Active Learning for Product Type Ontology Enhancement in E-commerce

Yun Zhu
The Home Depot
Atlanta, GA
yun_zhu@homedepot.com

Sayyed M. Zahiri
The Home Depot
Atlanta, GA
sayyed_m_zahiri@homedepot.com

Jiaqi Wang
The Home Depot
Atlanta, GA
jiaqi_wang@homedepot.com

Han-Yu Chen
The Home Depot
Atlanta, GA
hanyu_chen@homedepot.com

Faizan Javed
The Home Depot
Atlanta, GA
faizan_javed@homedepot.com

ABSTRACT

Entity-based semantic search has been widely adopted in modern search engines to improve search accuracy by understanding users' intent. In e-commerce, an accurate and complete product type (PT) ontology is essential for recognizing product entities in queries and retrieving relevant products from catalog. However, finding product types (PTs) to construct such an ontology is usually expensive due to the considerable amount of human efforts it may involve. In this work, we propose an active learning framework that efficiently utilizes domain experts' knowledge for PT discovery. We also show the quality and coverage of the resulting PTs in the experiment results.

CCS CONCEPTS

• **Information systems** → **Ontologies**; • **Computing methodologies** → **Information extraction**.

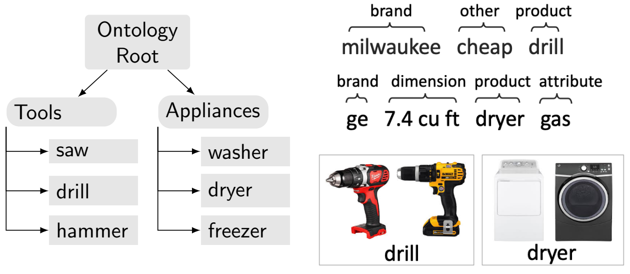
KEYWORDS

knowledge graph, ontology, active learning, semantic search, e-commerce

ACM Reference Format:
Yun Zhu, Sayyed M. Zahiri, Jiaqi Wang, Han-Yu Chen, and Faizan Javed. 2020. Active Learning for Product Type Ontology Enhancement in E-commerce. In *KDD '20: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Aug 21–26, 2020, San Diego, CA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In the past few decades, knowledge graph construction and applications have been rapidly developed and achieved significant outcomes. For better relevancy in web search, Google has been leveraging knowledge graph that represents real-world entities and their relationships to one another since 2012[12]. To identify those entities from text, named entity recognition (NER) techniques have been extensively studied and applied in many areas [9, 17] including e-commerce search [14, 15]. Such NER systems usually work with a well defined ontology to classify tokens in a sequence of words [4, 10]. A comprehensive and domain-specific PT ontology is beneficial to product search and discovery in an e-commerce platform [5, 7]. At The Home Depot (THD), PT ontology has been used tremendously by the online search to improve query understanding and product retrieval. For example, Figure 1a shows a snippet of our PT ontology that consists of known PT classes. The PTs in the ontology serve as the entity reference for the NER task (Figure 1b top) as well as the classes for SKU-PT mapping (1b bottom) on the catalog side that facilitates the retrieval of relevant products.



(a) (b)

Figure 1: Example of PT Ontology (a), NER (b, top) and SKU-PT mapping (b, bottom)

Discovering valid PTs is a key task to build or expand a PT ontology with a fundamental challenge regarding the definition of a PT. A PT can be defined from the demand side as atomic keywords/phrase that describes what customers look for [5] or from the supply side as a semantic tag/label that uniquely identifies a product. Within THD, we also have practical guidelines to distinguish between valid and invalid PTs like (i) no common attributes like color, brand, material, style etc in PTs (e.g., <u>*stainless steel*</u> *screw*, <u>*white*</u> *refrigerator* are not valid PTs) and (ii) it requires significant differences in the form, functionality or usage location to make a new PT comparing to existing ones (e.g., *utility sink* is qualified as it distinguishes itself from a standard *sink* in its usage whereas *cordless drill* is not as "cordless" doesn't change the core functionality of a *drill*).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '20, Aug 21–26, 2020, San Diego, CA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Obviously, neither the definition is definite nor the guidelines are exhaustive enough and there are always complicated cases and exceptions in which human judgement based on knowledge in merchandising, customer preference or just common sense is required. For example, a generic PT *range* can be broken down into more granular ones by fuel type like *gas range*, *electric range* or by other attribute like *induction range*, *convention range*. The word "wood" is material in *wood rolling pin* while is about usage in *wood glue*.

However, leveraging human knowledge in large scale problems is usually timely and expensive. To reduce such cost, this paper proposes an active learning framework that minimizes human effort in PT discovery by 1) identifying high quality candidates using phrase mining and user behavior. 2) limiting number of PT candidates for human validation.

2 RELATED WORK

Recently, incorporating structured human knowledge encapsulated in KG is proven to be very effective in various applications [13]. In this section we discuss some of the related works in the area of KG construction and completion. A technique to extract the information with no pre-defined ontology is proposed in [8]. The authors utilized semi-supervised label propagation approach to collect data and train a classifier to extract entities relations. While there are several generic knowledge bases, one of the challenges associated with domain-specific KG construction and completion is lack of publicly available knowledge base in that particular domain. To address aforementioned issue, a salable methodology is studied to expand the KG by integrating a domain-specific KG with a general domain one (such as Freebase) [18]. The authors employed graph neural network to automatically align the entities in multiple knowledge bases. In the domain of e-commerce, several unsupervised techniques have been used to generate a commercial product-brand knowledge base by leveraging customer behavior and search terms [1]. In addition, [16] provided a comprehensive comparison between generic KGs and product KGs. In this work, a self-attention based model utilized customer behavior data (queries, co-views,...) and product's content information (title, description,...) to learn product embedding and discovered the relationships between the e-commerce products. More recently, product KGs have been widely used to improve e-commerce search performance. [5] presented unsupervised and supervised approaches to identify e-commerce product types from searched queries. They demonstrated a performance comparison between diverse approaches: (1) unsupervised product type and attribute identification directly from queries in an unsupervised fashion (2) leveraged labeled data and trained convolutional neural networks to identify product type token(s) in a query (3) trained a named entity recognition model similar to the model described by [6] to detect the product types from the queries. In this work, we introduce an active learning approach to discover new *product types* by mining data from products' catalog and query logs.

3 METHOD

Figure 2 shows the active learning framework of our PT discovery process that interactively involves human knowledge and machine learning techniques. The implementation of the framework is being discussed in the rest of this section.

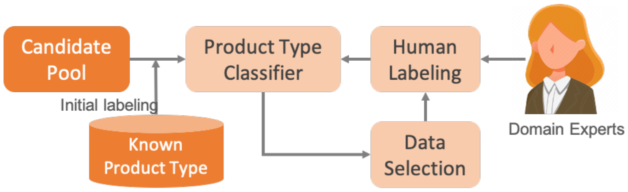


Figure 2: Product Type Discovery Framework

3.1 Candidate Pool

Instead of searching among all possible words and phrases for PTs, we prepare a list of selective candidates that are more likely to be product type. These candidates are extracted from two sources: search queries and catalog content.

3.1.1 Search Queries. As a commonly used knowledge source for PT discovery in e-commerce [5], search queries draw our attention for PT candidates as customers usually specify product types explicitly or implicitly in queries with some exceptions of model number, brand, SKU number etc. In an exploratory analysis, we observe that common search queries are very likely to be PTs, e.g.,

- all top 10 global most frequently searched queries are PTs.
- *"flashlight"*, *"flash light"*, *"uv flashlight"* etc are among the top search queries for the flashlights category.
- *"ceiling fan"* and *"ceiling fan with lights"* are the top 2 search queries that lead to the clicks of a particular ceiling fan SKU.

Based on this observation, we include the frequent search queries as PT candidates according to a volume threshold.

3.1.2 Catalog Content. Apparently, high volume search queries are biased towards popular products with poor coverage of other products which haven't met the search volume threshold. Lowering the bar can help but also introduce disproportionate noisy terms with other irrelevant attributes like brand, dimension etc (e.g., "ge" and "7.4 cu ft" in the NER examples in Figure 1b top). Comparing to arbitrarily formed search queries, catalog content like product title and description are in better format due to the format guidelines in our product onboarding process. We employ the technique and tool proposed from AutoPhrase[11] to automatically extract quality phrases from product title and description as complementary product type candidates to common search queries.

3.2 Known Product Type

Instead of building from scratch, there are thousands of known PTs previously created and validated manually. Moreover, there are two historical versions of our PT ontology: the very first and foundation version (V1) and an expanded version (V2) developed on top of V1. This enables the model evaluation that we can run the test on top of V1 (i.e., use V1 PTs for the initial labeling) and measure the outcome by comparing to V2 as the ground truth. Details of evaluation are provided in Section 4.1

3.3 Product Type Classifier

This classifier is learned from the labeled data from our domain experts and produce a confidence score of any given phrase being a valid product type. In each iteration of the active learning cycle, the classifier is trained using the positive-only distant training technique proposed in [11] with the latest labeling by domain experts. The implementation can be boiled down to two pieces:

3.3.1 Training Examples. To perform positive-only distant training, we split candidates obtained in Section 3.1 into a positive and a negative pool where training examples are drawn from. Positive pool consists of the set of valid PTs initialized by known PTs in 3.2 and updated with new ones approved in human labeling process (described in Section 3.5) in each iteration. All the rest candidates form the negative pool. The negative pool is noisy as it contains valid product types that haven't been discovered yet. Some of these undiscovered valid PTs are being moved to positive pool if validated by domain experts.

3.3.2 Feature Engineering. Given any candidate, we extract 30 features in total from the following categories:

- The outcome phrase quality score from the AutoPhrase model [11] trained on our catalog data described in Section 3.1.2.
- Intrinsic characteristics. E.g., the length, with brand name, with digits/numbers, with unit keywords like "cu ft", "mm", "volt" etc.
- Contextual characteristics w.r.t catalog data. E.g., occurrence in product titles, position in product titles etc.
- Contextual characteristics w.r.t search log. E.g., popularity as a search query in general and for a category and for individual SKUs, distribution of resulting clicks etc.

Following [11], we train a random forest model with each base classifier an unpruned decision tree learned from a "perturbed training set" [2], a subset of candidates drawn with replacement from a positive and a noisy negative pool.

3.4 Data Selection

We don't follow the typical active learning data selection strategy that selects the most informative data points for labeling to find the optimal classification boundary [3] because in our scenario the classifier is cost-sensitive, i.e., not all mis-classification errors are equal. Specifically, mistakenly approving an invalid PT could be much more damaging to our search ecosystem than missing a valid product type as many downstream applications are very sensitive the correctness of the PTs in ontology.So we have to enforce the correctness by only approving human validated PTs. In this case, the number of new valid PTs and hence the coverage of the PT ontology is bounded by the capacity of human labeling which is usually limited. To mitigate such limitation, we adopt a practical approach in data selection that presents domain experts the examples of high confidence score according to product type classifier for a higher yield of new valid PTs.

3.5 Human Labeling

As PT inherently is a concept instead of a fact, domain experts could have different opinions in validation especially for tricky cases like the *range* example mentioned in Section 1. To avoid such potential inconsistency and ensure the correctness, domain experts are advised to be conservative by only approving product types with great certainty and leaving others for a further review. This conservative strategy has an obvious impact to the classifier that the negatives are not necessarily true negatives, which echos the positive-only training technique for the product type classifier.

4 EXPERIMENTS AND RESULTS

In this section, we report two metrics: 1) lab metrics that measure effectiveness of our method and 2) business metrics that demonstrate the impact of PTs in online search context.

Table 1 shows the hyperparameters for the product type classifier training as well as the empirically selected values by grid search used in the experiment.

Hyperparameter	Tested	Selected
number of base classifiers	64, 128, 256, 512	256
max number of features to explore at each split	20%, 50%, 80%, 100%	50%
number of training examples for each base classifier	500, 1000, 2000, 5000	2000
% positive training examples	5%, 10%, 20%, 30%	10%

Table 1: Hyperparameters Grid Search

4.1 Lab Metrics

Given the limited domain expertise resources, we conducted the full cycle of experiment on one category (i.e., *Tools*) as the pilot study in which we discovered more than 200 new PTs.

In order to measure the effectiveness of our method in a broader range, we test it in a simulation by leveraging the two historical versions of the product type ontology mentioned in Section 3.2. In each iteration, classifier training and data selection are performed as described in Section 3 but with a simulated human labeling process. Specifically, with V1 PTs as the initial positive pool, PT candidates selected according to a confidence score threshold for human labeling get approved if matching any PT in V2.

4.1.1 Effectiveness. The blue curve in Fig 3 (top) shows the accumulative number of new product types being discovered as more iterations performed. More than 3500 new product types have been discovered after 50 iterations. As expected, fewer new product types can be discovered in later iterations, e.g, less than 30 in last few iterations v.s. more than 200 in the beginning. We quantified this diminishing marginal utility by precision, i.e., the ratio of correct product types among those candidates for labeling in each iteration. As the orange curve indicates, precision dropped to 13% in the last iteration from 32% in the beginning.

4.1.2 Coverage. Coverage is another critical metric to measure how complete is the resulting PT ontology. i.e., if there were X product types to cover the entire catalog, how many of them are discovered. Although the true number of product types is hard to obtain without an exhaustive labeling, we managed to estimate the

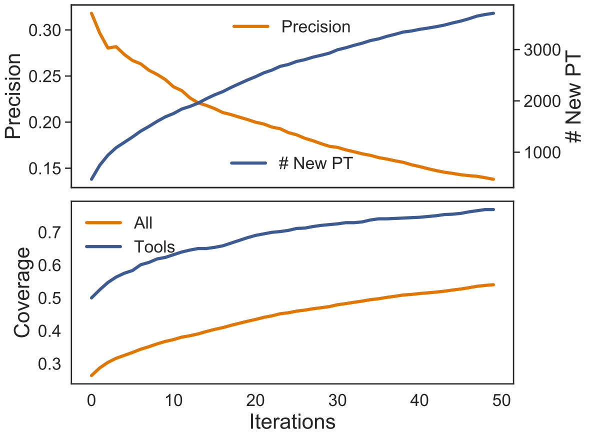


Figure 3: Simulation Results

coverage on the *Tools* category due to the following extra efforts by our domain experts including

- validate all PTs of *Tools* in V2 and remove invalid or uncertain ones from the positive pool.
- extensively examine the *Tools* category for more undiscovered PTs and add them to positive pool.

From the classifier's point of view, domain experts are essentially denoising training data for *Tools* category, i.e. removing false positives and recovering missing positives. The benefit of cleaner data is shown in Figure 3 (bottom) that there is a significant lift of coverage for *Tools* at over 70% w.r.t the denoised positive PTs vs. 50% for all categories after 50 iterations.

4.2 Business Metric

In online search scenario, a PT ontology provides a foundation to two key functionalities: 1) query understanding for PT recognition from query 2) PT-SKU association for relevant products retrieval. So we measure the impact of new PTs from the following two perspectives with results shown in Table 2:

4.2.1 PT Recognition. High PT coverage helps to recognize PT from more queries. We sample 300k queries from one category and the percentage of queries with PT recognized is compared with and without the new PTs discovered by our model.

4.2.2 Search Performance. Key search metrics including click-through rate (CTR), add-to-cart rate (ATCR) and conversion rate for a set of 150k queries sampled from another category are measured and compared for the same time period of two consecutive years (4th quarter of 2018 and 2019), one before and one after the new PTs are added.

Business Metrics	PT Recognition	Search Performance		
		CTR	ATCR	Conversion
Improvement	+800 bps	+140 bps	+40 bps	+10 bps

Table 2: Business Metrics Summary

5 CONCLUSION & FUTURE WORK

In this work, we propose an active learning framework for product type discovery that leverage domain expertise in an efficient way. The effectiveness of the framework is demonstrated by the quality and coverage of the resulting product types in the experiments as well as the positive business impact. Experiment results also show that training data denoising is significantly beneficial to method performance. There are two kinds of future work including: 1) Feature engineering of PT classifier by exploiting more textual and/or image data 2) Design a denoise procedure and add it as an additional component into the framework.

REFERENCES

[1] Omar Alonso, Vasileios Kandylas, and Rukmini Iyer. 2019. Unsupervised Construction of a Product Knowledge Graph. (2019).

[2] Leo Breiman. 2001. Randomizing Outputs To Increase Prediction Accuracy. *Machine Learning* 40 (09 2001). https://doi.org/10.1023/A:1007682208299

[3] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. 2007. Learning on the border: active learning in imbalanced data classification. (2007), 127–136.

[4] Rafael Glater, Rodrygo L.T. Santos, and Nivio Ziviani. 2017. Intent-Aware Semantic Query Annotation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 485–494. https://doi.org/10.1145/3077136.3080825

[5] Aliasgar Kutiyanawala, Prateek Verma, et al. 2018. Towards a simplified ontology for better e-commerce search. *arXiv preprint arXiv:1807.02039* (2018).

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* (2016).

[7] Taehee Lee, Ig hoon Lee, Suekyung Lee, Sang goo Lee, Dongkyu Kim, Jonghoon Chun, Hyunja Lee, and Junho Shim. 2006. Building an operational product ontology system. *Electron. Commer. Res. Appl.* (2006), 16–28.

[8] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. OpenCeres: When Open Information Extraction Meets the Semi-Structured Web. (2019), 3047–3056.

[9] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticæ Investigationes* 30, 1 (2007), 3–26. https://doi.org/10.1075/li.30.1.03nad

[10] Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. 2004. KIM – a Semantic Platform for Information Extraction and Retrieval. *Nat. Lang. Eng.* 10, 3–4 (Sept. 2004), 375–392. https://doi.org/10.1017/S135132490400347X

[11] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han. 2018. Automated Phrase Mining from Massive Text Corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[12] Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. *Official google blog* 16 (2012).

[13] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[14] Musen Wen, Deepak Kumar Vasthimal, Alan Lu, Tian Wang, and Aimin Guo. 2019. Building Large-Scale Deep Learning System for Entity Recognition in E-Commerce Search. *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (2019).

[15] Chao-Yuan Wu, Amr Ahmed, Gowtham Ramani Kumar, and Ritendra Datta. 2017. Predicting Latent Structured Intents from Shopping Queries. *Proceedings of the 26th International Conference on World Wide Web* (2017).

[16] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product Knowledge Graph Embedding for E-commerce. (2020), 672–680.

[17] Vikas Yadav and Steven Bethard. 2018. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. (Aug. 2018), 2145–2158. https://www.aclweb.org/anthology/C18-1182

[18] Qi Zhu, Hao Wei, Bunyamin Sisman, Da Zheng, Christos Faloutsos, Xin Luna Dong, and Jiawei Han. 2020. Collective Multi-type Entity Alignment Between Knowledge Graphs. (2020), 2241–2252.