OA-Mine: Open-World Attribute Mining for E-Commerce Products with Weak Supervision

[1]Xinyang Zhang, [2]Chenwei Zhang, [2]Xian Li, [3]Xin Luna Dong, [4]Jingbo Shang, [5]Christos Faloutsos and [1]Jiawei Han

[1]University of Illinois at Urbana-Champaign [2]Amazon, Inc. [3]Meta (Facebook)

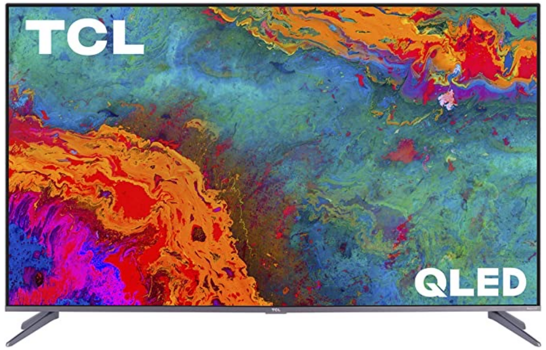
[4]University of California San Diego [5]Carnegie Mellon University

Presented by Xinyang

April 29, 2022

github.com/xinyangz/oamine

What is Product Attribute Mining?



BRAND SCREEN-SIZE PROD-LINE RESOLUTION HDR-COMPATABILITY

TCL 50-inch 5-Series 4K UHD Dolby Vision HDR QLED Roku Smart TV, Black

PANEL OPERATING-SYSTEM COLOR

With Deal: $449.00

Screen Size 50 Inches

Brand TCL

- Superior 4K Ultra HD: Picture clarity combined with the contrast, color, and detail of Dolby Vision HDR (High Dynamic Range) for the most lifelike picture
- QLED: Quantum dot technology delivers better brightness and wider color volume, Panel Resolution :3840 x 2160, Viewable Display Size: 49.5 inch

- ❑ Given product text
- ❑ Extract
  - ❑ Attribute (types). E.g., “resolution”
  - ❑ Values. E.g., “4K UHD”

What is Open-World and Why?

- The set of attributes (types) and values are not known beforehand
- Want to find new attributes and new values

	Attribute	Value
Prior work (NER)	Closed-world	Open-world
OA-Mine	Open-world	Open-world

- Why?
  - Existing types of products may get new attributes
    - E.g., TV, HDR compatibility not seen 10 years ago
  - New types of products may emerge
    - E.g., VR headsets not seen 10 years ago

Weak Supervision

- Full supervision is expensive and infeasible
  - E-commerce products expand every day
- Our supervision: seed examples
  - Give a few known attribute values, for each known product type
  - Example:
    - Tea: [[loose leaf, tea bag], [green tea, black tea]]
    - Coffee: [[whole bean, k-cup], [dark roast, light roast]]

Problem Setting

- Input
  - **Product data**: product text + product type
    - E.g., tea product: “Two Leaves and a Bud Organic Peppermint Herbal Tea Bags…”
  - **Weak supervision**: seed attribute values for a few known types
    - E.g., {tea: [[green tea, black tea], [loose leaf, tea bag]], coffee: [[whole bean, k-cup]]}
- Output
  - New attribute types and values

PT = Tea

(Item Form)
Loose leaf
Tea bag
Sachet
…

(Type)
Green tea
Black tea
Oolong tea
…

…

PT = Coffee

(Item Form)
Whole bean
K-cup
Sachet
…

(Flavor)
Cinnamon
Vanilla
Pumpkin hazelnut
…

…

…

Our Contributions

- New problem:
  - *Open-world* attribute mining
  - Weak supervision
- New data:
  - Amazon data with human annotations
- New solution:
  - A principled framework w/ a focus on attribute-aware representation learning.

Our Dataset

- 80.6K Amazon products from 100 product types
- Development set
  - Covers all 100 product types
  - Labels derived from Amazon product profiles
- Test set
  - Covers 1,943 products from 10 product types
  - Each labeled by 5 MTurk workers
  - Consolidated by expert knowledge associates

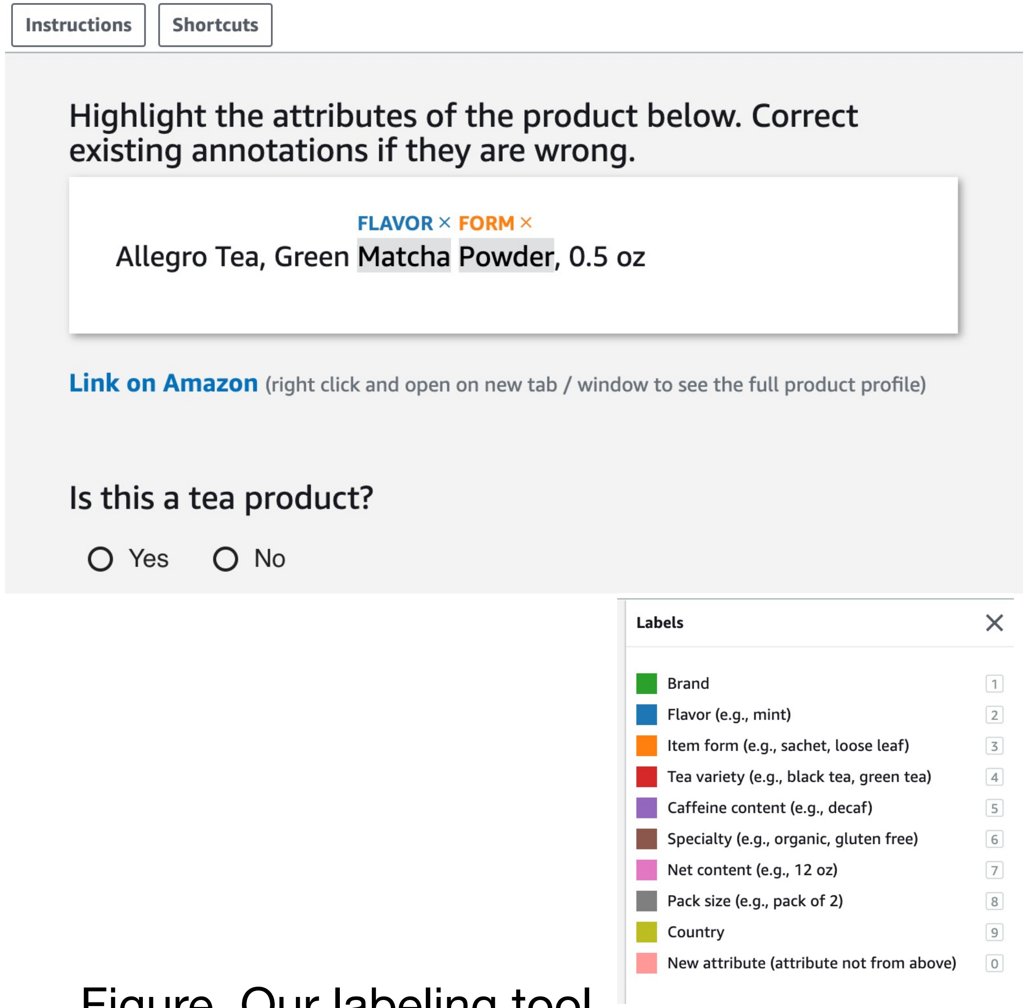
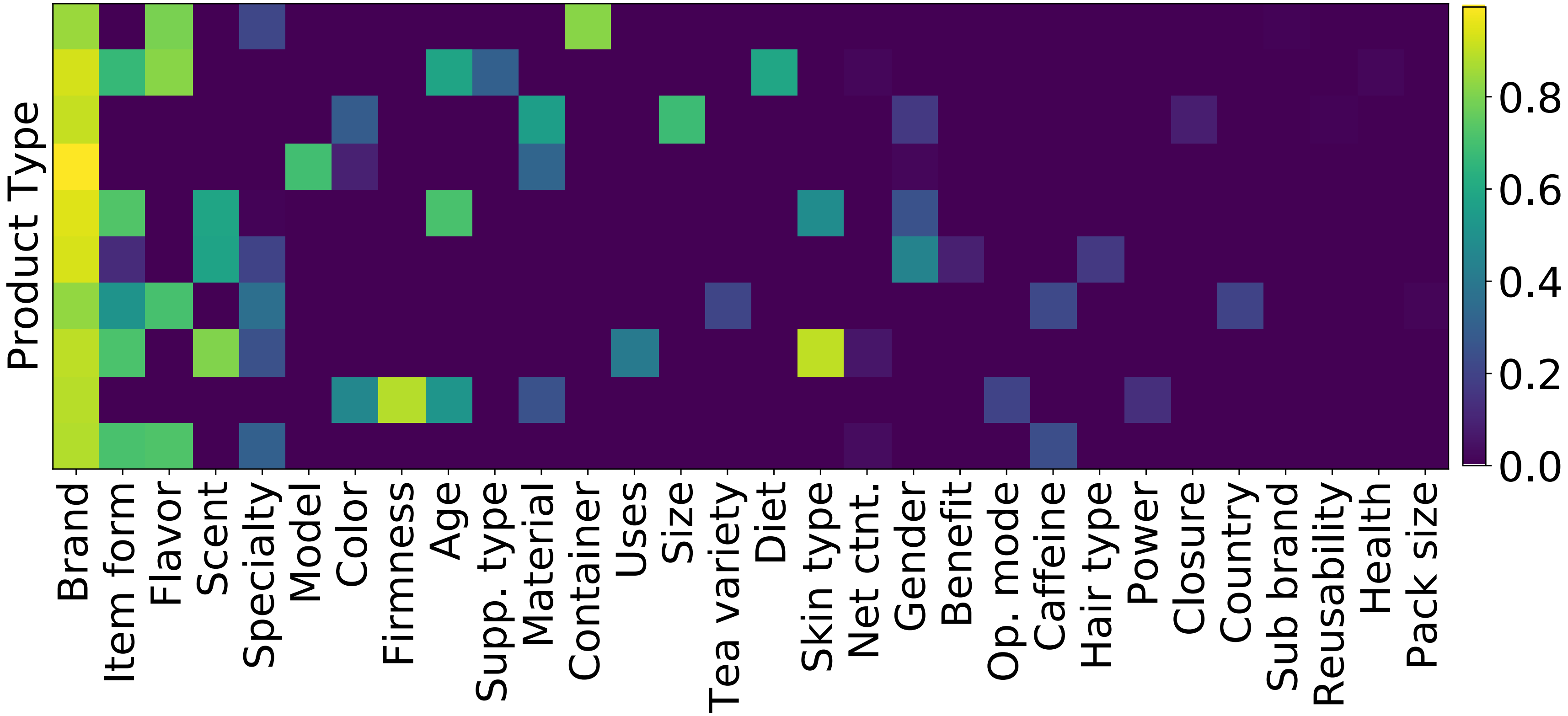


Figure. Our labeling tool

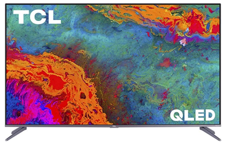
Why Open-World Attribute Mining? (cont')

- Attributes and values missing from the catalog
  - Humans found 51 attributes, 21 are missing
  - For the 30 attributes found in the catalog, 60% values missing



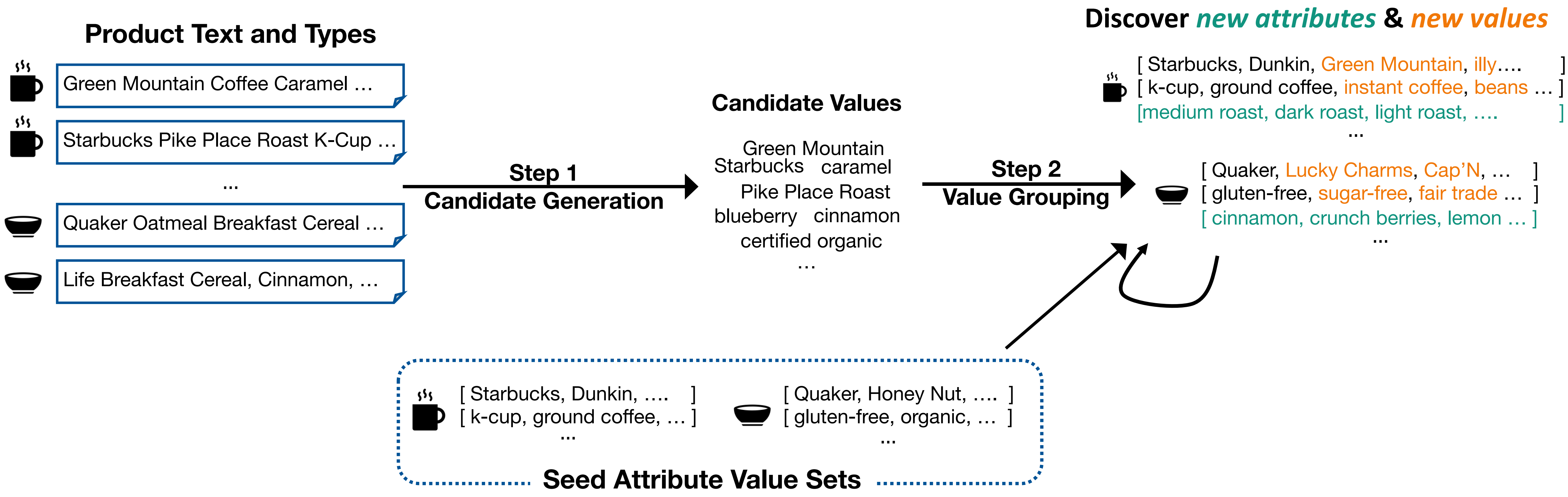
Observation from Data

- Observation 1 (title first)
  - To maximize exposure of products to customers, sellers usually pack the highlights of their product in the title
- Observation 2 (bag-of-values)
  - A product title rarely contains irrelevant information, and is a collection of attribute values
- Observation 3 (value exclusiveness)
  - With limited space in the title, the values seldom repeat



TCL 50-inch 5-Series 4K UHD Dolby Vision HDR QLED Roku Smart TV, Black

Framework Overview



Step 1: Attribute Value Candidate Generation

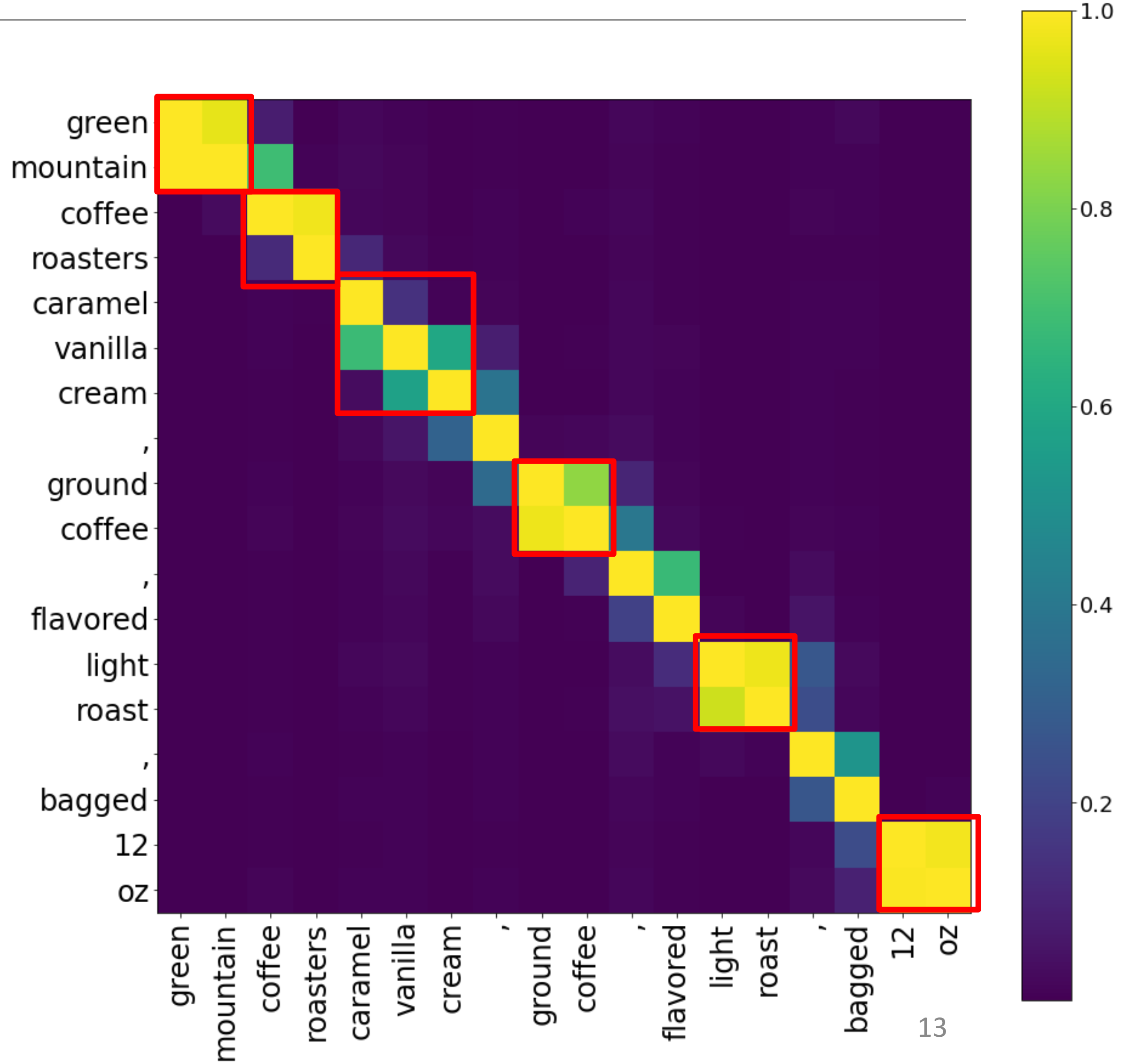
Attribute Value Candidate Generation: Goal

- Goal: obtain candidate attribute values from products with *high recall*
- Example
  - **Input**: Green Mountain Coffee Roasters Caramel Vanilla Cream, Ground Coffee, Flavored Light Roast, Bagged 12 oz
  - **Output**: "Green Mountain Coffee Roasters", "Caramel Vanilla Cream", "Ground Coffee", "Bagged", "12oz"

Method: Title Segmentation from Perturbed Masking

- Idea: pre-trained LM should capture word to word impact [1-3]
- Steps:
  - Language model fine-tuning
  - Build a word to word impact matrix
  - Chunk out attribute candidates based on scores in the matrix

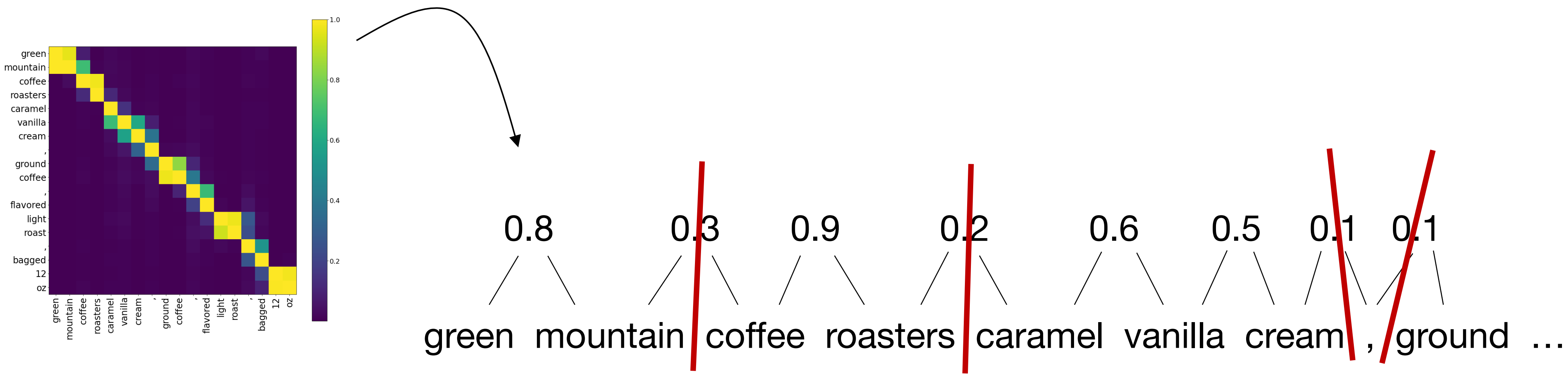
$$s(w_i, w_{i+1}) = d(\mathrm{BERT}(W/\{w_i\})_i, \mathrm{BERT}(W/\{w_i, w_{i+1}\})_i)$$



[1] Wu, Zhiyong, et al. "Perturbed masking: Parameter-free probing for analyzing and interpreting bert." ACL (2020)
[2] Kim, Taeuk, et al. "Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction." ICLR (2020)
[3] Gu, Xiaotao, et al. "UCPhrase: Unsupervised Context-aware Quality Phrase Tagging." *KDD* (2021).

Method: Title Segmentation from Perturbed Masking (cont')

- Chunking attribute values from the impact matrix
  - We use chunking based on impact scores of *adjacent tokens*. If score < threshold, we do a split.



Quantitative Results

Table 1: Evaluation on Attribute Value Candidate Generation. Methods are divided into pre-trained, distantly supervised, and unsupervised, from top to bottom.

Methods	Entity-Prec.	Entity-Rec.	Entity-F1	Corpus-Rec.
spaCy [7]	31.19	19.15	23.73	50.02
FlairNLP [1]	34.81	24.33	28.64	52.17
AutoPhrase [13]	26.58	29.67	28.04	32.39
UCPhrase [6]	35.01	19.66	25.18	37.50
OA-Mine	**42.53**	**53.29**	**47.30**	**64.10**

Step 2: Attribute Value Grouping

Value Grouping Goal

- Goal: group values into attributes with seed as guidance

PT = coffee

PT = tea

caramel vanilla cream

light roast

ground coffee

…

unsweetened

black tea

light roast
medium roast
dark roast
medium dark roast
…

vanilla
pumpkin
cinnamon
salted caramel
…

black tea
green tea
macha tea
…

bottled
sachet
packs
…

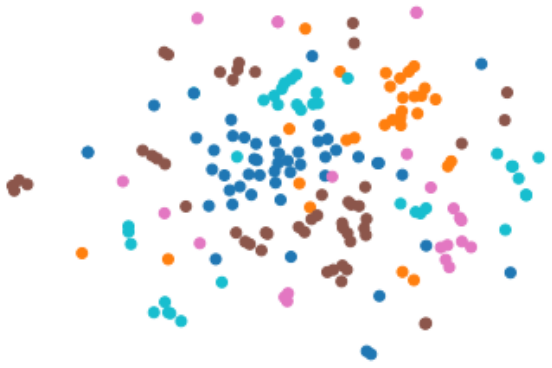
Seed (known attribute values): light roast, black tea, …

Value Grouping Overall Idea and Challenges

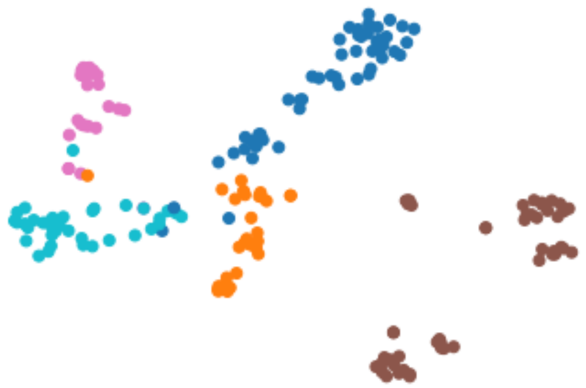
- **Overall idea**: clustering on value candidates
- **Challenge**:
  - Pre-trained BERT is not attribute-aware
  - Generalization to new attributes and product types
    - Some attributes may not have human given seed values
  - Noise from candidate generation

Problem with BERT Embedding for Attribute Grouping

- Why not BERT + clustering?
  - Distance metric between two embedding vectors does not fully capture attribute information



BERT + MLM



Our Fine-tuning

- Need to make phrase embedding attribute aware

Attribute-Aware Fine-Tuning

Value Candidates

Green Mountain
Starbucks caramel
Pike Place Roast
blueberry cinnamon
certified organic
…

Seed Attribute Value Sets

[Starbucks, Dunkin, ….]
[k-cup, ground coffee, …]
…

Unlabeled Data + Value Exclusiveness

T1: Binary Meta-Classification

(Dunkin, Starbucks) → same_attr
(Quaker, organic) → diff_attr

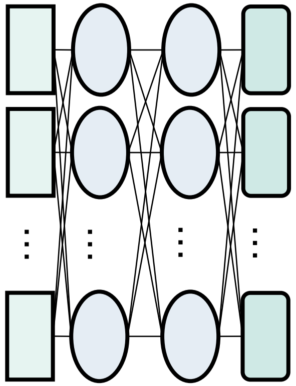
T2: Contrastive Learning

d(Dunkin, Starbucks) < d(Dunkin, organic)

T3: Multiclass Classification

Starbucks [SEP] coffee → coffee_brand
organic [SEP] cereal → cereal_specialty

Shared BERT Encoder



Multitask Fine-Tuning

$\mathcal{L}_{\text{binary}}$
$\mathcal{L}_{\text{contrastive}}$
$\mathcal{L}_{\text{classification}}$

Attribute-Aware Fine-Tuning: Model & Objectives

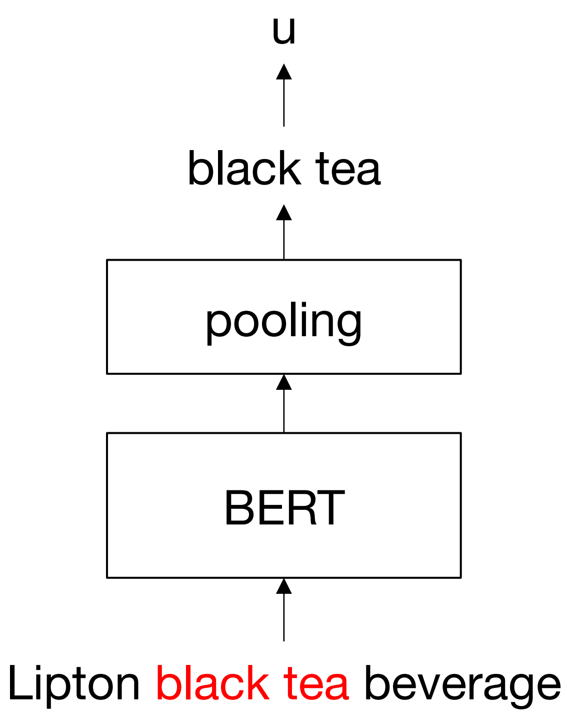
- Shared encoder: BERT + entity pooling
- Objectives

$$\mathcal{L}_{\text{binary}} = \sum_{(u,v)\in P} \left\|1 - f(u,v)\right\|^2 + \sum_{(u,v)\in N} \left\|-1 - f(u,v)\right\|^2$$

$$\mathcal{L}_{\text{contrastive}} = \sum_{(v_a, v_p, v_n)} \max\left(\left\|f(v_a, v_p)\right\|^2 - \left\|f(v_a, v_n)\right\|^2 + \alpha, 0\right)$$

$$\hat{\boldsymbol{y}} = \text{Softmax}(\text{Linear}(\text{BERT}(W[\text{SEP}]t)))$$

$$\mathcal{L}_{\text{classification}} = \text{CrossEntropy}(\hat{\boldsymbol{y}}, \boldsymbol{y})$$



Self-Ensemble Inference & Iterative Training

- Attribute discovery & noise handling: DBSCAN
  - Discover attribute value cluster by local density
  - Generates a large noise cluster
- Improving recall: classifier
  - Use the classifier to pick values back from noise cluster to discovered attributes
- Iterative training
  - Confident predictions from one iteration is used to train the next iteration
  - Benefit: next iteration will have a more complete set of attributes for training

Main Experiments

Table 2: End-to-end evaluation on development and test data. Results are average of 3 runs. Bold faced numbers indicate statistically significant results from t-test with 99% confidence.

		Dev Set (100 product types)				Test Set (10 product types)			
Method Type	**Method**	ARI	Jaccard	NMI	Recall	ARI	Jaccard	NMI	Recall
Sequence tagging (closed-world)	BiLSTM-Tag	0.299	0.354	0.422	0.565	0.175	0.219	0.374	0.162
	OpenTag [22]	0.244	0.324	0.334	0.593	0.160	0.247	0.357	0.165
	SU-OpenTag [18]	0.637	0.598	0.607	0.525	0.411	0.340	0.542	0.162
Unsupervised clustering	BERT+AG-Clus	0.249	0.446	0.585	0.742	0.386	0.308	0.504	**0.430**
	BERT+DBSCAN	0.133	0.146	0.507	0.131	0.385	0.412	0.575	0.186
Weakly sup. clustering	DeepAlign+ [21]	0.175	0.226	0.336	0.729	0.257	0.208	0.426	0.389
	OA-Mine (no multitask)	0.671	0.634	0.610	0.458	0.601	0.518	0.733	0.225
	OA-Mine	**0.704**	**0.689**	**0.629**	0.747	**0.712**	**0.650**	**0.781**	0.275

Generalization to New Attributes

- Training: hold out 20% attributes
- Evaluation: on held out attributes
- 5-fold cross validation

Table 3: Performance on discovering new attributes. Experiment conducted with 5-fold cross-validation, where each fold holds out 20% attributes from training.

Methods	ARI	Jaccard	NMI	Recall
BERT+AG-Clus	0.215	0.372	0.308	0.832
BERT+DBSCAN	0.199	0.431	0.129	0.370
DeepAlign+	0.192	0.329	0.303	0.831
OA-Mine	0.599	0.743	0.489	0.688

Generalization to New Attributes (cont')

Table 4: Comparing model predictions on unseen attributes during cross-validation. Red is error.

Attribute	Method	Predicted Cluster
Coffee Brand	BERT+AG-Clus	green mountain, folgers, coffee fool, maxwell house, coffee roasters, nescafe, eight o clock, ...
	DeepAlign+	gourmet, keurig brewers, starbucks, green mountain coffee, donut, dunkin donuts, ...
	OA-Mine	starbucks, green mountain, folgers, coffee fool, maxwell house, nescafe, san marco coffee, ...
Laundry Detergent Form	BERT+AG-Clus	powder, bottle, pacs, original, 2, pods, 32 loads, ...
	DeepAlign+	liquid, laundry, wash, pack, stain, natural, ...
	OA-Mine	liquid, powder, bottle, spray, carton, pods, soap, ...

Generalization to Product Types w/o Seed

- Training: 90 product types
- Evaluation: 10 new product types

Table 5: Performance on new product types. Models tested on product types not seen during training.

Methods	ARI	Jaccard	NMI	Recall
BERT+AG-Clus	0.386	0.308	0.504	0.430
BERT+DBSCAN	0.385	0.412	0.575	0.186
OA-Mine	0.658	0.609	0.702	0.231

Summary

- New problem:
  - *Open-world* attribute mining
  - Weak supervision
- New data:
  - Amazon data with human annotations for E2E evaluation
- New solution:
  - Attribute value candidate generation w/ LM
  - Value grouping with attribute-aware fine-tuning and self-ensemble inference

Thank you!