FABRIC: Fully-Automated Broad Intent Categorization in E-commerce

Anna Tigunova
tigunova@amazon.de
Amazon
Berlin, Germany

Philipp Schmidt
phschmid@amazon.de
Amazon
Berlin, Germany

Ezgi Akcora
akcora@amazon.lu
Amazon
Luxembourg, Luxembourg

Abstract

Predicting the user's shopping intent is a crucial task in e-commerce. In particular, determining the product category, which the user wants to shop, is essential for delivering relevant search results and website navigation options.

Existing query classification models are reported to have excellent predictive performance on the single-intent queries (e.g. '*running shoes*'), but there is little research on predicting multiple-intents for broad queries (e.g. '*running gear*'). While training data for broad query classification can be easily obtained, *evaluation* of multi-label categorization remains challenging, as the set of true labels for multi-intent queries is subjective and ambiguous.

In this work we propose *FABRIC* – an automatic method of creating evaluation data for multi-label e-commerce query classification. We reduce the ambiguity of the annotations by blending the label assessment from three different sources: aggregated click data, query-item relevance predictions and LLM judgments.

1 Introduction

The query classification component in e-commerce stores has a crucial influence on the customer's shopping experience. For instance, the predicted query category can be used to automatically route the user to a relevant department, or show related search filters. Given the importance of this task, there has been a lot of research on improving the performance of query classifiers (Luo et al., 2022; Bonab et al., 2021; Tigunova et al., 2024).

E-commerce query classification research focuses on single-intent user queries, which are the queries with a well-defined category intent, such as '*men's shoes size 43*' or '*cheap wireless speakers*'. At the same time, broad-intent queries, such as '*gifts for my grandma*', are left unexplored. Yet, broad queries constitute a non-negligible fraction of search traffic and need to be handled by the query classification models as well.

To be able to correctly address category prediction for broad queries, it is necessary to create multi-label data for classification model training and evaluation. Large-scale training data of the query classification models is usually obtained by distant supervision from user behavior, with the assumption that the query can be classified into the categories that the users clicked most, following their search (Lin et al., 2018; Luo et al., 2022; Zhu et al., 2022). Broad query classification can be trained using such data as well, however, it cannot be applied for the model evaluation, because of the noise and bias in user behavior.

Existing manual and automatic datasets for evaluating query classification performance are single-labeled and cannot be used to assess the quality on broad queries. The challenge of creating a broad query evaluation dataset is that the selection of the true labels is often subjective and therefore challenging even for the human annotators. To the best of our knowledge, there is no existing multi-label evaluation dataset for broad query classification.

To close this gap, we propose *FABRIC* (**F**ully-**A**utomated **B**road **I**ntent **C**ategorization) – a scalable procedure for collecting broad query evaluation data. Our method probabilistically aggregates the query annotations from diverse sources to reduce the ambiguity and select high-confidence labels. We utilize the labeling methods based on the aggregated click data, query relevance predictions and world knowledge of an LLM, which assess broad query labeling from different perspectives.

Manual audit of resulting labeling shows high quality of the annotations, proving its applicability in evaluation of the critical production models. Our approach can be applied to any e-commerce service, regardless of its domain or language. The fully automated procedure is modular and extensible, and allows to regularly refresh the dataset.

In this paper, we showcase the proposed methodology on a controlled pilot data sample. However, we also applied our method for a large-scale evaluation of the proprietary query classifier, proving its practical utility for system diagnostics and monitoring in industrial applications.

2 Background

2.1 Multilabel queries in e-commerce

Search query classification in e-commerce extracts shopping intent from the customer queries, namely, which category of products the customer wanted to buy or browse. In large e-commerce stores such product categories are fine-grain and can span thousands of values. Usually these categories are mutually exclusive, and the majority of the queries can be unambiguously classified to one of them (e.g. the query '*gym sneakers*' can be classified as *shoes*). In contrast, in this work we focus on *broad* queries, spanning multiple item categories.

Such queries indicate that the customer is looking for inspiration, information and/or recommendation with very broad product intent, which spans multiple product categories. In this work, we consider four examples of such broad classes: gifts ('*presents for 2yo son*'), activity supplies ('*everything for knitting*'), accessories ('*car essentials*') and outfits ('*clothes for toddler*').

2.2 Related work

Human annotation is a standard option to create high quality evaluation data for classification models. However, in query classification domain, only few studies report the results on human-labeled evaluation sets (Zhang et al., 2021b,a), due to high costs of large-scale annotation and the difficulty to refresh this data in response to the emerging trends in e-commerce.

Therefore, the majority of the studies use aggregated click data to infer query classification labels, which are abundantly available and are regularly refreshed (Lin et al., 2018; Zhu et al., 2023, 2022). However, when such data is used for both model training and evaluation, data biases are propagated to the model, and are impossible to identify during evaluation. To overcome this, Tigunova et al. (2025) propose a method to automatically label query classification datasets using query-item relevance predictions instead of click data.

All previously proposed approaches are generally used for single-label query classification, while a more challenging multi-label setting has not been addressed. In our work we extend the single-label click-through and relevance-based approaches for the multi-category case, additionally combining them with the LLM knowledge to derive reliable labels for the subjective broad queries.

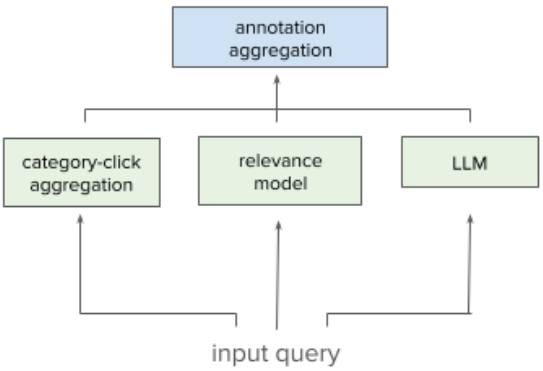

Figure 1: Schema of the proposed approach.

3 Methodology

In this section we describe the methodology to create the multi-intent query classification dataset. The process of creating the labeled query set can be split into 2 steps:

1. **Query selection** - involves selecting a representative list of multi-category queries for labeling. The goal is to have high volume and diversity of broad queries.

2. **Query labeling** - involves deriving high-quality annotations for the selected queries. For each multi-label query the annotations should be: i) precise - including only relevant categories, and ii) comprehensive - including all essential labels that should be included in the search results for this query.

In the following we describe these two steps.

3.1 Multi-intent query selection

We sample the candidate queries from aggregated historical click logs, pre-filtered to remove explicit single-category queries: the ones that mention the name of some product category or a product model name (found in the item catalog).

As the next step, we aim to select the queries relevant to our four category types: *gift*, *activity*, *accessories*, *outfits*. For *gift* type we selected queries mentioning the words 'present' or 'gift'; for *accessory* the queries containing words 'accessory'

activity	gift	accessory	outfits
ceramic cooking	gift 18 year old girl	truck dashboard accessory	cotton clothes for women
natural cleaning products	gifts for 12 month old	bridal party essentials	large female clothes
dog photo booth props	26 birthday gifts women	flower hair accessory set	clothes for girls 10-12
summer camp party	flower themed gifts	swimming pool accessories	toddler outfits boy

Table 1: Examples of queries per multi-category topic.

or 'essentials'; and for the *outfit* type the queries containing terms 'clothes' or 'outfit'. To detect queries from *activity* topic we used an NER model to identify the spans in the queries classified as an activity (e.g. 'hiking' or 'bachelor party').

Finally, from each category we select the top N queries by the click entropy. Click entropy is computed from the aggregated click logs using the frequencies of clicks on items: if the majority of the clicks is concentrated on a particular set of items, then it is less probable that the given query is broad.

The four selected categories are a demonstrative sample from a much larger set we applied internally (see Section 4.3). We selected them because they represent some of the most common and challenging broad-query categories in e-commerce, and are sufficient to reveal whether downstream classifiers can handle multi-intent queries.

3.2 Query annotation

When selecting a query annotation approach it is important to consider not only the quality, but also the cost and effort of annotation. For this reason human-labeling is out of scope of this work, as it is expensive, time-consuming and it does not guarantee to provide high quality annotations. Indeed, categorizing broad queries is a highly subjective task, leading to poor inter-annotator agreement. Moreover, it is extremely difficult for most humans to effectively distinguish among thousands of product categories for a given search query. Therefore, we consider only automated labeling approaches.

To build our search query annotation pipeline, we carefully chose three independent sources that provide multi-label predictions. One of which relies entirely on external knowledge:

1. **Aggregated click-through logs** - following related studies (Qiu et al., 2022; Zhu et al., 2023, 2022), for each candidate query we select the top clicked product categories. We only accept product categories if they exceed a predetermined click frequency threshold.
2. **Relevance labeling** - is a multi-label extension of the methodology of Tigunova et al. (2025), where the query categories are derived from the categories of the relevant search results for this query.
3. **LLM annotation** - we prompt an LLM to produce categories for each selected query.

We show a schematic overview of our method in Figure 1. The details on each of these methods are provided below.

Aggregated click-through logs. For this method we use aggregated historical click logs. For each candidate query q and product category c_k, the score $s(q, c_k)$ is derived as the fraction of items from category k, that the customers clicked following their search with query q, normalized by all clicks: $s(q, c_k) = \frac{clicks_{q,c_k}}{clicks_q}$. In the annotation for query q we include only the categories with the $s(q, c_k)$ score, surpassing a threshold $t1$, which we treat as a hyperparameter of our pipeline.

Using aggregated click-through logs is among the most established methods of collecting query classification annotations at scale. This approach reflects customer preferences, however, it can provide imprecise labels in certain cases. First, the derived labels are susceptible to noise, trends, and seasonality. For instance, in Table 1, for the query '*comfy male clothes*' in winter time the users will click more on *sweater* products and in summer time on *tunic*. Second, the click signal suffers from exposure bias, i.e., clicks are mostly obtained mostly on the items shown on top of the search page. For example, the click signal for query '*home spa gift basket*' expose only categories *skin care agent* and *bathwater additive*, potentially missing on other relevant categories.

Relevance labeling. To overcome the limitations of using aggregated click-through logs, we utilize a relevance labeling method (Tigunova et al., 2025), which derives query-category mappings from the *relevant* products shown on the search result page.

Unlike (Tigunova et al., 2025) to determine if a product is relevant, our relevance labeler relies on a proprietary classifier, fine-tuned on large-scale human-labeled query–product pairs. Importantly, this classifier is not trained on click-based signal.

We derive query-category associations from top-100 products shown on the search result page. For each query we select the items, predicted as relevant, and collect the product categories associated with them. As a post-processing step, for each query we remove the categories that have less than a threshold of $t2$ associated relevant items. We also treat $t2$ as a hyperparameter of our pipeline.

This approach mitigates the problems of using aggregated click-through logs as a source to label queries, as it is independent of what customers choose to click on. However, the reliability of annotations depend on the quality of the relevance model. Especially for broad queries with ambiguous categorization, we noticed that the relevance model can get confused and tends to assign irrelevant categories (which technically can still be vaguely applicable to the query). For instance, for the query '*home spa gift basket*' in Table 1, this approach predicts a category *cookie*, which can be used as a gift, but is not directly related to spa.

Relevance labeling can inherit biases from the underlying item retrieval system because it is based on products surfaced on the search page. This bias is indirect and substantially different from click bias: relevance model does not rely on user actions but instead on a learned classifier with broad background knowledge from manually annotated data.

LLM annotation. To complement the two previously described systems that we use to predict query categories, we have developed a solution based entirely on external knowledge, i.e., a method based on an LLM. This label source expands the variety of predicted categories and leverages world knowledge on query categorization.

In this method we prompt a pretrained LLM to select item categories, applicable to a given broad query. In pilot experiments we noticed that the LLM cannot correctly handle the direct classification task, when all category labels are given in the prompt (as the number of them is too large). Therefore we split the task into two smaller steps:

1. Given an input query, the model is instructed to output a comprehensive list of free-form product categories, which the customer might expect to see in search results;

2. each generated category is then mapped by the LLM to match one of the valid categories.

As a result, we obtain a list of LLM annotations for each candidate query.

Labeling search queries with an LLM overcomes the limitations associated the previous two methods, bringing in an alternative view for query classification. In the example with '*home spa gift basket*', in Table 1, the LLM proposed more creative categories, such as *candle* and *bath toy*, which were not captured by the other two approaches. This shows that external knowledge can help to complement our suite of labelers.

As mentioned earlier, human annotations are suboptimal for the broad query classification task because of individual preference differences (e.g., '*gift for boyfriend*' could mean a *plush toy* or a *screwdriver*). LLMs tend to "average out" such preferences, producing categories that reflect typical or popular expectations.

While being a valuable addition to the log-based methods, the LLM-based labeler does not scale as easily as the other two methods because computing predictions for it is more costly and slower. We account for this by designing our methodology in two steps: broad queries pre-selection and subsequent labeling saves resources, compared to labeling a large amount of queries and filtering the multi-labeled ones later.

In summary, we have chosen three distinct methods to provide query-category associations. Each one comes with its own set of advantages and underlying assumptions, data, and algorithms; the three methods were selected to complement each other. While each of the individual labeling approaches is established, our contribution lies in effectively combining these heterogeneous signals into a principled aggregation framework.

Additionally we observe that inferring query categories from aggregated click logs and from relevance labels is mildly correlated (at 0.5 Jaccard similarity on the per-query label sets). The LLM-based method stands out as the method that has the least label correlation with the other two (0.17 Jaccard similarity each).

query	'home spa gift basket'	'comfy male clothes'
clicks	[SKIN_CARE_AGENT, BATHWATER_ADDITIVE]	[SWEATER, SHIRT, PANTS]
relevance	[SKIN_CARE_AGENT, LIP_BALM, TOWEL, COOKIE, ROBE, BATHWATER_ADDITIVE]	[SHIRT, SKIRT, PANTS, SHORTS, TUNIC, TRACK_SUIT]
LLM	[CANDLE, SKIN_CARE_AGENT, TOWEL, ROBE, BLANKET, TOY_FIGURE]	[SHIRT, PANTS, SOCKS, COAT, TUNIC, PAJAMAS]
combined	[BATHWATER_ADDITIVE, SKIN_CARE_AGENT, TOWEL]	[SHIRT, PANTS, TUNIC, SOCKS]

Table 2: Examples of 3 labeling methods and their aggregation

3.3 Aggregation

As a final step, we aggregate the outputs of all three previously described signal sources into a single set for each query. We experimented with two distinct aggregation methods.

As a baseline approach, we implemented a simple majority voting scheme (MV), where for each category to be accepted as the final prediction for a query, we required that at least two methods predict any given category.

Secondly, we implemented the Dawid-Skene (Dawid and Skene, 1979) (DS) approach. Dawid-Skene is a probabilistic method of aggregating annotations, which estimates the expertise of the workers by building confusion matrices. Normally, in DS the labels assigned to the task by the annotators should be ordinal and not multi-label. To adapt it for multi-label cases, we learn the expertise matrices per category. For each category k the input to the aggregation model from each labeling method is a pair <query, prediction$_k$>, where prediction$_k$ is a binary value, indicating whether the method has outputted the category c_k for the query q.

When selecting an aggregation approach, practitioners can decide for one of the described methods, based on their characteristics. MV is a simple method, which can be computed with minimal compute cost even for very large query and category sets, while DS needs to be trained iteratively. On the other hand, DS allows to incorporate useful information about the annotator expertise per label. In the long-tailed multi-label setup, DS allows to preserve rare labels more frequently. That aligns with our empirical findings, described in Section 4.2, which is why we recommend DS as a preferable aggregation method.

Additionally, we experimented with incorporating relevance model confidence scores and click-through rates into Dawid–Skene, by bucketizing the confidence scores $s(q, c_k)$, associated with each label. However, upon inspection, the results of this modification proved to be worse than the binary label relevance estimations.

4 Implementation and analysis

Following the outlined annotation and aggregation methods, we collected an experimental sample of e-commerce queries in English. Across all three methods, we used an ontology of product categories to label the queries.

The size of the resulting pilot dataset amounted to approximately 5k, as we aimed to sample around 1.5k queries for each broad query type among *activity*, *gift*, *accessory* and *outfit*. We purposefully created this sample to analyze the proposed method; for real life tasks it can be trivially scaled (see Section 4.3), as all steps of our methodology are automated and require no human intervention. Table 1 shows examples of sampled queries per type.

To tune the hyperparameters $t1$ (minimal click-through rate) and $t2$ (minimum number of relevant products), we used a separate holdout set. This set consists entirely of *brand queries* for which we know what product categories they belong to (because we know the set of categories any given brand sells). On this brand set, we then selected $t1$ and $t2$ such that they maximize the overlap between the true labels and the ones from clicks and relevance methods respectively. The LLM[1] prompts used for labeling are provided in Appendix 6.

Finally, as an additional post-processing step, we removed search queries that had very long aggregated annotations (i.e., vague queries, like 'cheap supplies', which are out of scope here). For that we fixed the maximum number of predicted categories as $mean+3 \cdot std$ of all lengths in the dataset, which, given the mean of 7.1, resulted in a maximum 20 categories per query.

The obtained labeled English dataset can be extended to other languages by i) collecting broad queries with language-specific keywords (e.g., use keywords 'geschenk' in German to detect *gift* broad query category), and ii) labeling the these queries with language-specific click data, relevance model and prompting the LLM with the target language.

[1]We used Claude 3 Sonnet by Anthropic, March 4, 2024 https://www.anthropic.com/news/claude-3-family

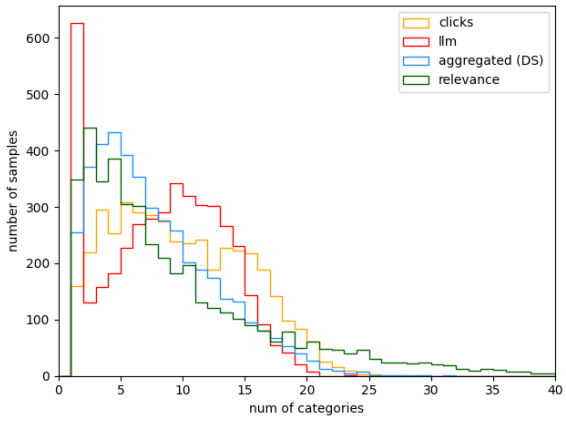

Figure 2: Histogram of the number of categories generated by the considered methods. The aggregated method (DS) plot averages the behavior of individual methods.

4.1 Analysis

Annotation lengths. In Figure 2 we show the distribution of annotation lengths for each approach. From this chart we can make the following observations:

- As noted in Section 3.2, click-based labeling is subject to exposure bias, and we therefore expected the distribution to be left-skewed towards fewer labels. Interestingly, Figure 2 shows that this effect is less pronounced in our data than anticipated.

- The relevance approach has the largest number of queries with >20 categories. The reason is that for a broad query a lot of items across many categories can potentially be of interest for the user, and therefore, the relevance model deemed those items as relevant.

- LLM-based annotations strike balance between short and long annotations, having the critical mass of them between 5-20 labels. However, LLM curve also has a spike at queries with 1-2 labels. Upon examination, we found that in those cases the LLM outputted a broad category (e.g. *camping equipment* for the broad query '*camping accessories*'), instead of multiple fine-grained ones.

- The count histogram for Dawid-Skene averages the aggregated components, effectively correcting the overly long annotations from relevance method and the single-label ones from LLM.

Labeling examples. In Table 2 we provide the examples of query annotations for the considered methods and their aggregation. We note that clicks and relevance methods have similar annotations. For the query '*home spa gift basket*' the relevance model outputted the category *cookie*, which makes sense as a gift but not as a shower gift: this can be attributed to the confusion of the relevance model. At the same time, for this query LLM produces more distinct and creative predictions: e.g., *toy figure*, which could be a rubber duck.

Similarly for the query '*comfy male clothes*' LLM predicts the category *pajamas*, that is absent in the other methods. The predictions of the click model are restricted to the few most popular categories. Moreover, the category *sweater* predicted by the click model is an indication of the method's dependency on the seasonality in user behavior.

4.2 Human assessment

We also conducted a manual audit of all labeling methods and the aggregated results. Given that the assessment for the broad queries is highly subjective, we formulated the task as a relative comparison among 5 labeling options (3 individual methods + 2 aggregated sets). For each considered query, we instructed the annotator to indicate the best option(s). The evaluators were asked to take into account the precision/recall of the annotations when making a decision. The evaluator could choose multiple acceptable annotations for the sample, in case the annotations are similar or insignificantly different, or select none.

The annotation task was carried out on 50 queries with two human evaluators. Both annotators were specifically trained for this task using detailed guidelines and example cases to ensure consistent and informed annotations.

The evaluation was performed in two stages. First, Annotator A labeled all instances according to the annotation guidelines. Then, Annotator B reviewed these annotations and either confirmed or corrected them as necessary. In cases where Annotator B disagreed with Annotator A's label, Annotator B's correction was treated as the final label. We did not compute inter-annotator agreement metrics, as annotations were not performed independently. Instead, we report the proportion of annotations that required correction as an indicator of initial annotation quality and guideline clarity. The fraction of queries with corrections was 11%, showing high annotator agreement.

	clicks	relevance	LLM	DS	MV
score	21/50	17/50	11/50	**31/50**	24/50

Table 3: Results of human audit of 50 queries. The scores reflect for each method the fraction of outputs rated as acceptable.

Results. In Table 3 we show the results of the manual assessment, where each score corresponds to the number of times the annotator selected the corresponding labeling method's predictions as acceptable. Although the LLM has less domain knowledge its contribution is significant.

We observe that both aggregation methods are favored by humans, with Dawid-Skene being the best method. DS was preferred over MV in 15% of cases; the annotators noted that DS produced longer and more diverse annotations. Achieving the majority agreement (2 out of 3 labelers) in MV is a strict measure and results in valuable product types removed from the annotation. As opposed to majority voting, Dawid-Skene assigns different weights to the predictions of the workers, which results in less restrictive filtering of the labels. As a result of that we found the average length of MV annotations to be 10% shorter than that of DS.

4.3 Practical applications

The presented dataset with approximately 5k queries across four categories is a controlled pilot, designed to allow careful auditing and to avoid repetitive queries. However, the entire pipeline is fully automated and trivially scales to millions of queries and dozens of categories.

In practice, we have used a larger-scale dataset built with this methodology to evaluate a production e-commerce query classifier. While results cannot be disclosed for privacy reasons, we note that this evaluation surfaced some weaknesses of the model on specific types of broad queries — demonstrating the practical value of *FABRIC* in real-world systems.

5 Conclusion

In this work, we have presented the methodology for labeling multi-category e-commerce search queries, which can be used to provide high quality multi-label data for the query classification model evaluation. Our solution is generic and can be used in various e-commerce stores that have aggregated click-through logs and a relevance model available.

As future work we plan to extend the scope of the considered types of broad queries and individual labeling approaches. Beyond that, we plan to explore additional methods for label aggregation.

6 Limitations

The described approach requires access to e-commerce aggregated search/click logs; while these are usually available for e-commerce stores, they are difficult to access in the academic setting. Moreover, the proposed approaches rely on sufficiently large observations of queries, which might not be available for all use cases.

References

Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-market product recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 110–119.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Yiu-Chang Lin, Ankur Datta, and Giuseppe Di Fabbrizio. 2018. E-commerce product query classification using implicit user's feedback from clicks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1955–1959. IEEE.

Chen Luo, William Headden, Neela Avudaiappan, Haoming Jiang, Tianyu Cao, Qingyu Yin, Yifan Gao, Zheng Li, Rahul Goutam, Haiyang Zhang, and 1 others. 2022. Query attribute recommendation at amazon search. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 506–508.

Yiming Qiu, Chenyu Zhao, Han Zhang, Jingwei Zhuo, Tianhao Li, Xiaowei Zhang, Songlin Wang, Sulong Xu, Bo Long, and Wen-Yun Yang. 2022. Pre-training tasks for user intent detection and embedding retrieval in e-commerce search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4424–4428.

Anna Tigunova, Ghadir Eraisha, and Ezgi Akcora. 2025. squirrel: Large-scale evaluation of e-commerce query classification models. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 485–489.

Anna Tigunova, Thomas Ricatte, and Ghadir Eraisha. 2024. Transfer learning for e-commerce query product type prediction. *arXiv preprint arXiv:2410.07121*.

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

and Qiang Yang. 2021a. 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*, pages 4362–4372.

Junhao Zhang, Weidi Xu, Jianhui Ji, Xi Chen, Hongbo Deng, and Keping Yang. 2021b. Modeling across-context attention for long-tail query classification in e-commerce. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 58–66.

Lvxing Zhu, Hao Chen, Chao Wei, and Weiru Zhang. 2022. 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)*, pages 141–150.

Lvxing Zhu, Kexin Zhang, Hao Chen, Chao Wei, Weiru Zhang, Haihong Tang, and Xiu Li. 2023. Hcl4qc: Incorporating hierarchical category structures into contrastive learning for e-commerce query classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3647–3656.

Appendix

Prompts for LLM annotation

System prompt

```
You are an experienced e-commerce website manager. You are an expert in identifying
which category of products the users of the e-commerce websites look for.
```

Step 1: assign free-form categories

```
You are given as input a customer query, and you need to list all possible product types
that the customer might be interested to see in their search results.
Try to give a very extensive list of all applicable product categories

Output the list of categories as a Python list of singular nouns.
Restrict your outputted list to the top 10-20 most essential product categories that I
need to show on the first search result page. Output categories whose products
are relevant for the query.
Do not write abstract categories (e.g. outputting 'accessories' is not allowed),
Each category needs to be grounded in a specific product that the user can
buy on an e-commerce website.

Sort the results starting with most relevant categories first
Input query: {query}
```

Step 2: map to real product categories

```
I will give you a list of standard names.
For each input concept, return the closest in meaning standard name from the
standard list.
For instance, the input concept 'herbal remedies' you will standardize
into "HERBAL_SUPPLEMENT", which is in the list.

If you cannot find a matching standard name, output None. It is ok to output same
output standard names for different input concepts.

You will be given a list of input concepts and you will need to output a list of
standardized concepts.

IMPORTANT: do not output anything outside of the given list of standard names.
Here is a list of standard names that you need to standardize the input into: {cat_list}
Input concepts: {inputs}
```