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

Danqing Zhang*[1], Zheng Li*[1], Tianyu Cao[1], Chen Luo[1], Tony Wu[1], Hanqing Lu[1],
Yiwei Song[1], Bing Yin[1], Tuo Zhao[2], Qiang Yang[3]
[1]Amazon.com Inc, CA, USA
[2]Georgia Institute of Technology, GA, USA
[3]Hong Kong University of Science and Technology, HK, China
[1]{danqinz,amzzhe,caoty,cheluo,tonywu,luhanqin,ywsong,alexbyin}@amazon.com
[2]tourzhao@gatech.edu, [3]qyang@cse.ust.hk

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*[1], Zheng Li*[1], Tianyu Cao[1], Chen Luo[1], Tony Wu[1], Hanqing Lu[1],, Yiwei Song[1], Bing Yin[1], Tuo Zhao[2], Qiang Yang[3]. 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, 10 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) [6] and attribute value normalization (AVN) [31].

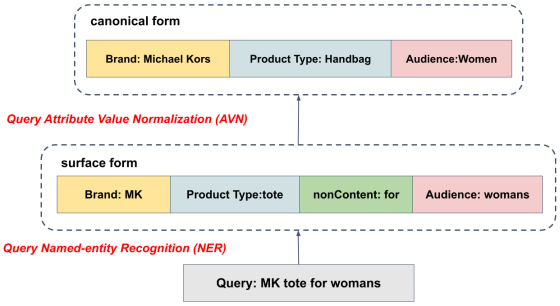


Figure 1: The ideal product attribute extraction pipeline.

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 womans*", we firstly use a NER model to identify the entity type "brand" for "*MK*", "product type" for "*tote*", and "audience" for "*womans*". 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 "*womans*" 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 "*womans*" to "*women*". In the E-commerce domain, extracting these attributes

*Authors contributed equally to this paper.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481946

Case#	Query & Ground-truth Labels	Clicked Product Attribute Values	Weak Labels
1	[*lg*][*smart tv*][*32*]	*lg*, *32-inch*, *television*	[*lg*] *smart tv 32*
2	[*womans*][*socks*]	*women*, *socks*	*womans* [*socks*]
3	[*braun*][*7 series*][*shaver*]	*braun*, *series 7*, *electric shaver*	[*braun*] *7 series shaver*
4	[*trixie*] [*cat litter tray bags*][*46 x 59*][*10 pack*]	*Trixie*, *waste bag*, *46 × 59 cm*	[*trixie*] *cat litter tray bags 46 x 59 10 pack*

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.

values from queries is critical to a wide variety of product search applications, such as product retrieval [4] and ranking [38], and query rewriting [14].

Unfortunately, existing works only focus on the surface form attribute value extraction based on NER while ignoring the canonical form transformation, which is impractical in the realistic scenarios [4, 9, 19, 38]. To bridge this gap, this paper proposes a unified query attribute value extraction system that involves both phases. By borrowing treasures from large-scale weakly-labeled behavior data to mitigate the supervision cost, we further improve the extraction performance. Considering the first NER stage, recent advances in deep learning models (e.g., Bi-LSTM+CRF) have achieved promising results [17, 32]. However, they highly rely on massive labeled data, where manual labeling for token-level labels is particularly costly and labor-intensive. To alleviate the issue in E-commerce, prior studies [4, 19, 38] resort to leveraging large-scale behavior data from the product side as the weak supervision for queries based on some simple string match strategies. Nonetheless, these weakly-supervised labels contain enormous noises due to the partial or incomplete token labels based on the exact string matching. For example, as shown in Table 1 case#1, when we use the attribute values of top-clicked product "LG 32-inch television", i.e., "brand" for "*LG*", "size" for "*32-inch*", "product type" for "*television*"as the weak supervision to match the query "*lg smart tv 32*", it can only generate the label "brand" for "*lg*", concealing useful knowledge for the unannotated tokens. For this reason, weak supervision-based methods [4, 35] usually perform very poorly, even worse after powerful pre-trained language models (PLMs) (e.g., BERT [10]) are introduced since PLMs are much easier to fit noises. To address the issue, we consider a more reliable regime, which further includes some strongly-labeled human annotated data to denoise the weak labels from the distant supervision. As such, the NER model can be improved by making more effective use of both the large-scale weakly-labeled behavior data and the strongly-labeled human-annotated data.

As for the second AVN phase, customers tend to use diverse surface forms to mention each attribute value in search queries due to the misspellings, spelling variants, or abbreviations. This circumstance occurs frequently in user queries and product titles of e-commerce. For example, eBay has noted that 20% product titles in the clothing and shoes category involve such surface form brand [31]. Thus, normalizing these surface form attribute values derived from the NER signals to a single normalized attribute value is critical. It is usually ignored by existing works [4, 9, 19, 38]. To mitigate human annotating efforts, weakly-labeled behavior data can also contribute to the AVN. For example, "*MK tote for womans*" mentioning the brand "*MK*" leads to the click of product items associated with the brand "*Michael Kors*". We can reasonably infer a strong connection between the surface form value "*MK*" and the canonical form value "*Michael Kors*" if this association occurs in many queries.

Motivated by those, we propose a unified **QUE**ry **A**ttribute Value Extraction in E**CO**mmerce (**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:

- 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.
- 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.
- 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 $\mathbf{X_i} = [x_1, x_2, ..., x_M]$ with M tokens, the goal of NER is to predict a tag sequence $\mathbf{Y_i} = [y_1, y_2, ..., y_M]$. We use the BIO [22] tagging strategy. Specifically, the first token of an entity mention with each entity type $C_o \in C$ (C is the entity type set) is labeled as $\mathrm{B-C_o}$; the remaining tokens inside that entity mention are labeled as $\mathrm{I-C_o}$; 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 $D_l = \{(\mathrm{X}_i^l, \mathrm{Y}_i^l)\}_{i=1}^{N_l}$, which is manually annotated by human annotators; 2) large-scale weakly-labeled data $D_w = \{(\mathrm{X}_i^w, \mathrm{Y}_i^w)\}_{i=1}^{N_w}$, where $N_l \ll N_w$. 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 $\mathbf{X_i} = [x_1, x_2, ..., x_M]$ with M tokens, QUEACO NER predicts a tag sequence $\widetilde{\mathbf{Y}}_\mathbf{i} = [\tilde{y_1}, \tilde{y_2}, ..., \tilde{y_M}]$. Given a entity type $C_o \in C$ (e.g., brand) and the NER prediction $\widetilde{\mathbf{Y}}_\mathbf{i}$, we can extract the query term $\mathrm{X}_i^{C_o}$ as the surface form attribute value for the entity type C_o. Assume that we have a diverse set of canonical form product attribute values $\mathbf{V}$ for the entity type C_o. For each canonical form attribute value $v \in \mathbf{V}$, we can define the relevance given the query X_i as

$$P(C_o = v|\mathrm{X}_i) = \frac{\sum_{d \in D} n(d, \mathrm{X}_i) \mathbb{1}(d_{C_o} = v)}{\sum_{d \in D} n(d, \mathrm{X}_i)}$$

where $n(d, \mathrm{X}_i)$ 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. $\mathbb{1}(d_{C_o} = c)$ indicates whether the product d is indexed with the value c for the entity type C_o. 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 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_o by $\arg\max P(C_o = \mathbf{V}|\mathrm{X}_i)$ as the normalized canonical form for the surface form attribute value $\mathrm{X}_i^{C_o}$.

2.2 Query NER Base Model

The recent emergence of the pre-trained language models (PLMs) such BERT [10] has achieved superior performance on a variety of public NER datasets. However, existing query NER works [4, 9, 19, 38] 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 [4]. Liang et al. [23] 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 [4, 35] 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 [33] 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.

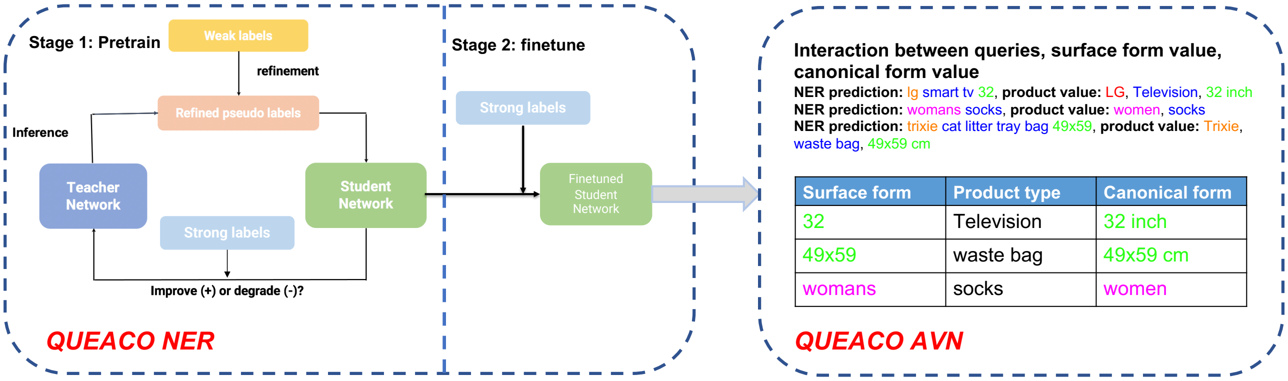
3.1 Overview

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 [21, 46]. 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 $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_S$. We use $f(\mathrm{X}; \boldsymbol{\theta}_T)$ and $f(\mathrm{X}; \boldsymbol{\theta}_S)$ denote the NER predictions of the query X for the teacher and student, respectively. $f(\mathrm{X}; \boldsymbol{\theta}_T)$ 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: $\mathcal{L}(f(\mathrm{X}; \boldsymbol{\theta}_T), f(\mathrm{X}; \boldsymbol{\theta}_S))$.

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



Interaction between queries, surface form value, canonical form value

NER prediction: lg smart tv 32, **product value:** LG, Television, 32 inch
NER prediction: womans socks, **product value:** women, socks
NER prediction: trixie cat litter tray bag 49x59, **product value:** Trixie, waste bag, 49x59 cm

Surface form	Product type	Canonical form
32	Television	32 inch
49x59	waste bag	49x59 cm
womans	socks	women

Figure 2: An overview of the proposed framework QUEACO, showing how weakly behavioral data contributes to the two interdependent stages of QUACO. Entity labels: Brand, ProductLine, Size, ProductType, namedPersonGroup, Color, Audience.

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 $\mathrm{X}_i^w = [x_1^w, x_2^w, ..., x_M^w]$, we convert the soft predictions of the teacher network into the hard pseudo labels, i.e., $\mathrm{Y}_i^p = \arg\max f(\mathrm{X}_i^w; \theta_T) = [y_1^p, y_2^p, ..., y_M^p]$. Additionally, we have weak labels $\mathrm{Y}_i^w = [y_1^w, y_2^w, ..., y_M^w]$ 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 $\mathrm{Y}_i^r = [y_1^r, y_2^r, ..., y_M^r]$ are generated by:

$$y_j^r = \begin{cases} y_j^p, & \text{if } y_j^w = \mathrm{O} \\ y_j^w, & \text{otherwise} \end{cases}$$

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. [29]. 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,

$$\begin{aligned} \min_{\theta_T} \quad & \mathcal{L}_{S,l}(\theta_S^{t+1}(\theta_T)) \\ \text{s.t.} \quad & \theta_S^{t+1}(\theta_T) = \arg\min_{\theta_S} \frac{1}{N_w} \sum_{i=1}^{N_w} \ell(\mathrm{Y}_i^r, f(\mathrm{X}_i^w; \theta_S^t)). \end{aligned}$$

where ℓ is the cross-entropy loss. The ultimate goal is to minimize the loss of the student θ_S^{t+1} on the strongly-labeled data after learning from the refined pseudo labels Y_i^r, i.e., $\mathcal{L}_{S,l}(\theta_S^{t+1}(\theta_T))$, which is a function of the teacher's parameters θ_T. $f(\mathrm{X}_i^w; \theta_S^t)$ 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 $\arg\min_{\theta_S}$ with one step gradient update of θ_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^r in order to move closer to the teacher,

$$\mathcal{L}_S = \frac{1}{N_w} \sum_{i=1}^{N_w} \ell(\mathrm{Y}_i^r, f(\mathrm{X}_i^w; \theta_S))$$

We update student network parameter θ_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 (θ_S^t) and after the update (θ_S^{t+1}, learning on the refined pseudo-labeled data),

$$\mathcal{L}_{S,1}^{(t)} = \frac{1}{N_l} \sum_{i=1}^{N_l} \ell(\mathrm{Y}_i^l, f(\mathrm{X}_i^l; \theta_S^t)),$$

$$\mathcal{L}_{S,1}^{(t+1)} = \frac{1}{N_l} \sum_{i=1}^{N_l} \ell(\mathrm{Y}_i^l, f(\mathrm{X}_i^l; \theta_S^{t+1})).$$

The difference between $\mathcal{L}_{S,1}^{(t)}$ and $\mathcal{L}_{S,1}^{(t+1)}$, i.e., $\lambda_{\text{meta}} = \mathcal{L}_{S,1}^{(t+1)} - \mathcal{L}_{S,1}^{(t)}$, 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 λ_{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 $\mathcal{L}_{\text{ssl}}$ and a meta learning loss $\mathcal{L}_{\text{meta}}$:

$$\mathcal{L}_{\text{T}} = \mathcal{L}_{\text{ssl}} + \mathcal{L}_{\text{meta}}.$$

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

$$\mathcal{L}_{\text{ssl}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{reg}}.$$

The supervised loss $\mathcal{L}_{\text{sup}}$ is defined as

$$\mathcal{L}_{\text{sup}} = \frac{1}{N_l}\sum_{i=1}^{N_l} \ell(\mathrm{Y}_i^l, f(\mathrm{X}_i^l;\boldsymbol{\theta}_T))$$

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

$$\mathcal{L}_{\text{reg}} = -\frac{1}{N_w * M}\sum_{i=1}^{N_w}\sum_{j=1}^{M}\frac{f(x_{ij}^w;\boldsymbol{\theta}_T)}{\tau}\log(f(\tilde{x}_{ij}^w;\boldsymbol{\theta}_T)))$$

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

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

$$L_{\text{meta}} = \frac{\lambda_{\text{meta}}}{N_w}\sum_{i=1}^{N_w}\ell(\mathrm{Y}_i^r, f(\mathrm{X}_i^w;\boldsymbol{\theta}_T))$$

The performance variation of the student network on the strongly-labeled data is formulated as the feedback signal λ_{meta} to dynamically adapt the teacher network's pseudo-labeling strategies. The teacher and student can have the same encoder (e.g., DistilBERT [33]), or a larger teacher for better prediction (e.g., BERT [10]) 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 $\mathrm{Y}_i^{\text{pt}} = P(C_o = \mathrm{V}|\mathrm{X}_i)$ of query X_i^w 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 $D = \{(\mathrm{X}_i^w, \mathrm{Y}_i^{\text{pt}})\}_{i=1}^{N_w}$ to train a multi-label query classification model [15, 18, 24] for predicting the product type distribution of less frequent queries. To meet the latency constraint, we also use DistilmBERT 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

Table 2: Case study on surface & behavior-based product type.

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, 8, 36] 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 by aggregating the query search logs. Then we can associate surface form brand $X_i^{w,b}$ and the most relevant behavior-based brand b_i through the query X_i^w. Given a surface form brand value m and a canonical form brand value v, we can define the mapping probability between them as,

$$P(v|m) = \frac{\sum_i^{N_w}\mathbb{1}(X_i^{w,b} = m, b_i = v)}{\sum_i^{N^w}\mathbb{1}(X_i^{w,b} = m)}.$$

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, 39]. 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, 13, 39, 43, 44]. 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:

$$P(v|m,p) = \frac{\sum_i^{N_w}\mathbb{1}(X_i^{w,b} = m, b_i = v, Y_i^{\text{pt}} = p)}{\sum_i^{N_w}\mathbb{1}(X_i^{w,b} = m, Y_i^{\text{pt}} = p)}$$

Case#	Query	entity	surface form	canonical form
1	lg smart tv 32	size	32	32 inch
2	fish tank 32	size	32	32 gallon
3	apple craft paint	brand	apple	apple barrel
4	apple macbook pro	brand	apple	Apple computer

Table 3: Case study on surface & canonical value.

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), Portugal (Pt), Turkish (Tr). The weakly-labeled dataset does not have Zh, CS, NI, and PI 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 [27]. 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.

Dataset	#Train	# Type	%Coverage	Span Precision	Span Recall
En	14144225	11	42.64	78.50	47.53
De	2004144	11	48.55	83.18	52.35
Es	322435	11	45.79	82.24	51.32
Fr	504309	11	49.00	81.15	51.56
It	475594	11	48.87	81.69	50.82
Jp	241078	11	20.80	67.67	25.53
Pt	134458	11	33.91	80.83	32.23
Tr	23980	11	32.87	86.12	34.95
Total	17850787	11	43.21	79.80	48.04

Table 5: The data statistics of weakly-labeled NER dataset. Type and Coverage denote the number of entity type and the ratio of non-O entity.

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

4.1.3 ***Analysis of the Base Encoder.*** We benchmark the DistilmBERT performance with the baseline models in the query attribute extraction literature. All RNN experiments use FastText multilingual word embeddings [7] and the TARGER implementation [5].

- RNN models: BiLSTM, BiGRU, BiLSTM-CRF and BiGRU-CRF models are benchmarked for the Home Depot query NER model [4].
- BiLSTM-CNN-CRF [20, 25] is the state-of-the-art NER model architecture before BERT [10, 45].
- DistilmBERT baselines: 1) DistilmBERT (Single) means separately finetuning DistilmBERT on the strongly-labeled data for each single language. 2) DistilmBERT (Multi) means finetuning DistilmBERT 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
DistilmBERT (Single)	71.72	74.16	72.92
DistilmBERT (Multi)	73.33	75.29	74.29

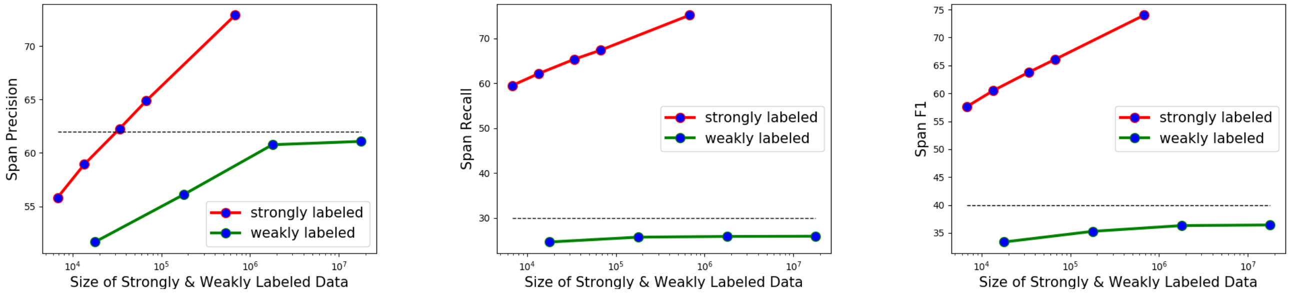
Table 6: Comparison of different encoders.

As shown in Table 6: the DistilmBERT has better performance than other non-BERT baselines. Furthermore, finetuning DistilmBERT 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 NER literature, where only weakly-labeled data is available. All experiments use DistilmBERT as the base NER model for a fair comparison.

In Figure 3, we subsample the strongly and weakly-labeled dataset and we find:

- The precision and recall of the model trained with the weakly-labeled data do not change much when the training data size



(a) Span Precision **(b) Span Recall** **(c) Span F1**

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 3a, 3b and 3c, span-level precision, recall and macro-f1 are shown.**

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 strong-labeled data can already achieve 48.78% recall.

The findings are consistent with the conclusion of BOND [23] that pre-trained language models can easily overfit to incomplete weak labels. And this explains why the existing query NER works [4, 9, 19, 38] 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.

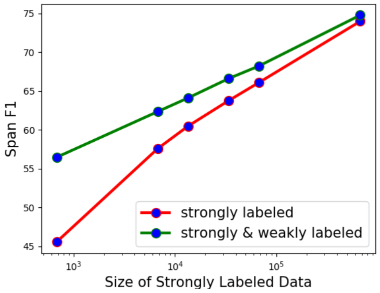


Figure 4: Size of Strongly Labeled Data vs. Macro span-level F1. "strongly labeled": a baseline that finetunes DistilmBERT with the strongly labeled data, "strongly & weakly labeled": a baseline that pretrains Distil-mBERT with weakly labels and then finetunes it on the strongly labeled data.

*4.1.5 **Implementation Details of QUEACO**.* We employ the DistilmBERT [33] 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 σ is tuned amongst {0.01, 0.1, 1.0}. The temperature factor for smoothness τ is tuned amongst {0.5, 0.6, 0.7, 0.8, 0.9}. The threshold ϵ 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.

*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 [41] 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 [26].
  - Weighted Weakly Supervised Learning (Weighted WSL): WSL with weighted loss, where weakly-labeled samples have a fixed smaller weight and strongly-labeled samples have weight = 1. We tune the weight and present the best result.
  - Robust WSL: WSL with mean squared error loss function, which is robust to label noises [12].

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: distillmBERT)			
QUEACO (Teacher: distilmBERT)	74.44	76.35	75.38
QUEACO (Teacher: mBERT)	**74.48**	**76.41**	**75.44**
Δ	(**+0.51**)	(**+0.36**)	(**+0.46**)

Table 7: Comparison between QUEACO and baseline methods on macro span-level F1.

- BOND (hard/soft): BOND [23] employs a state-of-the-art two-stage teacher-student framework with hard pseudo-labels or soft pseudo-labels [40].
- BOND (soft-high): only uses the soft pseudo-labels, with high confidence selection for student network training in the BOND framework.
- BOND (NoisyStudent): applies noisy student [41] to the BOND framework.

4.1.7 *Main results.*

From Table 7 and 8, our results obviously demonstrate the effectiveness of our proposed QUEACO query NER model:

- The proposed QUEACO query NER model achieves the state-of-the-art performance. More specifically, we can improve upon the best weakly-supervised baseline model by a margin of 0.4% on macro span-level F1. QUEACO query NER model with mBERT as the teacher network can further enhance the model performance.
- We also find weak labels improve by 1.09% w.r.t the best semi-supervised result, showing the weak labels have useful information if utilized effectively.
- Table 8 compares the span F1 between the baseline DistilmBERT model and the QUEACO query NER model for each language. We can observe consistent performance improvement for the high resource languages (En, De, Es, Fr, It, Jp). On the other hand, we observe performance drop for those low resources languages with a few or no weakly-supervised data (Cs, Nl, Pl, Tr). Pt is also a low-resource language but observes significant performance improvement because we have more than 100k weakly supervised training data for Pt. We believe we can further improve the performance of those low-resource languages if more weak supervised data is collected.

4.1.8 *Ablation Study.*

- QUEACO w/o student feedback $\mathcal{L}_{\text{meta}}$: use a fixed teacher network to generate pseudo labels for a student network.

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**)
Nl	False	75.46	75.30 (**-0.16**)
Pl	False	79.71	79.43 (**-0.28**)
Pt	True	58.24	62.00 (**+3.76**)
Tr	True	72.12	71.80 (**-0.32**)

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

- 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 finetuning.

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.

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

Table 9: Ablation study.

4.2 QUEACO Attribute Value Normalization

4.2.1 *Product type AVN.*

In the query NER, the span-level macro 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 classification model, as described in section 3.3.1, on a golden set. We can conclude the query classification approach, trained with weakly-labeled data, is more suitable to product type attribute extraction than query NER.

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

Table 10: Product type attribute value extraction performance.

4.2.2 AVN for other attributes. In Table 11, we show some attribute normalization result for brand, color and size attributes, using our proposed method. We can see that our proposed method is effective in finding common surface attributes, including:

- spelling error: brand "*Michael Kors*" is often misspelled as "*Micheal Kors*", "*Levi's*" is often misspelled as "*levi*";
- spelling invariants: for example, "*3 by 5*" and "*3x5*" are different variants with the same meaning.
- abbreviation: for example, "*mk*" is the abbreviation for "*Micheal Kors*", "*wd*" is the abbreviation for "*Western Digital*", "*in*" in the mention "*8 in*" is the abbreviation for unit "*inches*".

attribute	surface form	product type	canonical form
size	3 by 5	rug	3x5
size	2 pack	air filter	Value Pack (2)
size	28 foot	ladder	28 Feet
size	10.5 inch	screen protector	10.5 Inches
size	8 in	toy figure	8 inches
color	golden	belt	Gold
color	turquoise	dress	blue
color	navy blue	dress	blue
brand	levi	underpants	Levi's
brand	mk	watch	Michael Kors
brand	Micheal Kors	watch	Michael Kors
brand	wd	computer drive	Western Digital

Table 11: QUEACO attribute normalization result.

5 QUEACO ONLINE DEPLOYMENT

5.1 Online End-to-End Evaluation

We conducted an end-to-end evaluation of QUEACO on real-world search traffic. We have two evaluation metrics: span-level precision and token-level coverage. For span-level precision, we resort to a crowdsourcing data labeling platform called Toloka[1] and the reported overall precision of the QUEACO system is 97%. Since the query attribute value extraction is an open-domain problem, the human annotators cannot verify the recall of the extracted attribute spans. Therefore, we calculate token-level coverage, i.e., the percentage of tokens annotated by QUEACO, as an approximation of recall. The token-level coverage increased by 38.2% compared to the current system.

5.2 Application: Extracted Attribute Value for Product Reranking

To validate the effectiveness of QUEACO signal on the product search system, we design a downstream task, *product reranking*, 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%, proving that the QUEACO signals produce positive effects for the search engine to provide more relevant and accurate products to the customers.

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 [11, 30]. Later, several researchers [28, 31] 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 [17, 20, 25], and have achieved promising results. Following this trend, recent works on the product attribute value extraction task [27, 42, 47] also adopt variants of the BiLSTM-CRF model architecture. In addition, some recent studies have explored BERT-based [10] Machine Reading Comprehension (MRC) [42] and Question & Answering (Q&A) [37] formulation.

Query attribute value extraction works [4, 9, 19, 38] also employ the sequence tagging formulation and adopt BiLSTM-CRF model architectures as well as its variants. Recent works [4, 38] 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 the human labeling efforts, distant supervision are applied to various natural language processing tasks [16, 34]. 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 [23, 35]. Most of the existing works on NER with distant supervision [23, 35] focus on the settings that we can only access distant supervision for model training. Existing query NER works [4, 38] also only use the distant supervision, generated from partial query tagging, for NER model training.

However, in practice, we often can access both strongly-labeled data and a large amount of distant supervision. The strongly-labeled data, though expensive to collect, is critical to the query NER model performance.

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.

[1] https://toloka.yandex.com

REFERENCES

[1] Oshin Agarwal and Daniel M Bikel. 2020. Entity linking via dual and cross-attention encoders. *arXiv preprint arXiv:2004.03555* (2020).

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*. IEEE, 1–8.

[3] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. 179–188.

[4] Xiang Cheng, Mitchell Bowden, Bhushan Ramesh Bhange, Priyanka Goyal, Thomas Packer, and Faizan Javed. 2020. An End-to-End Solution for Named Entity Recognition in eCommerce Search. *arXiv preprint arXiv:2012.07553* (2020).

[5] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. Targer: Neural argument mining at your fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 195–200.

[6] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *TACL* 4 (2016), 357–370.

[7] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *arXiv preprint arXiv:1710.04087* (2017).

[8] Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Hinrich Schütze, and Stefan Rüd. 2014. The SMAPH system for query entity recognition and disambiguation. In *Proceedings of the first international workshop on Entity recognition & disambiguation*. 25–30.

[9] Brooke Cowan, Sven Zethelius, Brittany Luk, Teodora Baras, Prachi Ukarde, and Daodao Zhang. 2015. Named entity recognition in travel-related search queries. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 3935–3941.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[11] Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *ACM SIGKDD Explorations Newsletter* 8, 1 (2006), 41–48.

[12] Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[13] Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning Dense Representations for Entity Retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 528–537.

[14] Joan Guisado-Gámez, David Tamayo-Domenech, Jordi Urmeneta, and Josep Lluis Larriba-Pey. 2016. ENRICH: A Query Rewriting Service Powered by Wikipedia Graph Structure. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 10.

[15] Homa B Hashemi, Amir Asiaee, and Reiner Kraft. [n.d.]. Query intent detection using convolutional neural networks.

[16] Wenqi He. 2017. Autoentity: automated entity detection from massive text corpora. (2017).

[17] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[18] Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 414–419.

[19] Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. 2016. Recognizing salient entities in shopping queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 107–111.

[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 260–270.

[21] Dong-Hyun Lee et al. [n.d.]. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

[22] Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zheng, and Fei Huang. 2012. Joint bilingual name tagging for parallel corpora. In *CIKM*. 1727–1731.

[23] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: Bert-Assisted Open-Domain Named Entity Recognition with Distant Supervision. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[24] Heran Lin, Pengcheng Xiong, Danqing Zhang, Fan Yang, Ryoichi Kato, Mukul Kumar, William Headden, and Bing Yin. 2020. Light Feed-Forward Networks for Shard Selection in Large-scale Product Search. (2020).

[25] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1064–1074.

[26] Gideon S Mann and Andrew McCallum. 2010. Generalized Expectation Criteria for Semi-Supervised Learning with Weakly Labeled Data. *Journal of machine learning research* 11, 2 (2010).

[27] Kartik Mehta, Ioana Oprea, and Nikhil Rasiwasia. 2021. LaTeX-Numeric: Language-agnostic Text attribute eXtraction for E-commerce Numeric Attributes. *arXiv preprint arXiv:2104.09576* (2021).

[28] Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *arXiv preprint arXiv:1608.04670* (2016).

[29] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11557–11568.

[30] Katharina Probst, Rayid Ghani, Marko Krema, Andrew E Fano, and Yan Liu. 2007. Semi-Supervised Learning of Attribute-Value Pairs from Product Descriptions.. In *IJCAI*, Vol. 7. 2838–2843.

[31] Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 1557–1567.

[32] Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *EMNLP*. 1156–1167.

[33] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108 (2019).

[34] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[35] Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning Named Entity Tagger using Domain-Specific Dictionary. In *EMNLP*. 2054–2064.

[36] Chuanqi Tan, Furu Wei, Pengjie Ren, Weifeng Lv, and Ming Zhou. 2017. Entity Linking for Queries by Searching Wikipedia Sentences. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 68–77.

[37] Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to Extract Attribute Value from Product via Question Answering: A Multi-task Approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 47–55.

[38] 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. In *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*. 149–154.

[39] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6397–6407.

[40] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*. 478–487.

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

[42] 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.

[43] 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.

[44] 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).

[45] 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.]).

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

[47] 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.