A Two-Phased Approach to Training Data Generation for Shopping Query Intent Prediction

Gautam Kumar[1,†], Chikara Hashimoto[1,†]

[1]Rakuten Institute of Technology (RIT), Rakuten Group Inc., 1-chōme-14 Tamagawa, Setagaya City, Tokyo, Japan 158-0094

Abstract

Shopping Query Intent Prediction (SQIP) is, given an online shopping user's search query, e.g., "*lv bag*", to predict their intents, e.g., Brand: Louis Vuitton. SQIP is an extreme multi-label classification task for which many excellent algorithms have been developed. However, little attention has been paid to how to create training data for SQIP. Previous studies used pseudo-labeled data derived from query-click logs for training and suffered from the noise in the logs. Although there are more sophisticated training data generation methods, they cannot be directly applied to SQIP. In this paper, we propose a novel training data generation method for SQIP. The idea is to first build a *labeling model* that checks whether an intent is valid for a query. The model then works as an "annotator" who checks a number of pairs comprising an intent and a query to generate training data for SQIP. We show that such a model can be trained without manual supervision by utilizing a huge amount of online shopping data. We demonstrate that the SQIP model trained with data generated by our labeling model outperforms a model trained with query-click logs only and a model trained with data created by a competitive data-programming-based method.

Keywords

training data generation, data-centric ai, shopping query intent, text classification, query attribute value extraction, online shopping, e-commerce query intent

1. Introduction

Online shoppers use search queries to search for products, and most queries have search intents that indicate what products shoppers want. For example, the query "*lv bag zebra*" has Brand: Louis Vuitton and Pattern: Zebra as its intents, as shown in Table 1.[1]

In this study, we assume that queries' intents are represented with attribute values of products defined in an online shopping service. Notice that simple string matching between queries and intents would not work since queries are written in natural languages; they can be represented with abbreviations, e.g., "*lv*" for "*Louis Vuitton*", and ambiguous words, e.g., "*orange*", as indicated in Table 1. Moreover, intents might not always be explicitly written in queries, as the last example in the table illustrates.

These intents, once correctly predicted, would be utilized by a search system to retrieve relevant products, since most products sold at an online shopping service have attribute values such as Brand: Louis Vuitton. If we aggregate these intents in bulk, they will be very useful in understanding trend of different attributes e.g. shoes of which brand and color the users wanted the most in last month. Also, they will be very helpful in understanding the overall market demand which could help the merchants and the manufacturing companies.

Table 1
Examples of queries and their intents

Query	Intents
"*lv bag zebra*"	Brand: Louis Vuitton Pattern: Zebra
"*100% orange juice*"	Fruit taste: Orange
"*cologne orange blossom*"	Scent: Orange
"*sneaker mens orange*"	Color: Orange
"*wheel 19inch*"	Tire size: 18 - 19.9inch
"*nicole down jacket*"	Brand: Nicole Filling: Feather

Shopping query intent prediction (SQIP), given a query, predicts its intents by selecting the most relevant subset of attribute values from the attribute value inventory defined in an online shopping service. In other words, SQIP gives a natural language query a structure to facilitate the retrieval of products.

In brief, our proposed method has following two phases:

1. **Making of Labeling Model:** Our labeling model is a binary classification model which predicts whether given a (query, intent) pair is valid or

DL4SR'22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

[†] These authors contributed equally.

gautam.kumar@rakuten.com (G. Kumar);
chikara.hashimoto@rakuten.com (C. Hashimoto)
https://chikarahashimoto.wixsite.com/home (C. Hashimoto)

[1]Intents are represented in the form of Attribute-name: Attribute-value in this paper. We also represent attribute values of products in a similar way.

not? For this we generate good quality training data and train a BERT Sequence Classification model. For the data generation, we follow following steps:

   a) Create Base SQIP model trained on product catalog data with input "product title" (could be considered as long pseudo shopping query) and output "attribute values" (could be considered as the pseudo shopping query's intents)
   b) Generate (query, intent) pairs by getting intents of queries from Query Click Logs using the Base SQIP model and take intersection with (query, intent) pairs from Query Click Logs

2. **Training Data Generation for SQIP:** From raw queries, get intents using the Base SQIP model and filter these intents using the labeling model.

The contributions of this paper are following:

1. We present a novel two-phased approach to training data generation for SQIP that requires no manual supervision.
2. We present how to build the labeling model, the key module of our two-phased approach, by combining weak supervision signals readily available in online shopping services.
3. We empirically demonstrate that our two-phased approach is effective through large-scale experiments.

1.1. Background

SQIP is an extreme multi-label text classification task for which many excellent algorithms have been developed recently [1, 2, 3, 4, 5, 6, 7, 8]. These classification algorithms can be used for SQIP once high-quality training data is available.

However, obtaining high-quality training data for SQIP is not straightforward. First of all, manual creation of a sufficient volume of training data would be infeasible because there are tens of thousands of predefined intents and understanding shopping query intents would require deep knowledge of a large number of product domains. Accordingly, previous studies [9, 10] used query-click logs to automatically generate training data by assuming that if a product has an attribute-value like Brand: Louis Vuitton and the page of the product is clicked by a user who issued a query like "*lv bag zebra*," an intent of the query is Brand: Louis Vuitton. This heuristic suffers from the inherent noise in query-click logs due to, for instance, inconsistent click behaviors of fickle users or erroneous retrieval results. Besides, it cannot utilize a number of queries that are absent in query-click logs. Despite the notable difficulty of obtaining high-quality training data, little attention has been paid to the problem in previous SQIP studies. Due to the success of pre-trained models [11], transfer learning has also been popular recently [12], where pre-trained models can be seen as providing weak supervision. With this approach, one fine-tunes a model that has been trained on a relevant task for the purpose of the target task using a reasonable amount of quality training data, which we cannot expect in SQIP.

There have also been many studies on combining weak supervision signals to dispense with manually annotated training data [13, 14, 15, 16, 17, 18], which would be useful if we may devise more than one kind of weak supervision signal for a given task. For SQIP, however, it would be infeasible to assume that labeling functions [14, 15, 17] or keywords [16, 18] for target classes can be frequently applied to or matched against queries since queries are usually very short and diverse. It would also be infeasible to prepare labeling functions or keywords for each class since the number of classes in SQIP amounts to tens of thousands and also the classes can be changed over time.

Automatically correcting corrupted labels has also gained much attention recently [19, 20, 21]. These methods learn label corruption matrices, which would be prohibitively large in SQIP since it has to deal with tens of thousands of classes.

1.2. Preview of the Proposed Method

What makes training data generation for SQIP difficult? We think it is the large number of classes; considering many classes for a query at once tends to be difficult. We therefore propose to decompose the task into two phases. In the first phase, we build a *labeling model* that checks whether an intent is valid for a query. In the second phase, we use this labeling model to verify each pair comprising a query and an intent on a large scale. Here, the labeling model can be seen as an annotator who is asked to create training data for SQIP. Refer to figures 1 and 2 for more details.

How can we build the labeling model? We propose to utilize catalog data and query-click logs since they are readily available in online shopping services and provide weak but different supervision signals so that they would reinforce each other, as we will demonstrate in Section 4.

Base SQIP model is a weak SQIP model that takes queries as input and predicts their intents, from which we generate a set of (query, intent) pairs. The base SQIP model is trained with catalog data, the database of products sold at an online shopping service, where various information about products such as product titles and their attribute values are registered. Product titles are usually a set of words that describe the features of products such as "*Louis Vuitton Shoulder bag Leather Zebra print*," which

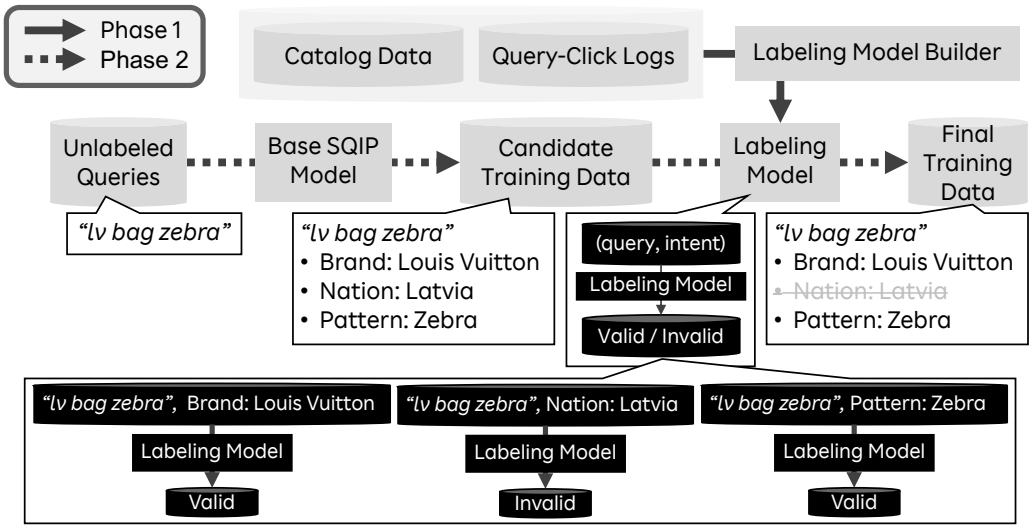

Figure 1: Overview of our training data generation method. In first phase we build the labeling model, which is depicted in Figure 2 in detail. In second phase, we generate candidate training data from unlabeled queries by using the base SQIP model. Afterwards, the labeling model filters out invalid (query, intent) pairs to generate the final training data.

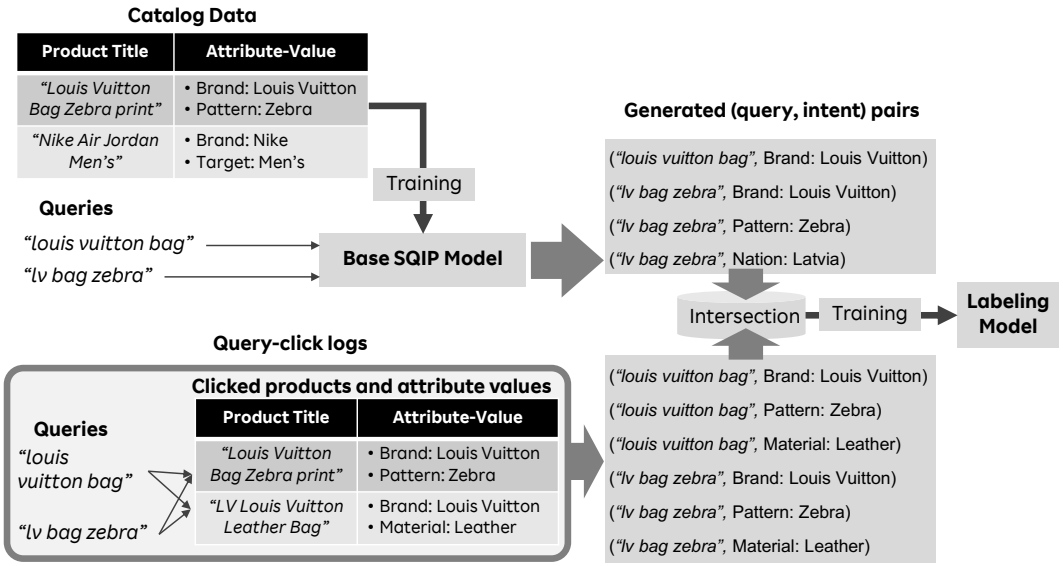

Figure 2: Closer look at the labeling model builder. Training data for the labeling model is the intersection of two sets of pairs comprising a query and an intent. Each set is generated by one of two weak generators; the base SQIP model and query-click logs.

can be seen as lengthy, detailed, merchant-made pseudo queries about the products. Since these titles (i.e., pseudo queries) are associated with attribute values of products (i.e., intents) we can use the catalog data to train the base SQIP model without manual annotation.

Query-click logs indicate the association between queries and clicked item's product attribute values (i.e. intents). We generate another set of (query, intent) pairs based on this association.

Catalog data provides the direct evidence of the association between product titles and attribute values (intents), but the titles are not real queries. In contrast, click logs

show the association between real queries and intents, but it is only indicated indirectly and tends to be noisy. However, these two data sources can generate reliable training data for the labeling model in tandem.

In summary, our proposed method creates a "machine annotator" namely, the labeling model, using huge amount of online shopping data to generate training data for SQIP on a large scale without requiring any manual labor.

Through large-scale SQIP experiments, we demonstrate that the model trained with data generated by our proposed method outperforms a model trained with query-click logs only and a model trained with data created by a competitive training data generation method based on data programming [14].

All the data used in this study were obtained from an online shopping service, Rakuten, and written in Japanese. However, the ideas and methods in this paper are independent of particular languages, and examples in this paper are written in English for ease of explanation.

2. Related Work

2.1. Shopping Query Intent Prediction

Previous methods for SQIP can be categorized into classification-based methods [9, 10] and sequence-labeling-based methods [22].

In this study, our proposed method generates training data for the classification-based methods for the following two reasons: First, with sequence-labeling-based methods, it would be more difficult to deal with tens of thousands of classes, while, for classification-based methods, there have recently been many excellent extreme classification algorithms that can handle a huge number of classes. Second, sequence-labeling-based methods deal with only intents that are explicitly written in queries. However, valid intents are not always explicit in queries; e.g., "*nicole down jacket*" has FILLING: FEATHER as its valid intent.

Our study is different from previous ones because we focus on how to obtain a huge volume of high-quality training data for SQIP, rather than how to classify queries. Previous studies simply used query-click logs to obtain pseudo-labeled data [9, 10], which tends to be noisy and unreliable. We will demonstrate that our proposed method can generate better training data in Section 4.

2.2. Learning with Weak Supervision

Our study can be seen as answering the research question of how to train supervised models without relying on manual annotation, and therefore studies on learning with weak supervision are quite relevant. As we discussed in Section 1, most of the previous weak-supervision methods are not appropriate for SQIP since they require external knowledge bases [23, 24], a reasonable amount of quality training data [12], labeling functions or keywords for target classes [14, 15, 16, 17, 18], or label corruption matrices to be learned [19, 20, 21]. Shen et al. proposed learning classifiers with only class names [25]. However, their method assumes that classes are organized in a hierarchy, so we cannot use their method for SQIP where classes (intents) are not organized in a hierarchy. Karamanolakis et al. [26] proposed a method that works with weak supervision such as lexicons, regular expressions, and knowledge bases of the target domain. However, such weak supervision would become obsolete quickly in SQIP, as discussed in Section 1. Zhang et al. [27] proposed a teacher-student network method which utilizes weakly labeled behaviour data for SQIP. However, they do use strongly labeled data in their training methodology to train the teacher network.

2.3. Extreme Multi-Label Classification

SQIP is an extreme multi-label classification (XML), which tags a data point with the most relevant subset of labels from an extremely large label set, that has gained much attention recently [1, 2, 3, 7, 8]. While many classification algorithms have been proposed, training data generation for XML has not been well studied. Zhang et al. [28] addressed data augmentation for XML, which assumed the existence of training data and thus cannot be applied to our setting. Our study therefore differs from previous XML studies since we directly tackle the task of training data generation, though our method is specifically designed for SQIP.

For a more comprehensive overview of classification algorithms and data sets for XML, refer to http://manikvarma.org/downloads/XC/XMLRepository.html.

3. Proposed Method

In this section, we describe each component of our method as illustrated in Figures 1 and 2; catalog data, the base SQIP model, query-click logs, the labeling model, unlabeled queries, candidate training data, and the final training data.

3.1. Catalog Data

Catalog data contains various information of products sold at the shopping service, including product titles, descriptions, prices, various attribute values such as brands, sizes, and colors, among others. We use product titles and attribute values to train the base SQIP model, since product titles are usually a set of words that indicate

Table 2
Examples of product titles and attribute values

Product title	Attribute values
"*[Next-day delivery] Nike Women's Zoom Vaper 9.5 Tour 631475-602 Lady's Shoes*"	Brand: Nike, Color: Red
"*TIFFANY&CO. tiffany envelope charm [NEW] SILVER 270000487012x*"	Brand: Tiffany & Co., Color: Silver
"*[Kids clothes/STUSSY] Classic Logo Strapback Cap black a118a*"	Clothing fabric: Cotton
"*Fitty Closely-attached mask Pleated type Slightly small size Five-pack*"	Mask shape: Pleated
"*[Unused] iQOS 2.4PLUS IQOS White Electric cigarette Main body 58KK0100180*"	Color: White
"*NIKE AIR MAX 90 ESSENTIAL Sneaker Men's 537384-090 Black [In-stock, May 15]*"	Shoe upper material: Leather, Brand: Nike

the features of products and can consequently be seen as lengthy, detailed queries about the products. Table 2 shows examples of product titles and their attribute values in our catalog data, and indicates differences between product titles and real queries. First, product titles sometimes contain tokens that would not appear in queries usually, such as "*[Unused]*" and "*[In-stock, May 15]*." Second, real queries are usually much shorter than product titles. Third, attribute values might not always mean intent. For example, color: red is not intent if we consider product title as shopping query in first example of table 2. Catalog data is a useful data source for training a SQIP model but is not sufficiently reliable by itself due to these differences.

To train the base SQIP model, we used 117 million product titles and their associated attribute values. The number of different attribute values was 19,416.

3.2. Base SQIP Model

The base SQIP model takes unlabeled queries, such as "*lv bag zebra*" as input and predicts their intents such as Brand: Louis Vuitton and Pattern: Zebra. We had to deal with hundreds of millions of training instances in our experiments (Section 4) and chose extremeText [29]. It was the only extreme multi-label classification method that we experimented with that could handle all training instances in our environment. Other extreme multi-label classification methods we experimented with include Parabel [1], Bonsai [2], LightXML [7], XR-Linear [30], and XR-Transformer [8].

The classification algorithm of extremeText is based on probabilistic label trees (PLT) [31], in which leaf nodes represent the target labels and the other nodes are logistic regression classifiers. PLT guides data points from the root node into their appropriate leaf nodes (labels) with the logistic regression classifiers. For training the model, we did not conduct extensive hyper-parameter tuning; we used its default hyper-parameters, except that we chose PLT as the loss function, and used the TF-IDF weights for words.

3.3. Query-Click Logs

We used one year of query-click logs, which contained 72 million unique queries. As illustrated in Figure 2, the query-click logs are used to generate (query, intent) pairs as part of training data for the labeling model. We simply enumerated all possible (query, intent) pairs such that a query is associated with an intent (attribute-value) via click relations in the logs.

3.4. Labeling Model

The labeling model takes a pair comprising a query (e.g., "*lv bag zebra*") and an intent (e.g., Brand: Louis Vuitton) as input and predicts whether the intent is valid for the query.

3.4.1. Model Architecture

BERT[11]-based models have been very promising for text pair classification and regression tasks, such as natural language inference (NLI) [32] and semantic textual similarity (STS) [33]. Since the task of the labeling model is binary classification, we used BertForSequenceClassification[2] where we use pretrained BERT model for Japanese[3].

We intentionally adopted a very simple approach so that we could demonstrate the effectiveness of our method.

3.4.2. Training Data

As shown in Figure 2, we automatically generate training data for the labeling model which is the intersection of two sets of (query, intent) pairs; one set is generated with the base SQIP model[4] and another is from the query-click logs. Although each of these two kinds of supervision signals is weak by itself, we can accurately obtain a number of valid (query, intent) pairs by combining them.

[2]https://huggingface.co/transformers/model_doc/bert.html.
[3]https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking
[4]The input to the base SQIP model is the queries in the query click logs.

To be specific, we obtained (query, intent) pairs such that the query is associated with the intent in the query-click logs and also, given the query as input, the base SQIP model predicted the intent with probability 1.0. As a result, we generated 5.3 million (query, intent) pairs as positive examples for training of the labeling model. We then generated 5.3 million (query, intent) pairs by randomly pairing queries and intents, which we used as negative examples.

3.4.3. Training Detail

The labeling model has been built with the training data and the model architecture, as described above. Training is done for one epoch with batch size 32 using AdamW [34] optimizer.

3.5. Unlabeled Queries, Candidate Training Data, and Final Training Data

The second phase starts with predicting intents for unlabeled queries from query logs with the base SQIP model to generate candidate training data. We then filter out erroneous intents with the labeling model to generate the final training data.

Unlabeled queries were obtained from seven years of query logs, which contained more than 1.5 billion unique queries.

Candidate training data were generated under the following condition: k=5, meaning that the base SQIP model predicted the most probable five intents for a query at most, and threshold=1.0, i.e., only those intents whose probability was 1.0 were outputted. As a result, we obtained 377 million (query, intent) pairs. The number of unique queries was 264 million.

The final training data were those (query, intent) pairs whose probability given by the labeling model was at least 0.99. Consequently, we obtained 169 million (query, intent) pairs. The number of unique queries was 145 million. We trained and evaluated the SQIP model with this final training data, as reported in Section 4. Table 3 shows examples of the final training data.

Table 3
Examples of the final training data

Query	Intents
"*alpha ma-1*"	Brand: Alpha Industries
"*orange t-shirt*"	Color: Orange
"*tropicana orange*"	Fruit taste: Orange Series: Tropicana
"*gres perfume orange*"	Scent: Orange, Brand: Gres
"*washbowl 750*"	Capacity: 600 - 899ml
"*original message carnation*"	Event/Holiday: Mother's Day

4. Experiments

In this section, through large-scale SQIP experiments in which one predicts intents of a given query, we claim the following:

1. Simply using query-click logs for training SQIP models delivers poor performance.
2. Using catalog data for training leads to better performance than simply using query-click logs but is still unsatisfactory.
3. Our proposed method that exploits both catalog data and query-click logs can generate even better training data.
4. Without the labeling model, the performance of our method degrades, indicating the effectiveness of the labeling model.
5. Our proposed method outperforms the competitive training data generation method based on data programming called *Snorkel* [14].

4.1. Experimental Conditions

In our experiments, we compared our proposed method with four baseline methods described in Section 4.2. All the compared methods differ only in how they obtain training data. For classification, they use the same architecture, extremeText; specifically, all the methods trained their SQIP model with the PLT loss function and the TF-IDF weights for words; the other hyper-parameters were set to the default values.

Test data has been manually created by a human annotator (who is not an author). The annotator was asked to check (query, intent) pairs that were automatically generated by pairing a query and an intent, such that at least one token in the query was semantically similar or relevant to the intent in order to exclude obviously erroneous (query, intent) pairs from all possible pairs in advance of manual annotation.[5] As a result, 5,615 different queries with at least one intent were obtained as test data, and 2.57 intents were given to a query, on average.

Evaluation was based on precision and recall, which were calculated with extremeText's test command. Precision and recall were calculated for top k outputs (i.e., intents) with k being 1, 3, and 5, and we drew precision-recall curves for the compared methods for each k with the probability threshold of extremeText changing from 0.0 to 1.0 with the interval of 0.01.

[5]The semantic similarity was measured by the cosine similarity between their sentence embeddings. We use fastText embeddings [35], which had been learned from the query logs. The threshold for the cosine similarity was set to 0.8.

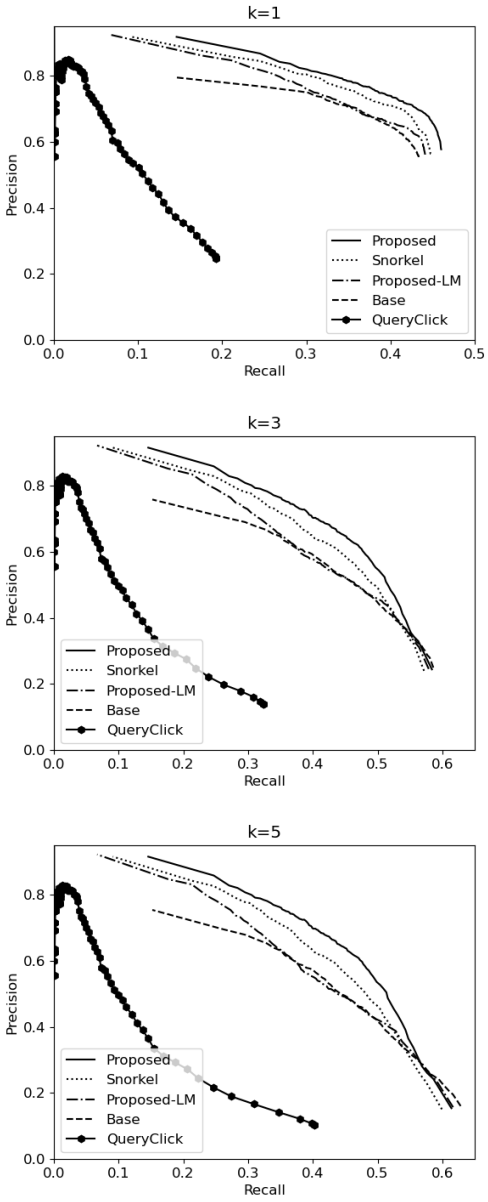

Figure 3: Precision-recall curves for all Experiments

4.2. Compared Methods

We compared the following five methods:

4.2.1. QueryClick

The simplest baseline is `QueryClick`, which uses query-click logs to generate training data in a similar way to the previous methods [9, 10]. Specifically, we used seven years of query-click logs and obtained (query, intent) pairs in which product pages that had the intent (i.e., attribute value) were clicked through the query at least ten times in the logs. The purpose of this was to reduce the inherent noise in the query-click logs. As a result, we obtained more than 670 million (query, intent) pairs. The number of unique queries was 7,962,605, which indicated that each query was given as many as 84 intents on average. This number is obviously too large given that most queries consist of less than ten tokens and supports our claim that simply using query-click logs as training data would be inadequate.

4.2.2. Base

This is the base SQIP model, which uses only product titles and their associated attribute values for training.

4.2.3. Proposed

This is a SQIP model trained with the final training data generated with our proposed method, as described in Section 3.

4.2.4. Proposed-LM

This is the same as `Proposed` except that it does not use the labeling model. `Proposed-LM` is then trained with the candidate training data in the second phase; its training process is similar to self-training. Note that the difference in performances between `Proposed-LM` and `Proposed` can be seen as indicating the effectiveness of the labeling model.

4.2.5. Snorkel

This baseline is the same as `Proposed`, except that the labeling model is replaced with Snorkel [14], a training data generation method based on data programming [13]. Like `Proposed`'s labeling model, Snorkel's labeling model can be learned without manual supervision. However, Snorkel requires *labeling functions* that implement a variety of domain knowledge, heuristics, and any kind of weak supervision that would be useful for a given task. Each labeling function takes unlabeled data points as input and predicts their class labels. Snorkel then uses these weakly-labeled data points to train a generative labeling model which is supposed to be able to label each data point more accurately than the labeling functions. Snorkel has influenced subsequent studies on training data generation [17], and has also been adopted by the world's leading organizations as described in https://www.snorkel.org/. We therefore think that comparing with the Snorkel-based baseline would effectively show `Proposed`'s performance.

Table 4
Proposed's best F1 scores

k	F1	Precision	Recall	Threshold
1	0.537	0.678	0.444	0.21
3	0.535	0.620	0.470	0.26
5	0.531	0.608	0.471	0.26

Our Snorkel baseline, to be specific, was implemented in the following way: The input and output of Snorkel's labeling model are the same as Proposed's labeling model; the input (query, intent) pairs are generated with the base SQIP model; the output is whether given (query, intent) pairs are valid or not. We defined following labeling functions that utilized the same two kinds of weak supervision as Proposed, i.e., the query-click logs and the base SQIP model which are following:

1. If given intent is associated with the given query in query-click logs, return *valid*; otherwise return *invalid*.
2. If output probability of the base SQIP model, given (query, intent) pair is 1.0, return *valid*; otherwise abstain.
3. Return *invalid* if the output probability is not greater than 0.995; otherwise abstain.

Snorkel's labeling model was trained with 11 million (query, intent) pairs that had been weakly-labeled with the three labeling functions.

Proposed's labeling model was trained with 10.6 million (query, intent) pairs as described in Section 3.4.2.

4.3. Results

Figure 3 shows precision-recall curves for the compared methods and from them we can make the following observations:

1. QueryClick's precision decreases sharply as we try to increase recall.
2. Base generally outperforms QueryClick, though its performance is still unsatisfactory.
3. Proposed outperforms all the other methods. Table 4 shows Proposed's best F1 scores and their corresponding precision, recall, and threshold values for each k.
4. Proposed-LM's performance is worse than that of Proposed.
5. Snorkel can deliver good performances but cannot outperform Proposed.

The relatively low performances of QueryClick and Base and the relatively high performances of Proposed and Snorkel indicate that query-click logs and catalog data alone can only provide weak supervision. However, combining them can lead to higher performances.

Comparing Proposed with Snorkel shows the superiority of our labeling model over Snorkel. We think this is because labeling functions of Snorkel or learning methods with weak heuristic rules in general have been known to suffer from *a low coverage* [26]; rules tend to be applied to only a small subset of instances. In fact, the first labeling function for Snorkel covered only 1.94% of the training instances. The second and third labeling functions covered 60.61% and 12.73%, respectively. On the other hand, the labeling model of Proposed is learned with natural language words and phrases, which BERT makes the maximum use of; that is to say, the labeling model of Proposed does not waste the training data.

4.4. Error Analysis

Table 5 illustrates examples of wrong prediction made by Proposed (k=1, threshold=0.21). Most of the errors were due to the *class imbalance* in the training data; i.e., the distribution of training instances across the intents is biased or skewed, and intents for which we have few or no instances tend to be difficult to predict [36]. Regarding the first example in Table 5, "*prince*" can be Brand: Prince and be part of Brand: Glen Prince. However, the frequency of the former intent in the final training data was 119,972, whereas that of the latter was only 33, which caused the SQIP model to choose the former for the query. Regarding the second example, the frequency of Color: Red was 1,486,315, while that of Brand: Red Wing was 15,592. For the last one, there was no training instance for Memory Standards: DDR3 in our final training data and thus, the SQIP model could not predict it.

4.5. Effect of Training Data Size

Figure 4 shows F1 scores of Proposed built with the final training data of different sizes ('K' and 'M' stand for 'thousand' and 'million'). The k and the threshold of extremeText were set to 1 and 0.21 uniformly. The graph indicates that increasing the data size leads to better performances and that our final training data is effective for SQIP. Although the improvement from 10M to 145M is small, it is noteworthy that additional data could improve the model trained already with as many as 10M instances.

5. Future Direction

For training data generation, one possible direction is to use product genre/category information. If we could

Table 5
Examples of wrong prediction made by Proposed

Query	True Intents	Predicted Intents
"*glen prince*"	Brand: Glen Prince	Brand: Prince
"*red wing engineer boots us 7.5*"	Brand: Red Wing	Color: Red
"*pc 3 12800 ddr 3 sdram*"	Memory Standards: DDR3	–

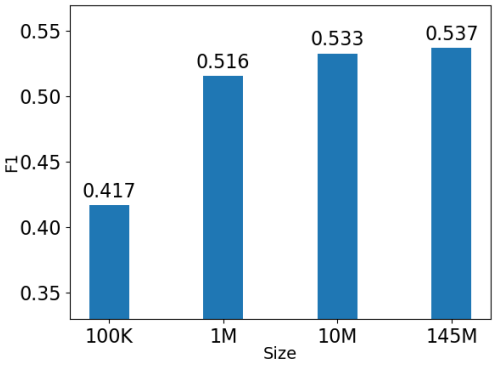

Figure 4: Changes in F1 due to different training data sizes for proposed

create query to product genre mapping of reliable quality, we can filter (query, intent) pairs further and create higher quality training data. Also, we could utilize neighbor signals, since similar queries should have more labels in common, to remove noise from the dataset further.

For the classification model, one possibility is to use label (i.e. intent) context information to create embedding vector of input text (i.e. shopping query). Similar previous work is by Chen et al. [37] who uses LGuidedLearn [38] for Product Item Category Classification. Another possible method could be Label-Specific Document Representation for Multi-Label Text Classification by Xiao et al. [39]. Also, Cai et al. [40] proposes a hybrid neural network model to simultaneously take advantage of both label semantics and fine-grained text information. Another possibilities are to consider Contrastive Learning and KNN based methods [41, 42].

Another direction is to extend our proposed method in other domains. If we could find a way to exploit weak supervision signals readily available in a domain for building the labeling model, we can easily apply our approach to the domain. In the case of text classification into Wikipedia categories [43], for instance, not only the category information in Wikipedia articles but also the links among corresponding articles in different languages and the class hierarchy in Wikidata [44] can be exploited.

As we have seen in section 4.4 that data imbalance is an issue, in the future we aim to address this.

6. Conclusion

In this paper, we proposed the novel two-phased training data generation method for SQIP. The idea is to first build a labeling model that checks whether an intent is valid for a query. The model then works as an "annotator" who checks a number of pairs comprising an intent and a query to generate training data for SQIP. We presented how to train such a model without manual supervision by utilizing a huge amount of online shopping data. Through the series of large-scale experiments with the data from a real online shopping service, we have demonstrated the effectiveness of our proposed method.

Acknowledgments

We thank our annotator Saki Hiraga-san for helping us in creation of evaluation dataset. We thank all the researchers in RIT for their support for this project.

References

[1] Y. Prabhu, A. Kag, S. Harsola, R. Agrawal, M. Varma, Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising, in: Proceedings of the 2018 World Wide Web Conference, WWW '18, 2018, p. 993–1002.

[2] S. Khandagale, H. Xiao, R. Babbar, Bonsai – diverse and shallow trees for extreme multi-label classification, 2019. arXiv:1904.08249.

[3] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, I. S. Dhillon, Taming Pretrained Transformers for Extreme Multi-Label Text Classification, KDD '20, 2020, p. 3163–3171.

[4] K. Dahiya, D. Saini, A. Mittal, A. Shaw, K. Dave, A. Soni, H. Jain, S. Agarwal, M. Varma, Deepxml: A deep extreme multi-label learning framework applied to short text documents, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, 2021, p. 31–39.

[5] A. Mittal, K. Dahiya, S. Agrawal, D. Saini, S. Agarwal, P. Kar, M. Varma, Decaf: Deep extreme classi-

fication with label features, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, 2021, p. 49–57.

[6] A. Mittal, N. Sachdeva, S. Agrawal, S. Agarwal, P. Kar, M. Varma, Eclare: Extreme classification with label graph correlations, in: Proceedings of the Web Conference 2021, WWW '21, 2021, p. 3721–3732.

[7] T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, F. Zhuang, Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, 2021, pp. 7987–7994.

[8] J. Zhang, W.-C. Chang, H.-F. Yu, I. S. Dhillon, Fast multi-resolution transformer fine-tuning for extreme multi-label text classification, in: 35th Conference on Neural Information Processing Systems, NeurIPS 2021, 2021.

[9] C. Wu, A. Ahmed, G. R. Kumar, R. Datta, Predicting latent structured intents from shopping queries, in: Proceedings of the 26th International Conference on World Wide Web, WWW '17, 2017, pp. 1133–1141.

[10] J. Zhao, H. Chen, D. Yin, A dynamic product-aware learning model for e-commerce query intent understanding, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, 2019, p. 1843–1852.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, NAACL-HLT '19, 2019, pp. 4171–4186.

[12] M. Ben Noach, Y. Goldberg, Transfer learning between related tasks using expected label proportions, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP '19, 2019, pp. 31–42.

[13] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, C. Ré, Data programming: Creating large training sets, quickly, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 29 of *NeurIPS '16*, 2016.

[14] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, Snorkel: Rapid training data creation with weak supervision, Proceedings of the VLDB Endowment 11 (2017) 269–282.

[15] B. Hancock, M. Bringmann, P. Varma, P. Liang, S. Wang, C. Ré, Training classifiers with natural language explanations, Proceedings of The 56th Annual Meeting of the Association for Computational Linguistics 2018 (2018) 1884–1895.

[16] Y. Meng, J. Shen, C. Zhang, J. Han, Weakly-supervised neural text classification, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, 2018, p. 983–992.

[17] A. Awasthi, S. Ghosh, R. Goyal, S. Sarawagi, Learning from rules generalizing labeled exemplars, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeuexBtDr.

[18] Y. Meng, Y. Zhang, J. Huang, C. Xiong, H. Ji, C. Zhang, J. Han, Text classification using label names only: A language model self-training approach, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP '20, 2020, pp. 9006–9017.

[19] G. Patrini, A. Rozza, A. K. Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: A loss correction approach, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '17, 2017, pp. 2233–2241.

[20] D. Hendrycks, M. Mazeika, D. Wilson, K. Gimpel, Using trusted data to train deep networks on labels corrupted by severe noise, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, 2018, p. 10477–10486.

[21] G. Zheng, A. H. Awadallah, S. Dumais, Meta label correction for noisy label learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35 of *AAAI '21*, 2021.

[22] X. Li, Y.-Y. Wang, A. Acero, Extracting structured information from user queries with semi-supervised conditional random fields, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, 2009, p. 572–579.

[23] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, 2009, p. 1003–1011.

[24] F. Brahman, V. Shwartz, R. Rudinger, Y. Choi, Learning to rationalize for nonmonotonic reasoning with distant supervision, in: The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI '21, AAAI Press, 2021, pp. 12592–12601.

[25] J. Shen, W. Qiu, Y. Meng, J. Shang, X. Ren, J. Han, TaxoClass: Hierarchical multi-label text classifica-

tion using only class names, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT '21, 2021, pp. 4239–4249.

[26] G. Karamanolakis, S. Mukherjee, G. Zheng, A. H. Awadallah, Self-training with weak supervision, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL '21, 2021, pp. 845–863.

[27] D. Zhang, Z. Li, T. Cao, C. Luo, T. Wu, H. Lu, Y. Song, B. Yin, T. Zhao, Q. Yang, 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, Association for Computing Machinery, New York, NY, USA, 2021, p. 4362–4372. URL: https://doi.org/10.1145/3459637.3481946. doi:10.1145/3459637.3481946.

[28] D. Zhang, T. Li, H. Zhang, B. Yin, On data augmentation for extreme multi-label classification, CoRR abs/2009.10778 (2020). URL: https://arxiv.org/abs/2009.10778.

[29] M. Wydmuch, K. Jasinska, M. Kuznetsov, R. Busa-Fekete, K. Dembczyński, A no-regret generalization of hierarchical softmax to extreme multi-label classification, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NeurIPS '18, 2018, p. 6358–6368.

[30] H.-F. Yu, K. Zhong, I. S. Dhillon, Pecos: Prediction for enormous and correlated output spaces, arXiv preprint arXiv:2010.05878 (2020).

[31] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, E. Hullermeier, Extreme f-measure maximization using sparse probability estimates, in: M. F. Balcan, K. Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, 2016, pp. 1435–1444.

[32] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, X. Zhou, Semantics-aware BERT for language understanding, in: the Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.

[33] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, L. Specia, SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 1–14.

[34] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR '19, 2019.

[35] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, arXiv preprint arXiv:1607.04606 (2016).

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority oversampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[37] L. Chen, H. Miyake, Label-guided learning for item categorization in e-commerce, in: NAACL, 2021.

[38] X. Liu, S. Wang, X. Zhang, X. You, J. Wu, D. Dou, Label-guided learning for text classification, 2020. URL: https://arxiv.org/abs/2002.10772. doi:10.48550/ARXIV.2002.10772.

[39] L. Xiao, X. Huang, B. Chen, L. Jing, Label-specific document representation for multi-label text classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 466–475. URL: https://aclanthology.org/D19-1044. doi:10.18653/v1/D19-1044.

[40] L. Cai, Y. Song, T. Liu, K. Zhang, A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification, IEEE Access 8 (2020) 152183–152192. doi:10.1109/ACCESS.2020.3017382.

[41] L. Zhu, H. Chen, C. Wei, W. Zhang, Enhanced representation with contrastive loss for long-tail query classification in e-commerce, in: Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 141–150. URL: https://aclanthology.org/2022.ecnlp-1.17. doi:10.18653/v1/2022.ecnlp-1.17.

[42] X. Su, R. Wang, X. Dai, Contrastive learning-enhanced nearest neighbor mechanism for multi-label text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 672–679. URL: https://aclanthology.org/2022.acl-short.75. doi:10.18653/v1/2022.acl-short.75.

[43] O. Dekel, O. Shamir, Multiclass-multilabel classification with more classes than examples, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, 2010, pp. 137–144.

[44] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledge base, Communications of the ACM 57 (2014) 78–85. URL: http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext.