Explicit Attribute Extraction in E-Commerce Search

Robyn Loughnane, Jiaxin Liu, Zhilin Chen, Zhiqi Wang,
Joseph Giroux, Tianchuan Du, Benjamin Schroeder, Weiyi Sun
Wayfair LLC
{rloughnane,jliu2,zchen3,zwang6,jgiroux,tdu1,beschroeder,wsun1}@wayfair.com

Abstract

This paper presents a model architecture and training pipeline for attribute value extraction from search queries. The model uses weak labels generated from customer interactions to train a transformer-based NER model. A two-stage normalization process is then applied to deal with the problem of a large label space: first, the model output is normalized onto common generic attribute values, then it is mapped onto a larger range of actual product attribute values. This approach lets us successfully apply a transformer-based NER model to the extraction of a broad range of attribute values in a real-time production environment for e-commerce applications, contrary to previous research. In an online test, we demonstrate business value by integrating the model into a system for semantic product retrieval and ranking.

Keywords: attribute extraction, e-commerce search, named-entity recognition

1. Introduction

E-commerce applications use a range of structured information that describe their catalog of products or services. This allows customers to browse via a taxonomy of product categories and filters, using the structured information directly to narrow down their search. For example, a customer may click on categories “Furniture”, then “Bedroom Furniture”, then “Nightstands”, then the attribute “Color:Blue”, to find an item they like.

However, when customers search using natural language, a mapping of the query to relevant structured information must happen automatically. This mapping can principally be used in two ways to improve search engine performance: first, scores that judge the relevance of a query to particular product information can be used to inform a relevance-based ranking (Liu et al., 2022). Second, explicit filters can be applied dynamically, restricting the result set to products with attributes matching those identified in the query, in particular where the query interpretation has a high degree of confidence in its prediction.

Attribute value extraction (AVE) in search can be approached in different ways. While it is possible to e. g. use a multi-label classification task at query level, we opted for a token-level classification approach related to the more general problems of slot filling and named-entity recognition (NER). This yields a more intuitive mapping from spans in the query to explicit filtering, providing a more transparent user experience.

Two major challenges of a search application are, first, that text input is short and contains non-standard grammar and spelling; and, second, the stringent latency requirements for model inference, which needs to be computed online in real time. Both of these make the use of transformers,[1] the current state of the art in NER, challenging since they use contextual information typically only found in longer strings, and they contain a high number of model weights. In addition, there are challenges with the large size of the label space and the fact that NER training and evaluation requires token-level labelling, which can be prohibitively expensive. Indeed, Xu et al. (2019) claim the search problem is not solvable with an NER-style model.

To the contrary, we demonstrate the feasibility of a transformer-based NER-style model. Using a lightweight transformer that meets the latency requirements, we show that it performs almost as well as a larger model, indicating data quality may be more important than model size. We show a way to create quality data via a regime of hierarchical normalization applicable to any e-commerce catalog to deal with the problem of a large label space. We use weak labeling to create abundant inexpensive labeled data. We report model accuracy and nDCG gains and, lastly, demonstrate the business value of this approach via an online A/B test.

2. Related Work

In search systems, query understanding models aim to decipher and interpret users’ search goals so as to aid the downstream retrieval and ranking applications (Deng and Chang, 2020). In the general web search domain, query understanding consists of two dimensions: intent classification and topic detection (Brenes et al., 2009). Query intent classi-

[1] Transformer-based models are among the state of the art in NER tasks, e. g. the BERT-based model from Li et al. (2020) for Ontonotes v5 (Weischedel et al., 2013).

The Seventh Workshop on e-Commerce and NLP (ECNLP 7), pages 125–133
21 May, 2024.

User query text	*mod bright red sofa*			
Query tokens	mod	bright	red	sofa
Identified text spans	mod	bright red		sofa
NER label	`STYLE`	`COLOR`		`O`
Intermediate attribute values	Modern	Red		
Final attribute title	Product Styles	Upholstery Color		
Final attribute value	Modern & Contemporary	Red		

Table 1: An example user query with two attributes identified

fication determines users' desired search actions, such as informational, navigational and transactional (Broder, 2002; Rose and Levinson, 2004). Query topic detection maps users' search queries to a predefined taxonomy of topics, such as sports, entertainment and so on (Li et al., 2005). Intent classification and topic detection problems also exist in the e-commerce domain. Both intents and, in particular, query topic categories in e-commerce can differ greatly depending on industry vertical (Tsagkias et al., 2021), but the the overarching principles are applicable to any e-commerce domain.

E-commerce query topic detection maps user search queries to the structured product catalog taxonomy (Wen et al., 2019). For instance, the search query "KitchenAid 4.5 Qt" maps to products in the "Small Appliances/Mixers" category, with attributes of "Brand: KitchenAid" and "Capacity: 4 - 5 Qt". The query-to-product-type mapping part of the problem ("KitchenAid 4.5 Qt" to "Small Appliances/Mixers") has been well studied as text classification (Hashemi et al., 2016; Kim et al., 2016; Lin et al., 2020).

There has been less prior work focusing on the query AVE and normalization part of the problem. Following Luo et al. (2022), a distinction can be made in AVE between *explicit* (e.g. Cowan et al. 2015; Kozareva et al. 2016; Wen et al. 2019; Cheng et al. 2020; Zhang et al. 2021) and *implicit* or latent attributes (e.g. Wu et al. 2017). Explicit attributes are represented by a span of text in the user query, whereas implicit attributes are not. Implicit attributes can be use directly, whereas extracted explicit attribute spans usually need to first be normalized to match misspellings or non-canonical forms against structured product data, as in the current approach.

A major challenge for AVE in e-commerce queries is the sparsity of available data (Cheng et al., 2020; Wen et al., 2019), especially where the number of product attributes is high with a long tail distribution of rare attributes. Cheng et al. (2020) address this via an iterative learning framework that utilizes both synthetic data and human annotated data to extract product categories and brands. Wen et al. (2019) went without human annotated data and leveraged only user behavior logs to build a sequential tagging model for attribute detection. Zhang et al. (2021) use a teacher-student network to better exploit a combination of human annotated labels and weak synthetic labels in their sequential tagging model. We address this problem via weak and synthetic labeling (Section 3.3) to generate data; and reducing the label space via normalization (Section 3.5).

Normalization is a challenge for explicit AVE and some studies leave it out altogether despite its necessity in an e-commerce setting (Zhang et al., 2021). Similarity measures are one possible approach per Putthividhya and Hu (2011), who use *n*-gram sub-string similarity to normalize results to match dictionary entries. Cowan et al. (2015) use a gazetteer approach for matching identified spans to attribute entities, as do Zhang et al. (2021) after a frequency analysis of user behavior. We propose a two-step normalization process using gazetteers: first identified spans are mapped on to a pre-defined schema of intermediate attribute value concepts, before finally mapping them on to product attribute data (Section 3.5). An example search query with intermediate and final attributes identified is given in Table 1.

An additional constraint in any business context is that there must be enough monetary value to justify the complexity of the system implemented. Business applications of attribute extraction from search queries include product retrieval (Cheng et al., 2020), recall filtering (Wen et al., 2019) and ranking (Cheng et al., 2020; Wen et al., 2019; Wu et al., 2017). In an A/B test, the current system brings value in ranking (Section 4) even on top of a semantic search system per Liu et al. (2022), where no previous studies known to the current authors have explicitly demonstrated this.

Outside of the search context, classic NER approaches inform the current work. NER is typically defined as the identification of phrases that contain the names of persons, organizations and locations (Tjong Kim Sang and De Meulder, 2003). Although many attribute values are not proper nouns, the mechanics of the problem in regards to span identification in written text are similar, and the phrases to be identified can be sorted into thematic groups in a similar fashion to those in classic NER. Publicly available, pre-trained, transformer-based models like BERT (Devlin et al., 2019) that create context-

	Human labels	Weak labels	Test set	F1 COLOR	F1 STYLE	F1 BRAND	F1 overall
Human only	90	0	33	**0.83**	0.447	0.525	0.453
Weak only	0	82	33	0.686	0.504	0.656	0.56
Weak and synthetic only	0	146	33	0.687	0.516	0.662	0.568
Production model	90	146	33	0.818	**0.625**	**0.71**	**0.654**

Table 2: Sample model training & test set sizes & statistics (data size given in thousands)

sensitive word embeddings enable token classification on top of these networks to be used as viable alternative to traditional NER methods.

3. Problem Definition & Methodology

The problem at hand is thus to map explicit attributes within a user search query string onto the structured data of a set of products, within the practical constraints of the live production environments of an e-commerce website. The structured data targeted in particular here are *attribute titles* and *attribute values*. Canonical examples of attribute titles are "Upholstery Color" and "Product Styles", which can have attribute values like "Red" or "Modern & Contemporary", and which are both distinct from a product's category, e. g. "Sofas".

For explicit AVE, a number of studies use methodologies from NER, such as a conditional random field (CRF) (Cowan et al., 2015), long short-term memory network (LSTM) plus CRF (Kozareva et al., 2016; Wen et al., 2019), bidirectional gated recurrent unit (GRU) network with a CRF layer plus LSTM-based character embeddings (Cheng et al., 2020) and, more recently, pre-trained transformer-based language models (Zhang et al., 2021; Luo et al., 2022). Other studies use a question answering approach (Shinzato et al., 2022; Xu et al., 2019). The model presented here is an NER model, using token classification on top of a distilled pre-trained transformer-based language model. Distilled versions of larger language models (Sanh et al., 2019) enable performance to be largely be maintained without the drag on latency.

As this paper focuses on explicit AVE, normalization is required, which we approach using an initial gazetteer, plus a second layer of custom normalization depending on the attribute type. Contrary to Xu et al. (2019), who claim that an NER-based model cannot deal with a large attribute space, we demonstrate how this can indeed be done by using this two-stage normalization approach.

The model training pipeline consists thus of multiple steps. First, human annotation using a predefined attribute schema is conducted (Section 3.2). The human annotation is supplemented by weak and synthetic labels generated from the structured data in the catalog (Section 3.3). The model itself is a token-classification transformer network (Section 3.4). The identified span is normalized before use (Section 3.5). The model is evaluated via both offline and online means (Section 4).

3.1. Production Environment & Baseline

The current system replaces the previous rules-based system in production, which is applied to around half of all search experiences, covering just under 15% of the most common distinct search queries in a given month. The other half of search experiences are characterized by a long tail of diverse search queries, which previously had not had attributes extracted at all. However, some attribute information is implicitly used in the embedding-based semantic search product retrieval model, to the extent that this attribute information was available in the product information used to train the model (product name, product category and so forth). However, to use this information explicitly for relevance scoring or dynamic filtering, individual attributes need to be predicted and then mapped to product data.

A challenge peculiar to the current production system is that it is not set up to have consistent attribute information across product categories. Product attributes are specific to a given product category. This means that the attributes of a product of type "Chair" include "Upholstery Material" and "Leg Material", whereas a product of type "Saucepan" includes "Lid Material". In addition, attributes of different product categories may have distinct IDs and values, even where the attribute title is the same or similar, e.g. "Primary Material". This puts the number of attributes at around 86,000 distinct attribute ID-title pairs with 314,000 distinct attribute ID-value pairs. This makes a direct classification difficult due to data sparsity for the long tail of less-common attributes; in the current paper we address this challenge via two-stage normalization (Section 3.5).

3.2. Human Annotation

Human annotators were provided with lists of historical customer search queries to label with up to three attributes. All queries were initially labeled by two human annotators. If both annotators did not exactly agree on an attribute, it was submit-

	F1 COLOR	F1 STYLE	F1 BRAND	F1 overall
Production model	0.818	0.625	0.71	0.654
No noisy spelling	0.838	0.617	0.69	0.651
No synthetic subjects	0.76	0.614	0.702	0.65
No synthetic SKU IDs	0.754	0.654	0.704	0.649
No synthetic product categories	0.759	0.619	0.694	0.63

Table 3: Ablation study

ted to a third annotator to make a final decision. Annotators were asked to label non-overlapping sub-strings of the original query with attribute titles and values from a pre-defined schema of intermediate attributes.

The schema of intermediate attributes was created from a frequency analysis of commonly searched attributes from historical user interactions. This schema is used both for annotation and as the intermediate attributes for the first stage of normalization. Annotators could label attribute values outside of the predefined schema by selecting an umbrella “Other” option. Common attribute values identified in this way were added to the predefined schema when appropriate.

For model training, the human annotation is converted to BIO (beginning-inside-outside) labels using the IOB2 schema (Krishnan and Ganapathy, 2005) for the attribute-type named entities $\mathcal{E} = \{$BRAND, MATERIAL, COLOR, DIMENSION, SUBJECT, LIFE_STAGE, FEATURE, LOCATION, SIZE, FINISH, PRICE, STYLE, SHAPE, PATTERN, NUMBER_ITEMS, NUMBER_COMPONENTS$\}$.

The inter-annotator agreement (IAA) on entity level is around 68% F1, calculated by holding one annotator’s labels as ground truth and the other as system output. Among the entities that the annotators agreed on, the option normalization accuracy is around 94%.

3.3. Weak Labels & Synthetic Data

To produce weak labels, a variety of attributes and other structured product data (e.g. product category for O labels per the BIO schema) from known add-to-cart events were string-matched against the preceding user query. As conflicting information came from the various sources, an unweighted majority vote was then applied to the candidates per Ratner et al. (2017). This allowed the use of the available diverse structured data sources to reduce noise. In the future, other methods for weak label selection could be applied per Ratner et al. (2016). The resultant weak labels were used together with human-labeled data to train the NER model.

Zhang et al. (2021) use a similar system for getting large amounts of weak labels, and report that models trained with weakly labeled data alone (F1 0.6) are inferior to those trained with much less human labeled data alone (F1 0.62). However, in our system, the weak labels appear to be of higher quality than the human data, at least for some labels, as shown in Table 2, which may point to an opportunity to improve task design.

Adding generative model predictions from Flan-T5-XL (Chung et al., 2022) to the weak labels had a neutral affect on the performance of the model. Further experimentation with larger generative models is planned.

In addition to weak labels from customer add-to-cart events, synthetic search experiences and corresponding labels were created. Synthetic search queries were created in a number of ways: random distortions to create misspellings for existing labeled data; various subjects, e.g. animal types, from the structured product data with SUBJECT labels; and using SKU IDs and product category names as O labels. Adding additional O labels via product categories had the greatest positive effect as shown in the ablation study in Table 3.

3.4. Model Architecture

The training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ consists of N training examples where $(x^{(i)}, y^{(i)})$ is the *i*th instance consisting of a tuple of the input query $x^{(i)}$ and its labels $y^{(i)}$. A single search query is symbolized for the *i*th training example with a variable natural number, $M^{(i)}$, of tokens by a vector

$$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_{M^{(i)}}^{(i)}),$$

where each element, $x_j^{(i)}$, is a natural language token. The two-dimensional array $y^{(i)}$ represents the labels for the *i*th training sample, such that

$$y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_{M^{(i)}}^{(i)}),$$

where there are $M^{(i)}$ label vectors and each $y_j^{(i)}$ is an NER label in the form of a one-hot vector of the fixed size L of the label set.

The DistilBERT model (Sanh et al., 2019) from Huggingface (Wolf et al., 2020) was used to generate a two-dimensional array of logits corresponding to the labels as follows, where *f* is the DistilBERT feed-forward transformer network consisting of an embedding layer, five transformer block layers and a linear classifier:

$$\hat{y}^{(i)} = f(x^{(i)}).$$

The DistilBERT uncased pre-trained English base model was fine-tuned using the Huggingface NER pipeline with `seqeval` (Nakayama, 2018) as the metric for evaluation and multi-class cross-entropy loss, such that the loss function can be expressed as

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\sum_{j}^{M^{(i)}} \sum_{k}^{L} y_{jk}^{(i)} \log \left(\sigma(\hat{y}_j^{(i)})_k \right),$$

with σ, the softmax function, defined as follows:

$$\sigma(\hat{y}_j^{(i)})_k = \frac{e^{(\hat{y}_{jk}^{(i)})}}{\sum_l^L e^{(\hat{y}_{jl}^{(i)})}}.$$

Missing token labels, i.e. tokens where no BIO label could be identified during the weak labeling process, were ignored for purposes of calculating the model loss. The model was then fine-tuned with the objective of minimizing the loss over the training set using `AdamW`, i. e. Adam (Kingma and Ba, 2014) with weight decay (Loshchilov and Hutter, 2017).

3.5. Attribute Value Normalization

For span-identification-based AVE, a normalization step is required to translate the text span in the customer query and its NER label onto an attribute value and title respectively in the production system. In the simplest case, a one-to-one mapping of NER labels to attribute titles exists and only the text spans need to be normalized to attribute values. This is not the case in our system, where there is a many-to-many mapping of NER labels to attribute titles, which may have different names and unique IDs across product categories even when synonymous.

In the present study, we thus take a two-step normalization approach. First, in the human annotation, the spans identified by the annotators are matched to both an intermediate attribute title and an intermediate attribute value from a pre-defined schema of frequently searched attributes. The intermediate attribute schema groups synonymous and similar attribute titles and attribute values together, so “Uphostery Material” and “Leg Material” become just `MATERIAL`. Likewise, in the search query “crimson sofa”, the annotator would mark up the string “crimson” with “`COLOR`: Red”, where ‘Red’ is the intermediate attribute value. At inference time, a gazetteer created from the human annotation is used to normalize text spans onto the intermediate attribute values. An exception is numerical attributes and `BRAND`, which are mapped onto an intermediate attribute title only.

The second normalization step occurs when mapping the intermediate attribute titles and values onto actual product attributes in the structured data, where both the attribute title and value may have forms different to the intermediate ones and the number of attribute titles and values is much larger. The strategy used for the second stage of normalization varies, depending on the type of attribute as described below.

In the second normalization step at the attribute title level, some NER labels can easily be mapped onto a small number of class-agnostic attribute titles, as is the case for `BRAND`, `PRICE` and `STYLE`; all products have these attributes and they are agnostic to the product category. For other NER labels, such as `MATERIAL`, there are many different attribute titles that these could map to, many of which are dependent on the product category and call out the material of the components of the product, e. g. “Upholstery Material” and “Frame Material” are prominent in the “Beds” category, but “Top Material” and “Base Material” are used for “Dining Tables”. In this case, a statistical methodology similar to that used by Zhang et al. (2021) is required to map NER labels onto attribute titles.

Different approaches are also used at the attribute value level, depending on the type of attribute. Some attribute titles have an open set of attribute values. For `DIMENSION` and `PRICE`, there is no restriction on the value it can take, except that they are positive floating-point numbers. Likewise for `BRAND`, new brands are added to the catalog continuously, so these do not form a closed set. Other attribute titles, e.g. `STYLE`, do have a small, fixed set of attribute values, which do not change much over time. These values are determined in a curated way by domain-owners in the company.

For example, for `BRAND`, which has a large open set of attribute values, the first stage of normalization consists only of mapping the NER label onto its intermediate attribute title. For the second stage of normalization, the NER label is mapped onto one of two actual brand name attribute titles in the product data. The identified text spans can be mapped onto attribute values per the methodology used by Zhang et al. (2021). For the online experiment (Section 4) using the `BRAND` span identified by this model in the ranker, however, we did not use this methodology as it added the complication of needing the resulting mapping available in production. Instead, for the second layer of normalization, we used token-level Jaccard index as a measure of text similarity between the actual brand name attribute values and the span identified by the model. This avoids having to update a mapping as new brands are added to the catalog and is easy to implement in production.

For `STYLE`, an example of an NER label with a

	F1 COLOR	F1 STYLE	F1 BRAND	F1 overall
DistilBERT	0.818	**0.625**	**0.71**	**0.654**
RoBERTa base	0.797	0.619	0.682	0.632
RoBERTa large	0.793	0.601	0.683	0.623
SimCSE	**0.821**	0.591	0.677	0.626

Table 4: Change in performance by language model

	BRAND	MATERIAL	COLOR	DIMENSION	SUBJECT	FINISH
NER label	0.71	0.73	0.87	0.85	0.55	0.73
End to end	-	0.66	0.82	-	0.50	0.73

	SIZE	LIFE_STAGE	PRICE	NUMBER_ITEMS	SHAPE	PATTERN
NER label	0.84	0.83	0.75	0.87	0.80	0.59
End to end	0.83	0.81	-	0.84	0.77	0.59

	FEATURE	LOCATION	STYLE	NUMBER_COMPONENTS
NER label	0.41	0.76	0.64	0.58
End to end	0.41	0.74	0.63	0.52

Table 5: F1 scores for NER labels (i.e. attribute titles) & micro-averaged end-to-end (i.e. attribute value) F1 scores

closed set of intermediate attribute values, there is a single final attribute title across classes. Human annotation was initially used to create a mapping from surface forms to normalized forms of the attribute values. Per Zipf's law, the most commonly occurring 200 tokens cover the majority of token instances, so the effort for doing this is low, as this data is already annotated for training. Without any additional annotation, it results in a first-stage normalization accuracy of 0.985. Normalization accuracy is calculated as the sum of the end-to-end true positives (true positive intermediate attribute value and true positive NER label), divided by the true positives for the NER label, as measured against the human-annotated test set. An almost one-to-one mapping of the intermediate attribute title and values to structured data is then applied. In this instance, more advanced mapping methodologies would bring diminishing returns.

4. Evaluation

We used the seqeval package (Nakayama, 2018) to evaluate the model against a hold-out set of the human-annotated data. F1 scores at attribute title (i.e. NER label) and attribute value (i.e. end-to-end) level are reported in Table 5, with example experiments on combinations of synthetic, weak and human labels shown in Table 2. BRAND, DIMENSION and PRICE are not normalized to attribute values, so performance is only recorded for these at the attribute title level.

The current model meets the latency requirements, with an average speed in offline testing of 6 ms on GPU[2] and 12 ms on CPU[3] for a single query. In online testing, there were small but significant increases in the range of 0.5% to 2.6% in latency for search queries overall, although these were largely offset by preprocessing improvements after launch. Other transformer models, e.g. RoBERTa (Liu et al., 2019), SimCSE (Gao et al., 2021), were tested (Table 4) but they did not meet the latency requirements. In addition, the bigger transformer models did not give a boost in performance as shown Table 4; the hypothesis is that a larger model quickly over-fits for short search queries and a simpler model with fewer trainable parameters is preferable.

Offline ranking experiments were conducted where the brand name identified by the model in user queries was compared against the brand name and product name for products to be ranked. Token-level Jaccard index between these was calculated and used to boost relevant products. The product ranking with boosted brand names was then compared against existing product rankings. When using the brand attribute value from the model, an average lift of 4.65% was observed in the normalized discounted cumulative gain at rank 48 (nDCG$_{48}$).

The nDCG score was calculated in the typical fashion with

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)},$$

[2] Run on a machine with a single NVIDIA® Tesla® P100 GPU.

[3] Run on a machine with 16 CPUs of type Intel® Xeon® Processor E5-2630.

where the relevance value is calculated as

$$rel_i = \alpha * view_i + \beta * atc_i + \gamma * order_i$$

and $view_i$ is a binary value 1 if the product page was viewed and 0 if not. Analogously atc_i is whether an add-to-cart event occurred and $order_i$ is whether an order was placed. The weights α, β and γ are heuristically determined by the relative frequency of view, add-to-cart and order events respectively, with the most weight placed on orders. The DCG score is then normalized as follows:

$$nDCG_p = \frac{DCG_p}{IDCG_p},$$

where Ideal DCG (IDCG) is the maximum possible DCG score with the products ordered according to relevance value, with highest relevance values first.

An A/B test was conducted to evaluate the model's impact, with the variation using the methodology and the weights determined in the offline testing described above to rank a subset of products retrieved by a semantic search system per Liu et al. (2022). This resulted in significant overall lifts in product views (1.45%) and add-to-cart (3.45%) and conversion rates (2.99%). The variant helped to reduce friction during users' search journey and enabled users to find relevant products with less effort, as we observed significant decreases in reformulation rate (-1.25%) and landing page exit rate (-0.76%). Further A/B tests on other attributes and use cases are planned.

5. Conclusion & Discussion

In this paper, we presented an NLP application in the domain of e-commerce, focusing on identifying explicit attributes in customer search queries using weak labels and a transformer-based NER approach. To overcome the challenge of a large label space for attributes, we employed a two-stage normalization process. For the first stage, we leveraged human annotation to create a normalization gazetteer, while the second stage of normalization varied depending on the specific attribute under consideration.

The model met the strict latency requirements of an e-commerce website and was put into production. It showed significant business value via an initial A/B test using the BRAND output, and more tests are planned for additional attributes and use cases going forward, including dynamic filtering of products.

We showed that an increase model size and complexity did not necessarily increase the performance of the model, although further experiments with larger language models are planned. A significant boost was gained by adding product category as a separate O label, as well as by adding synthetic data. Contrary to previous studies, our human data did not outperform our weak and synthetic data on most labels, indicating a possible opportunity to improve both task design and the intermediate attribute schema.

Also left to explore are implicit attributes per Luo et al. (2022); using more powerful generative AI models to generate labels; and a multilingual version of the model for non-English catalogs.

Overall, this study demonstrates the successful application of weak labels and transformer-based NER for explicit attribute identification. The deployment of our model in a real-world setting, along with the observed business value, highlights its practical significance. Our proposed future directions open up exciting opportunities for further advancements in this domain.

6. Bibliographical References

David J. Brenes, Daniel Gayo-Avello, and Kilian Pérez-González. 2009. Survey and evaluation of query intent detection methods. In *Proceedings of the 2009 Workshop on Web Search Click Data*, WSCD '09, pages 1–7, New York, NY, USA. Association for Computing Machinery.

Andrei Broder. 2002. A taxonomy of web search. *ACM SIGIR Forum*, 36(2):3–10.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:15098–15106.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Brooke Cowan, Sven Zethelius, Brittany Luk, Teodora Baras, Prachi Ukarde, and Daodao Zhang. 2015. Named entity recognition in travel-related search queries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(2):3935–3941.

Hongbo Deng and Yi Chang. 2020. An introduction to query understanding. In Yi Chang and Hongbo Deng, editors, *Query Understanding for Search Engines*, pages 1–13. Springer International Publishing.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Homa B. Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In *International conference on web search and data mining, workshop on query understanding*.

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)*, pages 414–419. IEEE.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

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)*, pages 107–111, Berlin, Germany. Association for Computational Linguistics.

Vijay Krishnan and Vignesh Ganapathy. 2005. Named entity recognition. *Stanford Lecture CS229*.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Ying Li, Zijian Zheng, and Honghua (Kathy) Dai. 2005. Kdd cup-2005 report: Facing a great challenge. *ACM SIGKDD Explorations Newsletter*, 7(2):91–99.

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. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'20)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zheng Liu, Wei Zhang, Yan Chen, Weiyi Sun, Tianchuan Du, and Benjamin Schroeder. 2022. Towards generalizeable semantic product search by text similarity pre-training on search click logs. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 224–233, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Chen Luo, William Headden, Neela Avudaiappan, Haoming Jiang, Tianyu Cao, Qingyu Yin, Yifan Gao, Zheng Li, Rahul Goutam, Haiyang Zhang, and Bing Yin. 2022. Query attribute recommendation at amazon search. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 506–508, New York, NY, USA. Association for Computing Machinery.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

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*, pages 1557–1567, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282.

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29:3567–3575.

Daniel E. Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on*

World Wide Web, WWW '04, pages 13–19, New York, NY, USA. Association for Computing Machinery.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2022. Simple and effective knowledge-driven query expansion for QA-based product attribute extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 227–234, Dublin, Ireland. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Manos Tsagkias, Tracy Holloway King, Surya Kallumadi, Vanessa Murdock, and Maarten de Rijke. 2021. Challenges and research opportunities in eCommerce search and recommendations. *ACM SIGIR Forum*, 54(1):1–23.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes release 5.0 LDC2013T19. *Linguistic Data Consortium, Philadelphia, PA, USA*.

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*, BDCAT '19, page 149–154, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chao-Yuan Wu, Amr Ahmed, Gowtham Ramani Kumar, and Ritendra Datta. 2017. Predicting latent structured intents from shopping queries. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1133–1141, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Huimin Xu, Wenting Wang, Xin 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*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, and 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 & Knowledge Management*, CIKM '21, page 4362–4372, New York, NY, USA. Association for Computing Machinery.

The Seventh Workshop on e-Commerce and NLP (ECNLP 7), pages 125–133
21 May, 2024.