QUEACO: Borrowing Treasures from Weakly-labeled Behavior Data for Query Attribute Value Extraction

Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, Qiang Yang

1Amazon.com Inc, 2Georgia Institute of Technology, 3Hong Kong University of Science and Technology

ABSTRACT

We study the problem of query attribute value extraction, which aims to identify named entities from user queries as diverse surface form attribute values and afterward transform them into formally canonical forms. Such a problem consists of two phases: named entity recognition (NER) and attribute value normalization (AVN). However, existing works only focus on the NER phase but neglect equally important AVN. To bridge this gap, this paper proposes a unified query attribute value extraction system in e-commerce search named QUEACO, which involves both two phases. Moreover, by leveraging large-scale weakly-labeled behavior data, we further improve the extraction performance with less supervision cost. Specifically, for the NER phase, QUEACO adopts a novel teacher-student network, where a teacher network that is trained on the strongly-labeled data generates pseudo-labels to refine the weakly-labeled data for training a student network. Meanwhile, the teacher network can be dynamically adapted by the feedback of the student’s performance on strongly-labeled data to maximally denoise the noisy supervisions from the weak labels. For the AVN phase, we also leverage the weakly-labeled query-to-attribute behavior data to normalize surface form attribute values from queries into canonical forms from products. Extensive experiments on a real-world large-scale e-commerce dataset demonstrate the effectiveness of QUEACO.

CCS CONCEPTS
- Information systems → Query intent. Information extraction: Online shopping.

KEYWORDS
query attribute value extraction; named entity recognition; attribute value normalization; weak-supervised learning; meta learning

ACM Reference Format:
Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, Qiang Yang. 2021. QUEACO: Borrowing Treasures from Weakly-labeled Behavior Data for Query Attribute Value Extraction. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3481946

1 INTRODUCTION

Query attribute value extraction is the joint task of detecting named entities in the search queries as the diverse surface form attribute values and normalizing them into a canonical form to avoid misspelling and abbreviation problems. These two sub-tasks are typically called named entity recognition (NER) [7] and attribute value normalization (AVN) [41].

As shown in Figure 1, we illustrate the process of the ideal query attribute value extraction. When a user enters the query "MK tote for women", we firstly use a NER model to identify the entity type "brand" for "MK", "product type" for "tote", and "audience" for "women". These extracted named entities are in the informal surface form of attribute values. However, such an informal surface is not accordant with the products indexed with canonical form attribute values in the formal written style. Specifically, "MK" is an abbreviation of brand "Michael Kors", "tote" is a hyponym of the product type "handbag", and "women" contains a spelling error. This misalignment poses tremendous challenges to the product search engine to retrieve relevant product items that users really prefer. Therefore, the AVN module is equally important to transform the surface form for each attribute value into the canonical form, i.e., "MK" to "Michael Kors", "tote" to "handbag" and "women" to "women". In the E-commerce domain, extracting these attributes values from queries is critical to a wide variety of product search applications, such as product retrieval [5] and ranking [48], and query rewriting [15].

Unfortunately, existing works only focus on the surface form attribute value extraction based on NER while ignoring the canonical
Table 1: Ground-truth labels and noisy weakly-labels for query NER examples based on the behavior data from the product side. We use colors to denote the entity type and use brackets to indicate the entity boundary. Entity labels: Brand, ProductLine, Size, ProductType, Audience.

| Case# | Query & Ground-truth Labels | Clicked Product Attribute Values | Weak Labels |
|-------|-----------------------------|---------------------------------|-------------|
| 1     | "LG 32-inch television"     | [lg] smart tv 32               |             |
| 2     | "women's socks"             | women, socks                    |             |
| 3     | "braun 7 series shaver"     | braun, series 7, electric shaver|             |
| 4     | "Trizzie cat litter tray bags 46 x 59 10 pack" | Trizzie, waste bag, 36 x 39 cm |             |

Motivated by those, we propose a unified QUEry Attribute Value Extraction in ECommerce (QUEACO) framework that efficiently utilizes the large-scale weakly-labeled behavior data for both the query NER and AVN. For query NER, QUEACO leverages the strongly-labeled data to denoise the weakly-labeled data based on a novel teacher-student network, where a teacher network trained on the strongly-labeled data generates pseudo-labels to refine the weakly-labeled data for teaching a student network. Unlike the classic teacher-student networks that can only produce pseudo-labels from a fixed teacher, our pseudo-labeling process from the teacher is continuously and dynamically adapted by the feedback of the student’s performance on the strongly-labeled data. This encourages the teacher network to generate better pseudo-labels to teach the student, maximally mitigating the error propagation from the noisy weak labels. For query AVN, we utilize the weakly-labeled query-to-attribute behavior data and QUEACO NER predictions to model the associations between the surface form and canonical form attribute values. As such, the surface form attribute values from queries can be normalized to the most relevant canonical form attribute values from the products. Empirically, extensive experiments on a real-world large-scale E-commerce dataset demonstrate that QUEACO NER can significantly outperform the state-of-the-art semi-supervised and weakly-supervised methods. Moreover, we qualitatively show the effectiveness and the necessity of QUEACO AVN.

Our contributions can be summarized as follows: (1) To the best of our knowledge, our work is the first attempt to propose a unified query attribute value extraction system in E-commerce, involving both the query NER and AVN. QUEACO can automatically identify product-related attributes from user queries and transform them into canonical forms, by leveraging weak supervisions from large-scale behavior data; (2) Our QUEACO NER is also the first work that efficiently utilizes both human-annotated strongly-labeled data and large-scale weakly-labeled data from the query-product click graph. Moreover, the proposed QUEACO NER model can significantly outperform the existing state-of-the-art baselines; (3) We propose the QUEACO AVN module that uses aggregated query to attribute behavioral data to build the connections among queries, surface form attribute value, and canonical form value. The proposed QUEACO AVN module can effectively normalize the surface form values with spelling errors, spelling variants, and abbreviations problems.

2 PRELIMINARIES

In this section, we introduce some preliminaries before detailing the proposed QUEACO framework, including the problem formulation and the query NER base model.
2.1 Problem Formulation

2.1.1 QUEACO Named Entity Recognition. We firstly introduce the task definition for the QUEACO NER.

NER Given a user input query \( X_1 = [x_1, x_2, \ldots, x_M] \) with \( M \) tokens, the goal of NER is to predict a tag sequence \( Y_1 = [y_1, y_2, \ldots, y_M] \). We use the BIO [24] tagging strategy. Specifically, the first token of an entity mention with each entity type \( C_0 \in C \) (\( C \) is the entity type set) is labeled as B—\( C_0 \); the remaining tokens inside that entity mention are labeled as I—\( C_0 \); and the non-entity tokens are labeled as O.

Strongly-Labeled and Large Weakly-Labeled Setting For our query NER, we have two types of data: 1) strongly-labeled data \( D_s = \{ (X_s^i, Y_s^i) \}_{i=1}^{N_s} \}, \) which is manually annotated by human annotators; 2) large-scale weakly-labeled data \( D_w = \{ (X_w^i, Y_w^i) \}_{i=1}^{N_w} \}, \) where \( N_s \ll N_w \). The goal is to borrow treasures from large-scale noisy weakly-labeled data to further enhance a supervised NER model trained on the strongly-labeled data.

2.1.2 QUEACO Attribute Value Normalization. For each query \( X_i = [x_1, x_2, \ldots, x_M] \) with \( M \) tokens, QUEACO NER predicts a tag sequence \( Y_i = [y_1, y_2, \ldots, y_M] \). Given an entity type \( C_0 \in C \) (e.g., brand) and the NER prediction \( Y_i \), we can extract the query term \( X_i^{c_0} \) as the surface form attribute value for the entity type \( C_0 \). Assume that we have a diverse set of canonical product attribute values \( V \) for the entity type \( C_0 \). For each canonical form attribute value \( v \in V \), we can define the relevance given the query \( X_i \) as

\[
P(C_0 = v | X_i) = \frac{\sum_{d \in D} n(d, X_i)i(d_{C_0} = v)}{\sum_{d \in D} n(d, X_i)}
\]

where \( n(d, X_i) \) is the number of total clicks on the product \( d \) of the the searches using query \( X_i \) in a period of time, such as one month. And \( D \) is the set of all products. \( i(d_{C_0} = c) \) indicates whether the product \( d \) is indexed with the value \( c \) for the entity type \( C_0 \). In a nutshell, we quantify the query-attribute relevance using the query-product relevance and the product-attribute membership. The query-product relevance is measured by the number of clicks in the query logs, which can be viewed as the implicit feedback from customers. Finally, we can get the most relevant attribute value of the entity type \( C_0 \) by arg max \( P(C_0 = v | X_i) \) as the normalized canonical form for the surface form attribute value \( X_i^{c_0} \).

2.2 Query NER Base Model

The recent emergence of the pre-trained language models (PLMs) such as BERT [11] has achieved superior performance on a variety of public NER datasets. However, existing query NER works [5, 10, 21, 48] still rely on the shallow deep learning models (e.g., BiLSTM-CRF) while not equipping with the powerful PLMs.

Why PLMs are not deployed for existing query NER works? Due to labeled data scarcity in user queries, previous query NER works can only rely on the noisy distant supervision data for model training. In such a condition, using the powerful mPLMs as the encoder has even worse performance than a shallow Bi-LSTM for the query NER [5]. Liang et al. [31] have found that the PLM-based NER models are easier to overfit the noises from the distant labels and forget the general knowledge from the pre-training stage. On the other hand, distant supervision based methods for NER [5, 45] usually underperform, which cannot meet the high performance requirement for query NER used by various downstream applications in product search like retrieval and ranking. To tackle the issue, we target a different query NER setting, which leverages some strongly-labeled human-annotated data to train a more reliable PLM-based NER model and uses the weakly-labeled data from the distant supervision to further improve the model performance. To meet the strict latency constraint, we choose DistilmBERT [43] as the base NER model and we do not add the CRF layer.

3 QUEACO

In this section, we firstly give an overview of how weakly-labeled behavior data contributes to both the query NER and AVN and then detail the two components for QUEACO, respectively.

3.1 Overview

Figure 2 shows an overview of QUEACO. At a high level, QUEACO leverages weakly-labeled behavior data for both the query NER and AVN. For QUEACO NER, we have the strongly-labeled data and the large-scale weakly-labeled data for training. Specifically, the QUEACO NER has two stages: the weak supervision pretraining stage and the finetuning stage. 1) In the pretraining stage, we adopt a novel teacher-student network where the teacher network is dynamically adapted based on the feedback from the student network. The goal is to encourage the teacher network to generate better pseudo labels to refine the weakly-labeled data for improving the student network’s performance. 2) After the pretraining stage, we continue to finetune the student network on the strongly-labeled data as the final model. For QUEACO AVN, we extract the surface form attribute values based on the NER predictions and leverage the weakly-labeled query-to-attribute behavior data to transform them into the canonical forms.

3.2 QUEACO Named Entity Recognition

3.2.1 Model architecture. Teacher-Student Network Before introducing the QUEACO NER model, we give some preliminary of the teacher-student network of self-training [23, 56]. Self-training stands out among semi-supervised learning approaches, in which a teacher model produces pseudo-labels for unlabeled samples, and a student model learns from these samples with generated pseudo-labels. We give the mathematical formulation of self-training in the context of NER. Let \( T \) and \( S \) respectively be the teacher and student network, parameterized by \( \theta_T \) and \( \theta_S \). We use \( f(X; \theta_T) \) and \( f(X; \theta_S) \) denote the NER predictions of the query \( X \) for the teacher and student, respectively: \( f(X; \theta_T) \) can be either soft or converted to hard pseudo labels. Then the knowledge transfer is usually achieved by minimizing the consistency loss between the two predicted distributions from the teacher and the student: \( L(f(X; \theta_T), f(X; \theta_S)) \).

Pseudo & Weak Label Refinement Weakly-labeled data suffers from severe incompleteness that the overall span recall is usually very low. Therefore, it is natural to use self-training to annotate the missing labels of the weakly-labeled data. The pseudo labels make up the missing tags for the weak labels, and meanwhile weak labels can provide high precision tags to restrict pseudo labels.
For each weakly-labeled sample \(X_1^w = [x_1^w, x_2^w, ..., x_M^w]\), we convert the soft predictions of the teacher network into the hard pseudo labels, i.e., \(Y_T^p = \arg \max f(X_1^w; \theta_T) = [y_1^p, y_2^p, ..., y_M^p]\). Additionally, we have weak labels \(Y_i^w = [y_i^w, y_2^w, ..., y_M^w]\) that partially annotate the samples, which can be used to further refine the pseudo labels. We maintain the weak labels of the entity tokens and replace the weak labels of the no entity tokens with the pseudo labels. Then the refined pseudo labels \(Y_i' = [y_i^p, y_2^w, ..., y_M^w]\) are generated by:

\[
y_i' = \begin{cases} 
y_i^p, & \text{if } y_i^w = \text{O} 
y_i^w, & \text{otherwise}
\end{cases}
\]

**QUEACO Teacher-Student Network** Prior teacher-student frameworks of self-training rely on rigid teaching strategies, which may hardly produce high-quality pseudo-labels for consecutive and interdependent tokens. This results in progressive drifts on the noisy pseudo-labeled data provided by the teacher (a.k.a. the confirmation bias [2]). In QUEACO NER, we propose a novel teacher-student network, where the teacher can be dynamically adapted from the student’s feedback to adjust its pseudo-labeling strategies, inspired by Pham et al. [39]. Student’s feedback is defined as the student’s performance on the strongly-labeled data. Formally, we can formulate our teacher-student network as a bi-level optimization problem:

\[
\min_{\theta_T} L_{S1}(\theta_S^{t+1}(\theta_T))
\]

s.t. \(\theta_S^{t+1}(\theta_T) = \arg \min_{\theta_S} \frac{1}{N_w} \sum_{i=1}^{N_w} \ell(Y_i', f(X_i^w; \theta_S'))\).

where \(\ell\) is the cross-entropy loss. The ultimate goal is to minimize the loss of the student \(\theta_S^{t+1}\) on the strongly-labeled data after learning from the refined pseudo labels \(Y_i'\), i.e., \(L_{S1}(\theta_S^{t+1}(\theta_T))\), which is a function of the teacher’s parameters \(\theta_T\). \(f(X_i^w; \theta_S')\) is the prediction logits of the student network on the weakly-labeled sample \(X_i^w\). By optimizing the teacher’s parameter in light of the student’s performance on the strongly-labeled data, the teacher can be adapted to generate better pseudo labels to further improve student’s performance. This bi-level optimization problem is extremely complicated, but we can approximate the multi-step arg min with one step gradient update of \(\theta_S\). Plugging this into the constrained optimization problem leads to an unconstrained optimization for the teacher network learning. This gives rise to the alternating optimization procedure between the student and the teacher updates.

**3.2.2 Model Training**

**Student Network** The student network is trained with refined pseudo-labeled data \(Y_i'\) in order to move closer to the teacher,

\[
L_S = \frac{1}{N_w} \sum_{i=1}^{N_w} \ell(Y_i', f(X_i^w; \theta_S))
\]

We update student network parameter \(\theta_S\) with one step of gradient descent. In our proposed framework, the feedback signal from the student network to the teacher network is the student’s performance on the strongly-labeled data. We use the student loss on the strongly-labeled data to measure the performance before the update (\(\theta_S^{t}\)) and after the update (\(\theta_S^{t+1}\)), learning on the refined pseudo-labeled data,

\[
L_S^{(t)} = \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(Y_i', f(X_i^s; \theta_S)), \quad L_S^{(t+1)} = \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(Y_i', f(X_i^s; \theta_S^{t+1})).
\]

The difference between \(L_S^{(t)}\) and \(L_S^{(t+1)}\), i.e., \(\lambda_{meta} = L_S^{(t+1)} - L_S^{(t)}\), can be used as the feedback to meta-optimize the teacher network towards the direction that generates better pseudo labels. If the current generated pseudo labels can further boost the student network, then \(\lambda_{meta}\) will be negative, and positive vice versa.

**Teacher Network** The teacher network is jointly optimized by two objectives: a typical semi-supervised learning loss \(L_{ssl}\) and a meta learning loss \(L_{meta}\):

\[
L_T = L_{ssl} + L_{meta}.
\]

For the SSL loss, it consists of the supervised loss on the strongly-labeled data and the regularization loss on the weakly-labeled data.

\[
L_{ssl} = L_{sup} + L_{reg}.
\]
The supervised loss $L_{\text{sup}}$ is defined as:

$$L_{\text{sup}} = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{N_w} \sum_{t=1}^{N_w} \sum_{j=1}^{M} \frac{f(X_i^{w_t}; \theta_T)}{\tau} \log(f(\hat{x}_{ij}^{w_t}; \theta_T)) \right)$$

The regularization loss $L_{\text{reg}}$ alleviates the overfitting of the teacher by enforcing the prediction consistency between the original and augmented weakly-labeled samples.

$$L_{\text{reg}} = -\frac{1}{N_w \times M} \sum_{t=1}^{N_w} \sum_{j=1}^{M} f(x_{ij}^{w_t}; \theta_T) \log(f(\hat{x}_{ij}^{w_t}; \theta_T))$$

where $f(x_{ij}^{w_t}; \theta_T)$ is the prediction logits of the teacher network on the $j$-th token of the $i$-th weakly-labeled sample $X_i^{w_t}$. $\log(f(\hat{x}_{ij}^{w_t}; \theta_T))$ is the prediction logits of the corresponding token of the augmented weakly-labeled sample $X_i^{w_t}$ and $\tau$ is the temperature factor to control the smoothness. Here, we do not explicitly augment the sentence and instead add random Gaussian noises $G(0, \sigma^2)$ to the BERT embedding of each token to increase the diversity of the sentence.

The meta loss $L_{\text{meta}}$ for the teacher network is defined as:

$$L_{\text{meta}} = \frac{\lambda_{\text{meta}}}{N_w} \sum_{i=1}^{N} \sum_{t=1}^{N_w} \sum_{j=1}^{M} (\log(f(\hat{x}_{ij}^{w_t}; \theta_T)))$$

The performance variation of the student network on the strongly-labeled data is formulated as the feedback signal $\lambda_{\text{meta}}$ to dynamically adapt the teacher network’s pseudo-labeling strategies. The teacher and student can have the same encoder (e.g., DistilBERT [43]), or a larger teacher for better prediction (e.g., BERT [11]) and a small student (e.g., DistilBERT) for fast online production inference.

### 3.3 QUEACO Attribute Value Normalization

In this section, we discuss two different types of AVN method for and the product type attribute and general attributes, respectively.

#### 3.3.1 AVN for Product type attribute

E-commerce websites usually have their own self-defined product category taxonomy, which is used for organizing and indexing the products. Thus, identifying the product type of a given query is one of the most critical components of the query attribute value extraction.

However, there are three challenges in directly normalizing the surface form product type: 1) some queries do not have explicit surface form product type while they are implicitly associated with some product types. For example, as shown in Table 2 case#2, there is no surface form product type in a movie query “wonder woman 1984", but the product type of the query is “movie”; 2) many entity mentions are the hyponyms of product type values. For example, as shown in Table 2 case#6, for the query “mini pocket detangler brush”, its surface form product type “detangler brush” is a hyponym of its product type “hair brush”; 3) the same surface form might correspond to different product types. For example, the product type of the query “tote for travel” is “luggage", but the product type of the query “mk tote for woman” is “handbag”.

Alternatively, we can get the query-to-productType associations using the weakly-labeled behavior data. For frequent queries, we use query search logs to get the product type relevance vector $Y_i^{\text{pt}} = P(C_i = Y|X_i)$ of query $X_i^{w_t}$ as defined in Section 2.1.2, and then get the most relevant product types. Given that not all queries have enough user-behavioral signals, we use this weakly labeled data $D = \{(X_i^{w_t}, Y_i^{\text{pt}})\}_{i=1}^{N_w}$ to train a multi-label query classification model [16, 20, 32] for predicting the product type distribution of less frequent queries. To meet the latency constraint, we also use DistilBERT as the encoder.

| Case# | Query       | Surface form | Behavior-based |
|-------|-------------|--------------|----------------|
| 1     | nike        | None         | shoes          |
| 2     | wonder woman 1984 | None | movie         |
| 3     | unicorn     | None         | clothes, toys  |
| 4     | lg smart tv 32 | smart tv    | television     |
| 5     | patio umbrella | patio umbrella | umbrella |
| 6     | mini pocket detangler brush | detangler brush | hair brush |
| 7     | tote for travel | tote          | luggage        |
| 8     | mk tote for women | tote         | handbag        |

### 3.3.2 AVN for general attributes

The attribute value normalization corresponds to the entity disambiguation task in entity linking. Prior entity linking works for search queries [3, 9, 46] leverage additional information, such as knowledge base and query log, and search results. Inspired by this, we propose to extract common surface form to canonical form mapping based on QUEACO NER predictions and weakly-labeled query-to-attribute associations.

We use the entity type "brand" $b$ as the example. Using the method defined in Section 2.1.2, we can get the most relevant brand $b_i$ for the query $X_i^{w_t}$ by aggregating the query search logs. Then we can associate surface form brand $X_i^{w_t}$ and the most relevant behavior-based brand $b_i$ through the query $X_i^{w_t}$. Given a surface form brand value $m$ and a canonical form brand value $v$, we can define the mapping probability between them as,

$$P(v|m) = \frac{\sum_{i=1}^{N_w} \mathbb{1}(X_i^{w_t} = m, b_i = v)}{\sum_{i=1}^{N_w} \mathbb{1}(X_i^{w_t} = m)}.$$  

However, we find the same surface form can be normalized to different canonical forms depending on the query context. For example, as shown in Table 2 case#1 and #2, the same surface form size “apple” can be mapped to “apple barrel” given the query “apple craft paint”, and “Apple computer” given the query “apple macbook pro”. The finding is consistent with the recent embedding-based entity linking works [1, 49]. However, due to the strict requirement on the inference latency and very high request volume, it is hard to directly apply the current state-of-the-art embedding-based entity disambiguation models, which use the context embedding for the candidate ranking, to the query side [1, 14, 49, 53, 54]. Alternatively, we simplify the setting by using query product type as the context of the query. We then define the probability of one surface form value $m$ conditioned on canonical form attribute value $v$, given product type $p$ as:

$$P(v|m, p) = \frac{\sum_{i=1}^{N_w} \mathbb{1}(X_i^{w_t} = m, b_i = v, Y_i^{\text{pt}} = p)}{\sum_{i=1}^{N_w} \mathbb{1}(X_i^{w_t} = m, Y_i^{\text{pt}} = p)}.$$
4 EXPERIMENTS

4.1 QUEACO query NER

4.1.1 Data Description. We collect search queries from a real-world e-commerce website and construct two datasets: (1) strongly-labeled dataset, which is human annotated, and (2) weakly-labeled dataset, which is generated through the partial query tagging, as shown in Table 1. The statistics of the strongly-labeled and the weakly-labeled datasets are shown in table 4 and table 5. The details of these datasets are shown below:

- We split the entire dataset into train/dev/test by 90%, 5%, and 5%. The size of strongly-labeled and the weakly-labeled training data are 677K and 17M. The weakly-labeled dataset is more noisy and is more than 26 times bigger than the strongly-labeled dataset.
- The strongly-labeled data contains 12 languages: English (En), German (De), Spanish (Es), French (Fr), Italian (It), Japanese (Jp), Chinese (Zh), Czech (Cs), Dutch (Nl), Polish (Pl), Portuguese (Pt), Turkish (Tr). The weakly-labeled dataset does not have Zh, Cs, Nl, and Pl languages.
- The non-O %coverage for the strongly-labeled dataset is 98.31%, and there are 13 non-O types. However, the non-O %coverage for weakly-labeled data is 43.21%, and there are 11 non-O types, indicating the weak labels suffer from severe incompleteness issues. The incomplete annotation is due to the exact string matching between query span and product attribute values [36]. Table 5 also presents the precision and recall of weak label performance on an evaluation golden set. In particular, the overall recall is lower than 50, which is consistent with the non-O %coverage. The low recall issue is even more severe for low-resource languages, like Jp, Pt, and Tr. At the same time, the weak labels also suffer from labeling bias since the overall precision is lower than 80%.

| Dataset | #Train | #Dev | #Test | Non-O Type | Non-O %Coverage |
|---------|--------|------|-------|------------|-----------------|
| En      | 256571 | 14193| 14269 | 13         | 98.87           |
| De      | 98980  | 5442 | 5473  | 13         | 95.49           |
| Es      | 63844  | 3600 | 3488  | 13         | 99.05           |
| Fr      | 79176  | 4383 | 4504  | 13         | 98.91           |
| It      | 52136  | 2933 | 2867  | 13         | 99.04           |
| Jp      | 77457  | 4422 | 4365  | 13         | 98.65           |
| Zh      | 22467  | 1238 | 1247  | 13         | 98.51           |
| Cs      | 4430   | 272  | 252   | 13         | 93.66           |
| Nl      | 8562   | 423  | 478   | 13         | 97.09           |
| Pl      | 4489   | 251  | 229   | 13         | 92.19           |
| Pt      | 4467   | 273  | 247   | 13         | 99.45           |
| Tr      | 5093   | 267  | 274   | 13         | 99.52           |
| Total   | 677672 | 37697| 37693 | 13         | 98.31           |

Table 4: The data statistics of strongly-labeled NER dataset.

4.1.2 Evaluation Metrics. We use the span-level micro precision, recall and F1-score as the evaluation metrics for all experiments. For the per language experiment, we only report the span-level micro F1-score for each language, due to the space limit.

4.1.3 Analysis of the Base Encoder. We benchmark the DistilBERT performance with the baseline models in the query attribute extraction literature. All RNN experiments use FastText multilingual word embeddings [8] and the TARGER implementation [6].

- RNN models: BiLSTM, BiGRU, BiLSTM-CRF and BiGRU-CRF models are benchmarked for the Home Depot query NER model [5].
- BiLSTM-CNN-CRF [22, 34] is the state-of-the-art NER model architecture before BERT [11, 55].
- DistilBERT baselines: 1) DistilBERT (Single) means separately finetuning DistilBERT on the strongly-labeled data for each single language. 2) DistilBERT (Multi) means fine-tuning DistilBERT on the strongly-labeled data for all languages.

| Method (Span level) | Precision | Recall | F1  |
|---------------------|-----------|--------|-----|
| BiLSTM              | 65.66     | 70.09  | 67.81|
| BiGRU               | 64.35     | 68.96  | 66.58|
| BiLSTM-CRF          | 71.04     | 69.36  | 70.19|
| BiGRU-CRF           | 69.45     | 67.98  | 68.71|
| BiLSTM-CNN-CRF      | 70.33     | 67.92  | 69.11|
| BiGRU-CNN-CRF       | 67.75     | 65.40  | 66.56|
| DistilBERT (Single) | 71.72     | 74.16  | 72.92|
| DistilBERT (Multi)  | 73.33     | 75.29  | 74.29|

Table 6: Comparison of different encoders.

As shown in Table 6: the DistilBERT has better performance than other non-BERT baselines. Furthermore, finetuning DistilBERT with all languages has better performance than training a separate model for each language.

4.1.4 Discussion on the training data. In this section, we discuss the use of training data for QUEACO query NER model. We benchmark our setting with the baseline in the query attribute extraction literature, where only weakly-labeled data is available. All experiments use DistilBERT as the base NER model for a fair comparison.

In Figure 3, we subsample the strongly and weakly-labeled dataset and we find:
All implementations are based on transformers. The temperature factor for smoothness $\tau$ with 8 NVIDIA A100-SXM4-40GB GPUs, configured with CUDA weak labels. And this explains why the existing query NER works do not adopt the state-of-the-art pre-trained language stages. For model training, we use an Amazon EC2 virtual machine on the validation set during both the pretraining and finetuning in Pytorch 1.7.0. To alleviate overfitting, we perform early stopping amongst $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. The threshold is tuned amongst $[8, 16, 32, 64]$. The Gaussian noise variance $\sigma$ is tuned amongst $[0.01, 0.1, 1.0]$. The temperature factor for smoothness $r$ is tuned amongst $[0.5, 0.6, 0.7, 0.8, 0.9]$. The threshold $\epsilon$ is tuned amongst $[0.5, 0.6, 0.7, 0.8, 0.9]$. All implementations are based on transformers in Pytorch 1.7.0. To alleviate overfitting, we perform early stopping on the validation set during both the pretraining and finetuning stages. For model training, we use an Amazon EC2 virtual machine with 8 NVIDIA A100-SXM4-40GB GPUs, configured with CUDA 11.0.

The precision and recall of the model trained with the weakly-labeled data do not change much when the training data size increases from 10% to 100%. However, both the precision and recall increase dramatically when size of strongly-labeled data increases, especially the precision.

The best precision that weakly-labeled data can achieve is around 60%. However, 34K strongly-labeled queries can already achieve 62.27% precision. And the precision reaches to 72.90% when trained with 677K strongly-labeled queries. With only weakly-labeled data, the best recall is only around 26%, much lower than that using strongly-labeled data. 7K strongly-labeled data can already achieve 48.78% recall.

The findings are consistent with the conclusion of BOND [31] that pre-trained language models can easily overfit to incomplete weak labels. And this explains why the existing query NER works do not adopt the state-of-the-art pre-trained language model.

In Figure 4, we show the performance improvement for introducing weakly-labeled data to different sizes of randomly subsampled strongly-labeled data. It is shown that the smaller strongly-labeled data, the bigger improvement the weak labels can introduce. However, the performance improvement is marginal when the strongly-labeled dataset is sufficient. In section 3.2, we introduce the QUEACO query NER model to better utilize the weak labels to further improve the query NER model performance.

### 4.1.5 Implementation Details of QUEACO

We employ the DistilmBERT [43] with 6 layers, 768 dimension, 12 heads and 134M parameters as our encoder. We use ADAM optimizer with a learning rate of $10^{-5}$, tuned amongst $[10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}]$. We search the number of epochs in $[1, 2, 3, 4, 5]$ and batch size in $[8, 16, 32, 64]$. The Gaussian noise variance $\sigma$ is tuned amongst $[0.01, 0.1, 1.0]$. The temperature factor for smoothness $r$ is tuned amongst $[0.5, 0.6, 0.7, 0.8, 0.9]$. The threshold $\epsilon$ is tuned amongst $[0.5, 0.6, 0.7, 0.8, 0.9]$. All implementations are based on transformers in Pytorch 1.7.0. To alleviate overfitting, we perform early stopping on the validation set during both the pretraining and finetuning stages. For model training, we use an Amazon EC2 virtual machine with 8 NVIDIA A100-SXM4-40GB GPUs, configured with CUDA 11.0.

Figure 3: Size of strongly & weakly labeled data vs. Performance. All results are produced by directly finetuning the DistilmBERT model with the subsampled dataset. We subsample 1%, 2%, 5%, 10% and 100% of the 677K strongly-labeled data, and subsampled 0.1%, 1%, 10% and 100% of the 17M weakly-labeled data. In (a), (b) and (c), span-level precision, recall and micro-f1 are shown.

- The precision and recall of the model trained with the weakly-labeled data do not change much when the training data size increases from 10% to 100%. However, both the precision and recall increase dramatically when size of strongly-labeled data increases, especially the precision.
- The best precision that weakly-labeled data can achieve is around 60%. However, 34K strongly-labeled queries can already achieve 62.27% precision. And the precision reaches to 72.90% when trained with 677K strongly-labeled queries. With only weakly-labeled data, the best recall is only around 26%, much lower than that using strongly-labeled data. 7K strongly-labeled data can already achieve 48.78% recall.

The precision and recall of the model trained with the weakly-labeled data do not change much when the training data size increases from 10% to 100%. However, both the precision and recall increase dramatically when size of strongly-labeled data increases, especially the precision.

The best precision that weakly-labeled data can achieve is around 60%. However, 34K strongly-labeled queries can already achieve 62.27% precision. And the precision reaches to 72.90% when trained with 677K strongly-labeled queries. With only weakly-labeled data, the best recall is only around 26%, much lower than that using strongly-labeled data. 7K strongly-labeled data can already achieve 48.78% recall.

Figure 4: Size of Strongly Labeled Data vs. Micro span-level F1. "strongly labeled": a baseline that finetunes DistilmBERT with the strongly labeled data, "strongly & weakly labeled": a baseline that pretrained DistilmBERT with weakly labeled and then finetunets it on the strongly labeled data.

### 4.1.6 Baseline Models

As discussed in section 2.2 and section 4.1.4, it is evident that the setting of using DistilmBERT as base NER model and using both strongly and weakly-labeled dataset as training data, outperforms the other settings. We also conduct baseline experiments in similar settings to show the effectiveness of the QUEACO query NER model. All experiments use DistilmBERT as the base NER model for the fair comparison.

- **Supervised Learning Baseline**: We directly fine-tune the pre-trained model on the strongly-labeled data.
- **Semi-supervised Baseline**
  - Self Training: self-training with hard pseudo-labels
  - NoisyStudent [51] extends the idea of self-training and distillation with the use of noise added to the student during learning.
- **Weakly-supervised Baseline**: Similar to QUEACO, these weakly-supervised baselines also have two stages: pretraining with strongly-labeled and weakly-labeled data, and finetuning with strongly-labeled data. We only report stage 2 performance.
  - Weakly Supervised Learning (WSL): Simply combining strongly-labeled data with weakly-labeled data [35].
  - Weighted Weakly Supervised Learning (Weighted WSL): WSL with weighted loss, where weakly-labeled samples have
Table 7: Comparison between QUEACO and baseline methods on micro span-level F1.

| Method (Span level) | Precision | Recall | F1   |
|---------------------|-----------|--------|------|
| Supervised Baseline |           |        |      |
| DistilmBERT (Single) | 71.72     | 74.16  | 72.92|
| DistilmBERT (Multi)  | 73.33     | 75.29  | 74.29|
| Semi-supervised Baseline (Encoder: DistilmBERT) | | | |
| ST                  | 73.29     | 75.44  | 74.35|
| Noisy student       | 73.28     | 75.38  | 74.32|
| Weakly-supervised Baseline (Encoder: DistilmBERT) | | | |
| unweighted WSL      | 73.81     | 75.93  | 74.85|
| weighted WSL        | 73.77     | 75.97  | 74.85|
| robust WSL          | 73.10     | 75.20  | 74.14|
| BOND hard           | 73.77     | 75.81  | 74.78|
| BOND soft           | 73.65     | 75.68  | 74.65|
| BOND soft high conf | 73.95     | 76.05  | 74.98|
| BOND noisy student  | 73.97     | 75.99  | 74.97|

| Ours (Student: DistilmBERT) | 74.44 | 76.35 | 75.38 |
| QUEACO (Teacher: DistilmBERT) | 74.48 | 76.41 | 75.44 |
| Δ                  | (+0.51) | (+0.36) | (+0.46) |

Table 8: Comparison between DistilmBERT (Multi) and QUEACO for each language on micro span-level F1.

| Language | Weakly Data available | DistilmBERT (Multi) | QUEACO |
|----------|-----------------------|---------------------|--------|
| En       | True                  | 75.42               | 76.97  (+1.55) |
| De       | True                  | 75.26               | 76.70  (+1.44) |
| Es       | True                  | 77.30               | 77.67  (+0.37) |
| Fr       | True                  | 71.56               | 73.20  (+1.64) |
| It       | True                  | 77.88               | 78.42  (+0.54) |
| Jp       | True                  | 65.49               | 65.88  (+0.39) |
| Zh       | False                 | 71.02               | 72.19  (+1.17) |
| Cs       | False                 | 72.61               | 70.93  (-1.68) |
| Ni       | False                 | 75.46               | 75.30  (-0.16) |
| Pt       | False                 | 79.71               | 79.43  (-0.28) |
| Tr       | True                  | 58.24               | 62.00  (+3.76) |

Table 9: Ablation study.

| Method (Span level) | Precision | Recall | F1   |
|---------------------|-----------|--------|------|
| QUEACO w/o student feedback | 74.09 | 76.11 | 75.09 |
| QUEACO w/o noise | 74.18 | 76.01 | 75.08 |
| QUEACO w/o weak labels | 74.04 | 75.77 | 74.89 |
| QUEACO w/o finetune | 63.31 | 66.62 | 64.92 |
| QUEACO          | 74.44 | 76.35 | 75.38 |

4.1.8 Ablation Study.

- QUEACO w/o student feedback $L_{meta}$: use a fixed teacher network to generate pseudo labels for a student network.
- QUEACO w/o noise: remove random Gaussian noise added to the BERT embedding when training the teacher network.
- QUEACO w/o weak labels: remove the pseudo & weak label refinement step, and only use the pseudo labels for student network training.
- QUEACO w/o finetune: remove stage 2: strong labels fine-tuning.

As shown in table 9, we find the final finetuning is essential to QUEACO NER. All components from QUEACO, including student feedback, random Gaussian noise to the BERT embedding and the pseudo & weak label refinement, are effective.

4.2 QUEACO Attribute Value Normalization

4.2.1 Product type AVN. In the query NER, the span-level micro F1-score for product type is only 77.12%. The performance for NER-based product type value extraction will be even worse, since many surface forms cannot be normalized. In Table 10, we show the product type precision, recall and F1 of the multi-label query NER. In Table 10, we can see that our proposed method is effective in finding common surface attributes, including:

- spelling error: brand "Michael Kors" is often misspelled as "Michael Kors", "Levi\'s" is often misspelled as "levi";
- spelling invariants: for example, "3 by 5" and "3\times5" are different variants with the same meaning.
Table 10: Product type attribute value extraction performance.

| Country | Eval Data Size | Precision | Recall | F1    |
|---------|----------------|-----------|--------|-------|
| USA     | 2746           | 85.13     | 81.1   | 83.08 |
| UK      | 2590           | 85.44     | 85.71  | 85.58 |
| Canada  | 2705           | 85.07     | 86.41  | 85.73 |
| Japan   | 2151           | 85.2      | 80.06  | 82.55 |
| Germany | 2254           | 85.01     | 88.54  | 86.74 |

We conducted an end-to-end evaluation of QUEACO on real-world whose goal is to rerank the top-16 products based on their relevance to the query intent. Specifically, we first use QUEACO to extract attributes for the product search queries. Then, we generate boolean features, such as is pt match, is brand match, based on the attribute values of queries and products. We refer to these boolean features as QUEACO features. We then train two learning-to-rank (LTR) models: one model uses QUEACO features while the other does not. All other features, settings and hyperparameters of these two models are the same. To compare these two models, we use NDCG@16, which is the normalized discounted cumulative gain (NDCG) score for the top 16 products of the search result. We conducted online A/B experiments for this reranking application in four countries: India, Canada, Japan, and Germany. On average, we improve the NDCG@16 by 0.36%.

6 RELATED WORK

6.1 E-commerce Attribute Value Extraction

Most of the previous works on e-commerce attribute value extraction focus on extracting surface-form attribute values from product titles and descriptions. Some early machine learning works formulate the task as a (semi-)classification problem [12, 40]. Later, several researchers [37, 41] employ a sequence tagging formulation and adopt the CRF model architecture. With the recent advances in deep learning, many RNN-CRF based models are applied to the sequence tagging task [18, 22, 34], and have achieved promising results. Following this trend, recent works on the product attribute value extraction task [36, 52, 58] also adopt variants of the BiLSTM-CRF model architecture. In addition, some recent studies have explored BERT-based [11] Machine Reading Comprehension (MRC) [52] and Question & Answering (Q&A) [47] formulation.

Query attribute value extraction works [5, 10, 21, 48] also employ the sequence tagging formulation and adopt BiLSTM-CRF model architectures as well as its variants. Recent works [5, 48] utilize large behavioral-based data to generate partial query tagging as distant supervision to train the NER model, and they also explore data augmentation and active learning to deal with the data quality issues.

6.2 NER with Distant Supervision

To alleviate human labeling efforts, various approaches such as transfer learning [38], semi-supervised learning [4], and weakly-supervised learning [59] are emerging and widely applied to low-resource NLP tasks [26, 33, 57], e.g., sentiment classification [27–30], information extraction [17, 25, 44], etc. Specifically, distant supervision is a type of weak supervision, and is automatically generated based on some heuristics, such as matching spans of unlabeled text to a domain dictionary [31, 45]. Existing works on NER with distant supervision [31, 45] mainly focus on the setting that can only access distant supervision. Besides, most existing query NER works [5, 48] only rely on the distant supervision, generated from partial query tagging, for NER model training.

However, in some cases both strongly-labeled data and a large amount of distant supervision are available. The strongly-labeled data, though expensive to collect, is validated to be critical to boost distant supervised NER performance [19].

7 CONCLUSION

This paper proposes to utilize the weakly-labeled behavioral data to improve the named entity recognition and attribute value normalization phases of query attribute value extraction. We conduct extensive experiments on a real-world large-scale E-commerce dataset and demonstrate that the QUEACO NER can achieve the state-of-the-art performance and the QUEACO AVN effectively normalizes some common customer typed surface forms. We also validate the effectiveness of the proposed QUEACO system for the downstream product reranking application.
[50] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In ICML. 478–487.

[51] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In CVPR.

[52] Huimin Xu, Wenting Wang, Xinnian Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 5214–5223.

[53] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. 250–259.

[54] Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Global entity disambiguation with pretrained contextualized embeddings of words and entities. arXiv preprint arXiv:1909.00426 (2019).

[55] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. [n.d.]. XLNet: Generalized Autoregressive Pretraining for Language Understanding. ([n. d.]).

[56] David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In 33rd annual meeting of the association for computational linguistics. 189–196.

[57] Danqing Zhang, Tao Li, Haiyang Zhang, and Bing Yin. 2020. On Data Augmentation for Extreme Multi-label Classification. arXiv preprint arXiv:2009.10778 (2020).

[58] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. Opentag: Open attribute value extraction from product profiles. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1049–1058.

[59] Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. National science review 5. 1 (2018), 44–53.