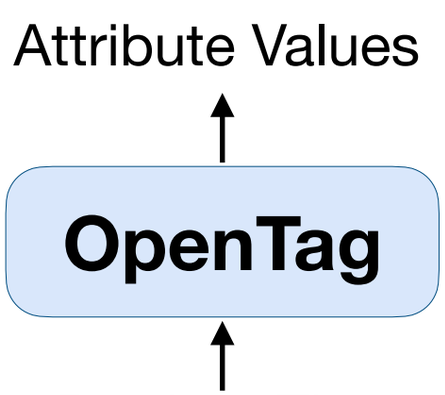
TXtract **considers** categories



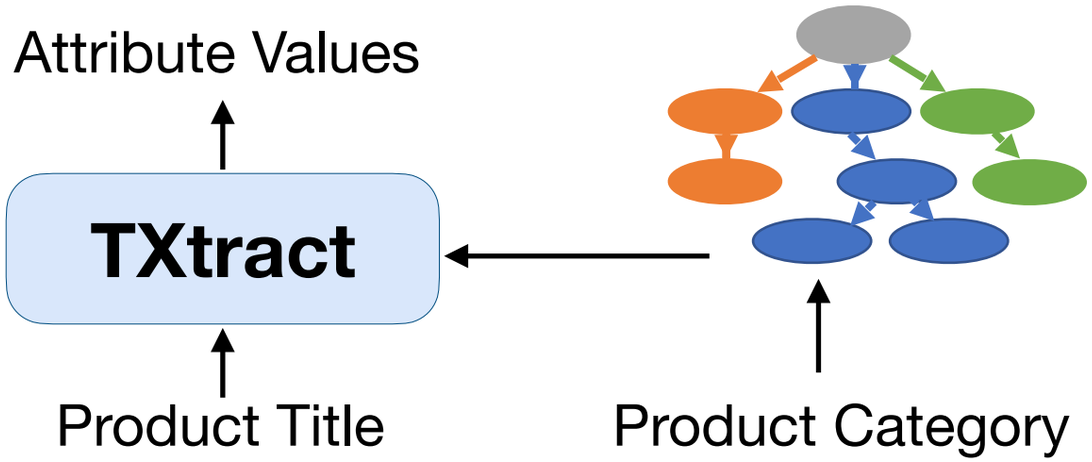
Model Comparison

- TXtract vs OpenTag [Zheng et al., KDD'18]
  - **Same** model (BiLSTM, CRF)

OpenTag **ignores** categories



TXtract **considers** categories



- TXtract vs **category-aware** approaches (from other domains & tasks)
  - Append artificial tokens in title [Johnson et al., TACL'17]
  - Concatenate category embeddings to word/BiLSTM embeddings
  - "Gating" layer [Cho et al., EMNLP'14] [Ma et al., KDD'19]
  - "Flat" multi-task learning

Leveraging the Taxonomy Improves Extraction

- Average performance across **ALL** categories & attributes:

	Coverage (%)	**Micro F1 (%)**	**Macro F1 (%)**
OpenTag	73.0	56.8	46.6
TXtract	**81.6** (**+11.7%**)	**60.4** (**+6.2%**)	**49.7** (**+10.4%**)

  - TXtract outperforms OpenTag across 4,000 categories

Leveraging the Taxonomy Improves Extraction

- Average performance across **ALL** categories & attributes:

	Coverage (%)	**Micro F1 (%)**	**Macro F1 (%)**
OpenTag	73.0	56.8	46.6
TXtract	**81.6** (**+11.7%**)	**60.4** (**+6.2%**)	**49.7** (**+10.4%**)

  - TXtract outperforms OpenTag across 4,000 categories

- TXtract improves OpenTag for **each** attribute & training configuration
  - up to **15%** improvement in Coverage
  - up to **10%** improvement in Micro F1

- We show the contribution of each TXtract component (ablation study)

TXtract Extracts Category-Specific Values

- TXtract extracts **44%** more (distinct) values than OpenTag
- TXtract extracts **category-specific** values:

Title: *"Controlled Labs Purple Wraath 90 Servings - Purple Lemonade"*
Category: "Vitamins"

- OpenTag (flavor): **(empty)**
- TXtract (flavor): **"purple lemonade"**

Title: *"Matte Eye Shadow Powder Palette Single Shimmer Eyeshadow"*
Category: "Eyeshadow"

- OpenTag (flavor): **"palette"**
- TXtract (flavor): **(empty)**

Title: *"Click - Espresso Protein Drink Vanilla Latte - 16 oz."*
Category: "Sports Nutrition"

- OpenTag (flavor): **"espresso"**
- TXtract (flavor): **"vanilla latte"**

Outline

1. Attribute Value Extraction from Product Profiles

2. TXtract: Taxonomy-Aware Attribute Value Extraction

3. Experiments: Taxonomy with 4,000 Product Categories

4. Conclusions and Ongoing Work

Attribute Value Extraction - Scaling Up to Thousands of Product Categories

- Product domain is challenging!
  - Diverse categories

  - Assignments to wrong categories

Attribute Value Extraction - Scaling Up to Thousands of Product Categories

- Product domain is challenging!
  - Diverse categories

  - Assignments to wrong categories

- TXtract: **hierarchical** taxonomies with **thousands** of categories

  (+) Efficient: single model trained on all categories in parallel

  (+) Effective:

  - Leverages taxonomy using conditional self-attention & multi-task learning
  - Improves extraction quality (e.g., up to 15% higher coverage)

Towards Better, Large-Scale Product Understanding

TXtract

flavor: “chocolate”

ingredients: “argan oil”, …

Towards Better, Large-Scale Product Understanding

Building an “automatic”
knowledge graph of products

TXtract

flavor: “chocolate”

ingredients: “argan oil”, …

Towards Better, Large-Scale Product Understanding

TXtract

flavor: “chocolate”

ingredients: “argan oil”, …

product graph

“Alexa, which shampoos contain argan oil?”

Thank you!

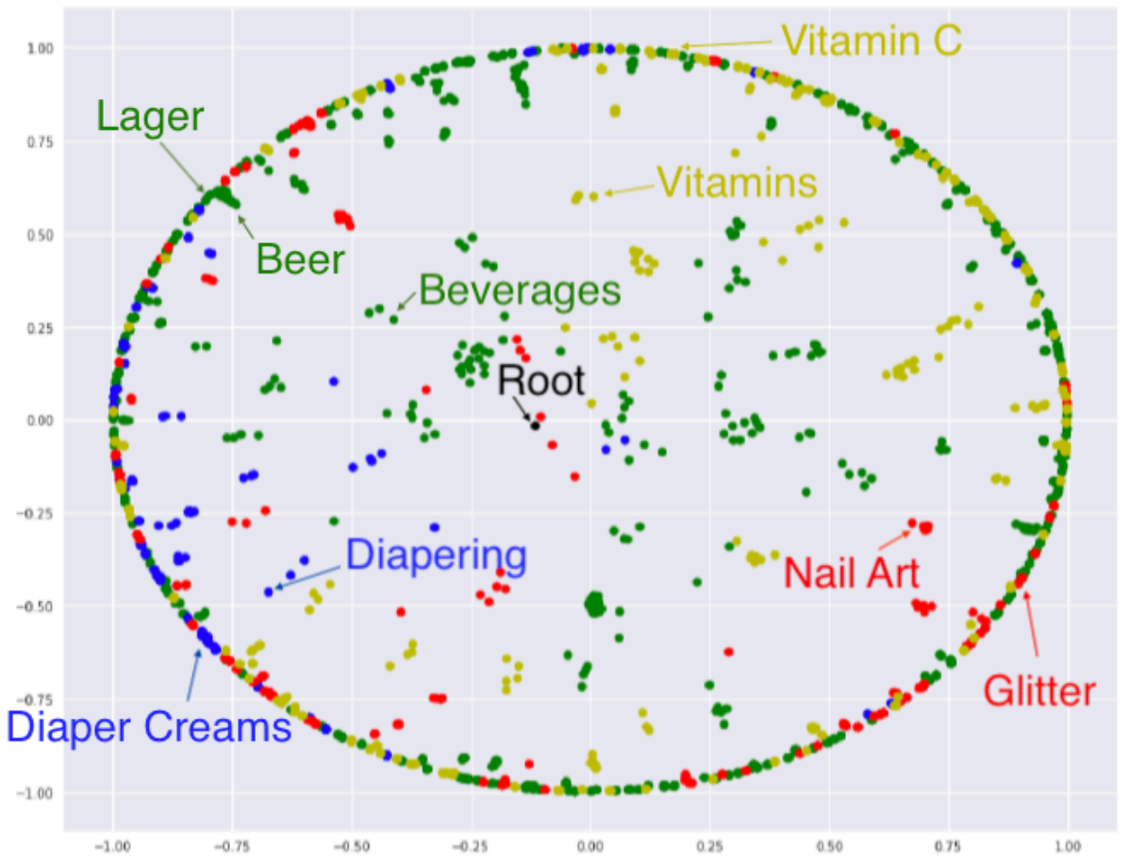
Contact
gkaraman@cs.columbia.edu
https://gkaramanolakis.github.io

COLUMBIA
UNIVERSITY

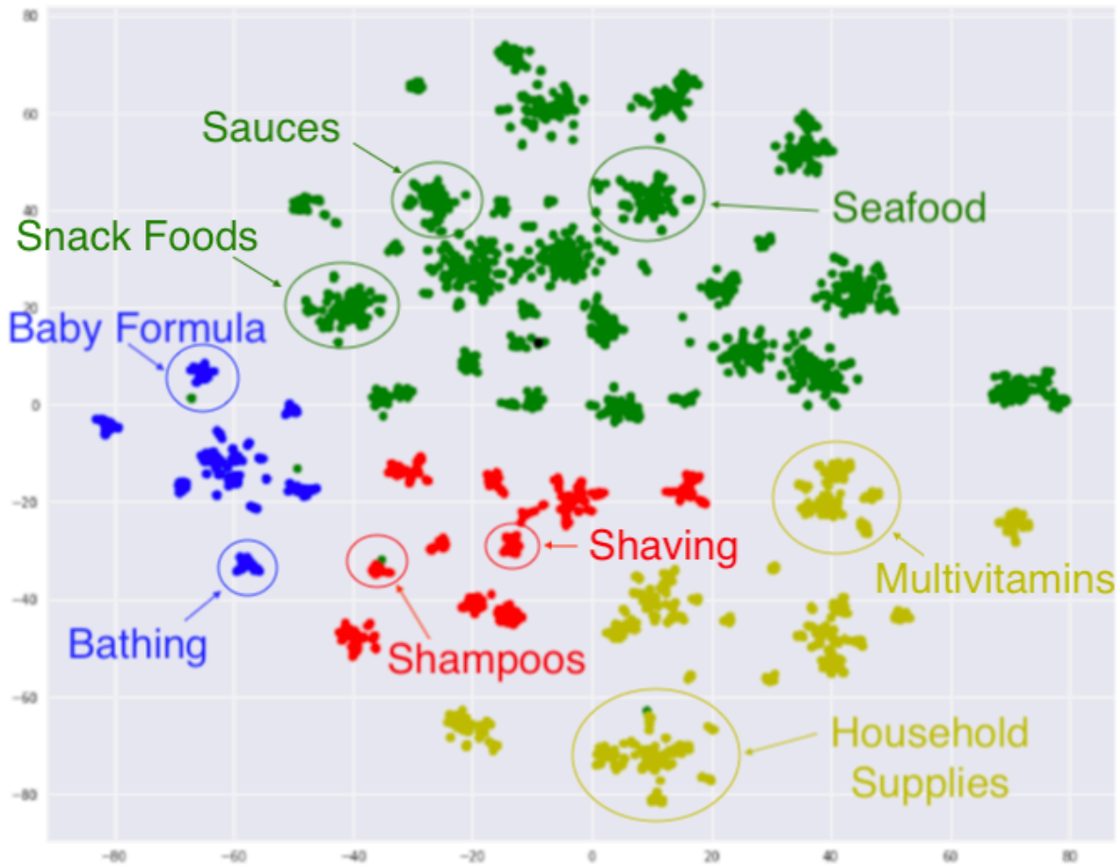
Extra Slides

Leveraging Hierarchical Product Categories in TXtract

1. Category Encoder: generates category **embeddings** e_c
   - Represent related categories as similar vectors
   - Captures **hierarchical** structure of categories
   - … using Poincaré embeddings



2D Poincaré Disk



50D Poincaré Ball $\xrightarrow{t\text{-}SNE}$ *2D Euclidean Space*

Ablation Study - How to Leverage Product Categories?

- We show the contribution of each component in TXtract:

Model	TX	MT	Micro F1
OpenTag	-	-	57.5
Title+id	✓	-	55.7 ↓3.1%
Title+name	✓	-	56.9 ↓1.0%
Title+path	✓	-	54.3 ↓5.6%
Concat-wemb-Euclidean	✓	-	60.1 ↑4.5%
Concat-wemb-Poincaré	✓	-	60.6 ↑5.4%
Concat-LSTM-Euclidean	✓	-	60.1 ↑4.5%
Concat-LSTM-Poincaré	✓	-	60.8 ↑5.7%
Gate-Poincaré	✓	-	60.6 ↑5.4%
CondSelfAtt-Poincaré	✓	-	61.9 ↑7.7
MT-flat	-	✓	60.9 ↑5.9%
MT-hier	-	✓	61.5 ↑7.0%
Concat & MT-hier	✓	✓	62.3 ↑8.3%
Gate & MT-hier	✓	✓	61.1 ↑6.3%
CondSelfAtt & MT-hier	✓	✓	**63.3** ↑10.1%

Category Prediction	AUPR	F1	Prec	Rec
Flat	0.61	53.9	74.2	48.0
Hierarchical	**0.68**	**62.7**	**80.4**	**56.9**

Experiments: Attribute Value Extraction

- TXtract improves OpenTag on **each** attribute and training config:

Attr.	Model	Vocab	Cov	Micro F1
Flavor	OpenTag	6,756	73.2	57.5
	TXtract	**13,093**	**83.9** ↑14.6%	**63.3** ↑10.1%
Scent	OpenTag	10,525	75.8	70.6
	TXtract	**13,525**	**83.2** ↑9.8%	**73.7** ↑4.4%
Brand	OpenTag	48,943	73.1	63.4
	TXtract	**64,704**	**82.9** ↑13.4%	**67.5** ↑6.5%
Ingred.	OpenTag	9,910	70.0	35.7
	TXtract	**18,980**	**76.4** ↑9.1%	**37.1** ↑3.9%
Average relative increase			↑11.7%	↑6.2%