Query Rewriting for Voice Shopping Null Queries

Iftah Gamzu
Amazon
Tel-Aviv, Israel
iftah@amazon.com

Marina Haikin
Amazon
Tel-Aviv, Israel
mhaikin@amazon.com

Nissim Halabi
Amazon
Tel-Aviv, Israel
nissimh@amazon.com

ABSTRACT

Voice shopping using natural language introduces new challenges related to customer queries, like handling mispronounced, misexpressed, and misunderstood queries. Voice null queries, which result in no offers, have negative impact on customers shopping experience. Query rewriting (QR) attempts to automatically replace null queries with alternatives that lead to relevant results. We present a new approach for pre-retrieval QR of voice shopping null queries. Our proposed QR framework first generates alternative queries using a search index-based approach that targets different potential failures in voice queries. Then, a machine-learning component ranks these alternatives, and the original query is amended by the selected alternative. We provide an experimental evaluation of our approach based on data logs of a commercial voice assistant and an e-commerce website, demonstrating that it outperforms several baselines by more than 22%. Our evaluation also highlights an interesting phenomenon, showing that web shopping null queries are considerably different, and apparently easier to fix, than voice queries. This further substantiates the use of specialized mechanisms for the voice domain. We believe that our proposed framework, mapping tail queries to head queries, is of independent interest since it can be extended and applied to other domains.

CCS CONCEPTS

• **Information systems → Information retrieval query processing**.

KEYWORDS

query rewriting, voice assistant, e-commerce search, null query, voice search

ACM Reference Format:

Iftah Gamzu, Marina Haikin, and Nissim Halabi. 2020. Query Rewriting for Voice Shopping Null Queries. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401052

1 INTRODUCTION

In the past few years, speech has emerged has a natural mean for communicating information need to various search engines via voice assistants such as Amazon Alexa, Apple Siri, and Google Assistant. Among various experiences, voice assistants enable customers to search and shop for products in an intuitive way using natural language. Providing a free-form shopping experience is a challenging task. This is especially true because the voice interface lacks assisting mechanisms, such as query completion and refinement, that can help customers easily express their need and iterate on it. As a consequence, a non-negligible portion of all voice shopping queries are *null queries*, that is, queries that result with no offers. Such interactions clearly have negative impact on customers shopping experience [40–42]. Query rewriting (QR) attempts to seamlessly replace null queries with alternatives that lead to relevant offers for the customer intent, and by that, help customers progress on their shopping journey.

The voice interface introduces technical and architectural challenges that are either unique or more emphasized compared to textual interfaces (e.g., web search). For example, the voice interface lacks assisting mechanisms that are common in the web, resulting in more shopping queries that contain erroneous attributes such as product brand, quantity, and others; While automatic speech recognition (ASR) and natural language understanding (NLU) techniques have made considerable progress in recent years (see, e.g., [27, 30, 36]), they still have failures that exhibit different patterns (e.g., phonetic) than common textual errors; Finally, customers fail in formulating a coherent spoken query in real-time, resulting in new types of speech imperfection errors such as stammered or misspronounced words. All these motivate the application of specialized mechanisms for the voice domain.

Our contribution. We introduce a pre-retrieval QR approach designed to handle voice shopping null queries, as illustrated in Figure 1. In a pre-retrieval approach [11], QR happens without knowledge about the offers that return for the original query or any of the generated alternatives. This approach is more efficient than post-retrieval methods, inducing less overhead to the search system. Upon receiving a transcribed query q, our QR framework applies a two step process of *alternative queries generation* and *alternative queries ranking*. As voice-user interface introduces a strong presentation bias towards the top-ranked offer [20], in most cases, only a single offer is communicated back to the customer. Therefore, practically, it is sufficient to identify a single "hero" alternative query to amend the customer query. The top-ranked alternative is passed along with the original query to a *controller*, which ensures the quality of the selected alternative. Specifically, the controller implements quality safeguards that prevent replacing the original query with an alternative query that is likely to retrieve irrelevant results to the intent of the original query. Then, the original query q is sent along with the selected alternative query r^* for retrieving offers from the e-commerce search engine. If q turns to be a null

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401052

query with no offers, the customer is presented with the offers retrieved for r^*.

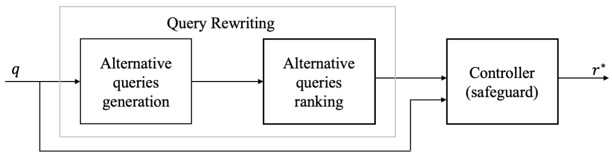

Figure 1: Query rewriting flow.

Our *alternative queries generation* component is based on identifying web and voice shopping queries that frequently lead to positive events (e.g., purchases), and indexing them in a search engine. Those positive queries are indexed using various analyzers that target different potential failures in voice shopping (e.g., analyzers based on textual, *n*-grams, and phonetic similarities). Given a query, one can retrieve multiple alternatives from the index in hope that one of them can amend it, without the need to isolate the exact error. Conceptually, the core of our generation approach is by mapping tail (low frequency) queries to alternative head (high frequency) queries. The main motivation for this approach comes from the fact that head queries are known to exhibit much better performance than tail queries, due to richer historical behavioral features [23]. In fact, the distinction is even more extreme in voice due to the aforementioned strong presentation bias. In addition, Ingber et al. [20] recently observed that products purchased through voice are much more limited in terms of diversity, namely, products purchased on a regular basis such as groceries, and not niche long-tail products. This provides another motivation for our approach as positive head queries correspond well with commonly and regularly purchased products.

The *alternative queries ranking* component then ranks the alternatives that are more probable to fix the original query. This machine learning-based component utilizes multiple features (like textual and semantic similarities between the originating query and the alternatives, behavioral features, and more) to make its decisions. While our alternative queries generation component aims to improve the recall, the alternative queries ranking component is in charge of tuning the precision. For instance, in an e-commerce rewriting scenario, it is essential for the alternative queries to retrieve offers that capture the same intent as the originating query [41]. Rewriting that leads to offers that do not respect the desired product type are clearly poor. Our ranking approach considers features that help preserve the original customer's intent (like, extracting the product type from queries), and by that, provides quality guarantees on the alternatives.

Consider for example the following scenario, which demonstrates the ability to fix ASR errors outside the ASR system. A customer that is interested in purchasing "epilepsy bracelets" communicates her need to a voice assistant. Due to ambient noise, the ASR transcribed query results in "apple upci uhhh bracelets", which turns to be a null query. However, "epilepsy bracelets" is a frequent positive query, and has the same phonetic representation, "APLP-SPRSLTS", as the transcribed query. Among the various alternatives, our QR framework selects "epilepsy bracelets", which is retrieved by a phonetic analyzer, and a relevant offer is presented.

We provide an experimental evaluation for both voice and web null queries based on data logs of a commercial voice assistant and an e-commerce website. Our evaluation demonstrates that our voice query rewriting (VQR) approach outperforms several baselines by large margins. For instance, we show that VQR improves over the effectiveness of a simple textual similarity retrieval-based QR system by roughly 22% on voice data, and a term-dropping QR system by roughly 46% on web data. Although our focus is on voice QR, the improvements observed on web data, reaffirms the utility of our approach. In fact, we believe that our proposed framework is generic enough to be successfully applied in other domains beyond e-commerce. One notable highlight from our evaluation is that a term-dropping QR approach applied to a random set of web null queries attains much better performance than when it is employed to a random set of voice null queries (by roughly 56%). This indicates that web e-commerce null queries are considerably different than voice null queries, and apparently easier to fix. This observation adds to previous line of research identifying differentiating factors between the voice and web domains. For example, it was observed that voice queries are closer to natural language than text queries in general search [15], voice reformulations are distinguishable from textual reformulations [17, 22], and that shopping categories and behavioral patterns defer between voice and web e-commerce search [20]. In summary, we make the following key contributions:

(1) We propose a new pre-retrieval QR framework for voice shopping null queries. Our approach maps tail queries to head queries, targets different potential failures in a voice-user interface, and aims to maintain the customer's intent.
(2) We reveal interesting insights regarding differences between voice and web null queries, establishing the need for specialized mechanisms treating the voice domain.

The rest of the paper is organized as follows: In Section 2, we review related work for QR. In Section 3, we describe our alternative queries generation approach, and in Section 4 we present our machine-learning model for ranking the alternatives. Section 5 presents an experimental evaluation of our proposed VQR approach. We discuss the positive findings of an online A/B tests that was conducted with our VQR approach on a commercial voice assistant in Section 6. We conclude the paper in Section 7.

We note that all processes performed as part of our analyses were conducted in accordance with strict privacy guidelines and all methods for handling the data are automated. We cannot share our data with the community due to its sensitivity.

2 RELATED WORK

Query rewriting has long been an important research area in information retrieval [3]. Extensive analysis has been done for handling and rewriting of queries in web search. The notion of query refinement, expansion, suggestion, substitution, and reformulation are sometime overloaded and have been commonly used synonymously with query rewriting. Most of the previous methods are not particularly suitable for e-commerce queries, which are shorter and more sensitive to context [34], let alone voice e-commerce queries. Focusing on tail low-frequency e-commerce queries highlights additional unique challenges and opportunities, especially around finding and ranking good query alternatives [13]. Using voice as a new medium for search also reveals differences from traditional search in both web [15, 17, 22] and e-commerce [20, 21]. In this work, we concentrate on studying voice e-commerce null (tail) queries.

One notable research direction in QR, which is also applicable for e-commerce, focuses on increasing the recall. This direction is especially important for null queries that yield no results. Alternative queries are generated by dropping [4, 24, 28, 46, 47] or substituting [7, 16, 25] tokens from the original query. For example, Jones et al. [25] proposed generating query alternatives by using a large set of ordered query pairs obtained from consecutive queries in web-search sessions. Then, various alternatives are generated by breaking a given query into segments and either dropping or generating substitutions for each of them separately. For dealing with null e-commerce queries, Singh et al. [41] suggested a post-retrieval approach for dropping terms from a given query that restricts the search results to the same taxonomy of results returned in the past for the original query. Tan et al. [43] suggested generating sub-queries by dropping unimportant terms based on their part-of-speech (POS) tag and additional features. Our results hint that term-dropping methods for e-commerce null queries do not adjust well to the voice domain. Other ideas for substitution-based solutions, using the query-flow graph [6], were also proposed [7, 16]. However, finding good recommendations for tail e-commerce queries based on session co-occurrence turns to be difficult [16]. For addressing also the long tail of the query distribution, Bonchi et al. [8] conceptually extend the query-flow graph with term nodes in addition to query nodes. Broccolo et al. [9] generate an inverted index of "successful" queries, i.e., ending query of a session with a click on its search result, and recommends queries retrieved from that index. Our approach has similarities with the later in using an index of successful queries, and extends the use of inverted index to include phonetic, sub-words, and semantic similarities upon retrieval.

Recently, several attempts were made to apply deep learning to various query rewriting tasks. Grbovic et al. [14] proposed to use embedding techniques to expand a query via a k-nearest neighbor search. He et al. [19] proposed a framework that learns to rewrite queries by unsupervised candidate generation and supervised candidate ranking. For unsupervised candidate generation, they presented a a sequence-to-sequence LSTM model, but also incorporated several existing QR systems suggestions. Their scoring function required training over a large web click data. Xiao et al. [44] applied a similar technique based on post-retrieval method for e-commerce web search. The applicability of these techniques to voice queries is still unclear, especially in light of the data sparsity challenges that still exist as voice interfaces are not yet widely adopted. Indeed, users do not tend to switch between voice and text when reformulating queries [39]. Jiang et al. [22] showed that reformulation patterns of voice queries are different from those in conventional textual searches using both lexical and phonetic changes. Hassan et al. [17] developed classifiers for distinguishing reformulation of voice query pairs from textual query pairs. They extended text-based approaches with voice signals such as phonetic similarity. This hints regarding the importance of phonetic representations in voice query rewriting. Our alternative queries generation approach does not require prior training and adjusts to the voice medium characteristics, considering its phonetic representation.

One related direction in QR focuses on improving its precision by narrowing down a search query. In this case, the goal is to refine the query such that the refined alternative retrieves a more relevant subset of results. Approaches towards this task include learning rewritings based on past users' query refinements [2, 31, 32] and applying pseudo-relevance feedback techniques [10, 29, 33, 45]. Those latter techniques commonly employ post-retrieval methods and iteratively query the search engine for newly added terms [12, 35]. In general, this direction is not suitable for handling null queries since the main problem with those queries is recall rather than precision. Those query refinement and rewriting techniques are mostly applicable to head queries.

3 GENERATING ALTERNATIVE QUERIES

The alternative queries generation component is responsible for identifying queries that have potential to successfully amend the original query. Our general approach is as follows. We build a search engine index (e.g., based on Elasticsearch [1]) that holds both web and voice queries that led to positive events, such as purchases or an adds-to-cart, with sufficient number of occurrences in the past. In practice, we indexed tens of millions of queries that were collected over a period of several months. Those queries are indexed while applying various analyzers that enable efficient retrieval. We discuss the specifics of those analyzers later on. Given a query q, an alternative queries set R_q is generated by utilizing each of the analyzers to retrieve a query with a maximal score (for that specific analyzer). We only consider the query with the highest score for each analyzer since we only need to identify a single "hero" offer by a single "hero" alternative to be presented to the customer due to the strong presentation bias in a voice-user interface.

We explore and combine multiple analyzers that target different potential types of errors in voice queries. An analyzer is typically composed of a tokenizer that splits the text into tokens, and a filter that is applied to the tokens, resulting with terms for indexing and retrieval. Arguably, the simplest analyzer that we utilize, referred to as MLT (more-like-this), splits the queries to their word-level. Then, given a query, the top retrieved queries are selected according to the Okapi BM25 scoring function [38] on the underlying terms. This analyzer can identify good alternative queries if the most distinctive words in the query are kept. For example, rewriting over-specified queries or miss-pronounced queries. Such a method, based on word tokens, fails when important distinctive words are corrupted, e.g., suffer from spelling mistakes. In such cases, it is much better to apply methods that focus on more local patterns. Taking this observation into consideration, we also utilize an analyzer based on character n-grams, that is, an analyzer that breaks the query into (partially overlapping) terms of character length n. When working with voice queries, there are additional special failures, like phonetic errors that originate in ASR systems. In order to cope with such errors, we also employ few phonetic analyzers based on Double Metaphone phonetic encoding [37]. One such analyzer works similarly to MLT, but on a phonetic encoding level.

It can successfully replace a query if the main keywords preserve their phonetic representation. Two additional phonetic analyzers that are used are a so-called full phonetic analyzer that considers the entire phonetic representation of a query as a single term, and a phonetic n-grams analyzer, which breaks the query into parts of length n and translates them into their phonetic representation. The former analyzer handles ASR errors that result in splitting or merging of words. For concreteness, we focus on n-grams-type analyzers with $n \in \{3, 4\}$. Table 1 presents a simple example illustrating the way different analyzers generate terms for indexing and retrieval. Table 2 outlines few synthetic examples that demonstrate the strengths of the various analyzers.

Table 1: Analyzers and their resulting query terms for the query "dog food".

Analyzer	Query terms
MLT (text words)	{"dog", "food"}
full phonetic (one term)	{"TKFT"}
phonetic (phonetic words)	{"TK", "FT"}
4-grams	{"dog ", "og f", "g fo", " foo", "food"}
phonetic 4-grams	{"TK", "AFK", "KF", "F", "FT"}

4 RANKING ALTERNATIVE QUERIES

Once the set of alternative queries R_q has been retrieved, the alternative queries ranking component evaluates the probability of each of the alternatives to successfully amend the originating query q. While our component provides a complete ordering between the alternatives, for our voice scenario, we are only interested in the top-ranked alternative $r^* \in R_q$. The reason lies in the presentation bias towards a single top-ranked offer.

Every alternative is retrieved as the top option by some analyzer to replace q. However, it may occur that a retrieved alternative is irrelevant for the originating customer intent. This is especially true for phonetic-based analyzers that due to their generality of sounds-alike retrieval may result in an alternative query with different intent, but it is also true for the other types of analyzers. For this reason, we build a machine learning approach that not only ranks the alternatives, but also provides a confidence score regarding their probability to properly replace q. In a sense, the alternative queries generation component aims to improve the recall of our system, while our pointwise ranking component is in charge of tuning its precision. Table 3 exhibits an example in which textual and character-based analyzers fail to generate relevant alternatives, but phonetic-based analyzers do. Notice that the product type "strips" appears in the original query and in the alternatives retrieved from the phonetic-based analyzers. Having agreement between extracted product types from queries is a strong indicator for keeping the customer's intent intact.

We assume to have a labeled dataset for voice shopping queries. Given a null query q, the dataset consists of a label $y_{(q,r)}$ for every (q, r) query pair, where $r \in R_q$ is a generated alternative for q. This dataset is the result of annotation by an internal team dedicated to voice shopping related annotation tasks. The annotators are given the customer's utterance that corresponds to q along with the top retrieved offers for each alternative $r \in R_q$. For each such offer, the annotators indicate whether it is relevant to the utterance of q or not. Because the annotators classify the relevance based on both the customer intent as implied by the utterance and the returned offers for the alternatives, the classification label captures both up-stream errors due to the voice interface and down-stream errors due to the products search engine. The relevance label $y_{(q,r)}$ is set to be the relevance annotation between q and the top returned offer for r.

We collect features associated to each alternative r as well as various features relating to the relation between q and r (e.g., their textual similarities). Those features are utilized in conjunction with the dataset to optimize over several machine learning models and hyper-parameters. We describe the details of the features and the ranking models in the following subsections.

4.1 Features For Ranking

We evaluated a large family of tens of handcrafted features as well as different standard manipulations of them for our alternative queries ranking approach. We refrain from providing an exhaustive list of evaluated features due to space constraints and since most of them had marginal contribution to the performance of the models. Instead, we provide an overview of the main feature categories that guided the feature engineering, and focus on a small subset of carefully selected features that were able to extract most of our performance improvements. Our main feature categories include:

Textual similarities between the originating query and an alternative query, and between their extracted metadata (e.g., product type and brand). We consider variants of the Jaccard similarity and Edit distance. We consider distance variants on both word-level and character-level. We also consider distances of lemmatized queries, and distances of the lexicographic representations of queries (i.e., ordering of words by lexicographic order). Finally, we use the BM25 relevance score of the specific analyzer that retrieved the alternative as another tokens similarity measure.

Semantic similarities between the originating query and an alternative query, and between their extracted metadata. We generated a specialized word embedding trained using the FastText algorithm [5, 26] on a proprietary e-commerce corpus that included shopping queries, product titles, and a concatenation of queries with the title of the product that was purchased as result of them. The semantic similarity between two sentences is defined as the cosine similarity between the average vector of their underlying normalized word embedding vectors.

Historical behavioral features of the alternative queries. Those features capture the aggregated past performance of the alternative queries. We consider the number of past occurrences and positive actions (purchase or add-to-cart) of a query on both web and voice shopping logs, and its implied conversion rate (i.e., the rate between positive actions and its overall number of occurrences). We note that the behavioral features of the alternative queries are rich as those queries are head queries.

We used standard feature selection techniques to avoid overfitting due to small amount of available labeled data (i.e., few thousands). In what follows, we focus on a short list of 7 features that were able to extract most of our performance improvement. Table 4 lists those features. In feature f_1 the sorted version of a query is the

Table 2: Representative synthetic examples of voice shopping null queries and their alternatives.

Original query	Alternative query	Error type	Analyzer
willie the children's toothpaste	waleda childrens toothpaste	ASR (splitting)	full phonetic
kokyu 10 vitamin	co q10 vitamin	ASR	full phonetic, phonetic
psn amino x	bsn amino x	ASR	4-grams, phonetic
pur purina pro plan fortiflora	purina pro plan fortiflora	miss-pronunciation	4-grams, MLT
moon and sun coin shaped silicone mold	sun moon silicone mold	over-specification	MLT
cara b moisturizing hair mist	caribbean moisturizing hair mist	ASR	phonetic 4-grams

Table 3: Alternative queries generated by different analyzers for the query "kitten maja strips", and the top products returned when running those against Amazon's web search.

Method	full phonetic	phonetic	MLT	4-grams
Alternative	ketone mojo strips	mojo ketone strips	maja	kitten mat
Top product title	Keto-Mojo 50 Blood Ketone Test Strips	Keto-Mojo 50 Blood Ketone Test Strips	Maja Soaps	Pieviev Cat Litter Mat Litter Trapping Mat
Product image				

Table 4: Top features utilized by the ranking models.

Feature	Category
f_1 - Normalized word-level edit distance between sorted versions of (q, r)	textual similarity
f_2 - Normalized word-level edit distance between queries (q, r)	textual similarity
f_3 - Jaccard similarity between queries (q, r)	textual similarity
f_4 - Semantic similarity between queries (q, r)	semantic similarity
f_5 - Semantic similarity between the product types $(\mathrm{PT}(q), \mathrm{PT}(r))$	semantic similarity
f_6 - Analyzer-specific BM25 score for r given q	tokens similarity
f_7 - Conversion rate of r	behavioral

one in which the words are lexicographically ordered. Note that in feature f_5, $\mathrm{PT}(q)$ indicates the product type implied for the query q as extracted by a specialized model. A key challenge in mapping a tail query to a head query is to maintain the purchase intent of the query. Product type alignment is arguably the most coarse way to keep the intent intact.

4.2 The Ranking Model

Our machine learning approach for the alternative queries ranking problem is a two-tier solution. First, for each analyzer type, we train a machine learning model that predicts whether an alternative r, generated by that analyzer, is a suitable replacement for a null query q. Then, we rank all alternative queries according to the different models predictions, and effectively, return the alternative with the highest prediction score. For the purpose of building each analyzer's model, we have experimented with several machine learning approaches (e.g., logistic regression, support vector machine, random forest) and various hyper-parameters (e.g., the main hyper-parameters for random forest were maximal depth and the number of estimators in the forest). Following an exhaustive search, we identified the best model for this task as a random forest model optimized over a cross-entropy loss. Further technical details about the models are provided in Section 5.3.

5 EXPERIMENTAL EVALUATION

5.1 Baselines

We compare our VQR approach against two baselines. The first is a term-dropping based QR system, which is common in various use-cases, especially when dealing with null queries [4, 24, 28, 41, 43, 46]. Term-dropping approaches have been demonstrated to be good for increasing the recall of search systems, but they often result in lower precision. The second baseline is a simple retrieval-based approach, build on top of our positive queries index, which only considers word-level textual similarity. This baseline has the ability to both relax or extend the originating query, taking into account the importance of words with respect to a reference collection of queries. This baseline is expected to have better precision than the term-dropping baseline.

Term-Dropping QR (TDQR). Alternative queries are generated by exhaustively considering all options to drop different set of terms from the originating query. Selecting a top alternative is done using a post-retrieval method that gives preference to alternatives that lead to more focused set of offers (i.e., having high query scope measure [18, 28]).

Retrieval-Based QR (RQR). An alternative query is selected based on a retrieval process that uses the same search index as VQR with an Okapi BM25 scoring function on the words. To guarantee high precision from this system, we required alternatives to have a sufficiently high relevance (specifically, a relevance score of 20). This threshold was identified as leading to good separation between higher and lower quality alternatives by manual annotation.

5.2 Datasets

For the purpose of evaluating our approach, we collected two datasets based on a random sample of voice null queries and a random sample of web (text) null queries. All sampled queries were unique, appearing only once in the sample, as expected from tail queries. Each query in the dataset is appended with a list of its alternative queries and their annotated relevance ranking. We relied on an internal team of annotators, dedicated to voice shopping related annotation tasks in order to produce those datasets.

The relevance annotations of alternative queries for an originating query were performed as follows. The annotators were given the customer's utterance (voice query) U that resulted in the textual query q, along with a list of product offers. They were asked to rate the relevance of each offer with respect to the customer intent as implied by the utterance U on a three-points-scale: 0 indicates a no match between U and the offer, 1 indicates a match that (at least) respects the requested product type, and 2 indicates a full match (i.e., the offered product contains all the attributes mentioned in the utterance). The list of product offers consists of the top-ranked offer retrieved for every alternative query r generated for the query q. The alternative queries were generated by the different approaches under evaluation, that is, the baselines and a set of 6 analyzers (MLT, full phonetic, phonetic, 4-grams, 3-grams, and phonetic 4-grams). The relevance label for each (q, r) query pair was set to be the relevance score between utterance U, which resulted in q, and the top returned offer for the alternative r, as agreed by majority of 3 annotators. Note that because the annotators classify the relevance of offers with respect to the customer utterance, the annotations also capture failures due to the voice interface, like ASR errors and user mispronunciations.

Voice null queries. A set of voice shopping null queries uniformly sampled at random over a period of one month from query logs of a commercial voice assistant. We consider only queries identified by the annotators to include a shopping intent. The dataset contains 3,564 unique null shopping queries along with their alternative queries rankings. The methods presented in this paper were developed using this dataset of voice null queries.

Web (text) null queries. A set of textual shopping null queries uniformly sampled at random over a period of one month from query logs of a commercial e-commerce website. The dataset contains 1,362 unique null shopping queries along with their alternative queries rankings. Note that this dataset is only used to evaluate the performance of the different methods for web e-commerce traffic. In this case, we consider multiple offers for each alternative (instead of only a single top-ranked offer) in order to evaluate the resulting offers ranking. This is motivated by the fact that the presentation bias on the web is much lower than on a voice interface, and thus, associating relevance only with a top offer makes less sense.

5.3 Training Random Forest Models

The models for predicting the quality of alternatives of each analyzer type were only trained on the voice null queries dataset. We split the dataset so that 75% of the examples are used for training and validation, and 25% are used for testing, and translated the three-points relevance labeling to a binary labeling by assigning double weight for full match cases. Because the dataset is relatively small, we train the model by using 2 repetitions of 3-fold cross-validation. Each of the random forest models for the analyzers were trained separately with same features and model parameters.

It is interesting to observe that some features have different importance for different analyzers. For example, for the full phonetic analyzer, textual similarity without words sorting is much more important than the one with sorting. This makes a lot of sense because full phonetic method retrieves alternatives that have very similar ordering of words, and therefore evaluating them based on a modified order can be detrimental. On the other hand, for the MLT analyzer, textual similarity with words sorting is more important than the one without sorting. Again, this is reasonable since MLT focuses on identifying alternatives that have similar intent but not necessarily with the same order of words.

We considered different combinations of analyzers and corresponding prediction models in order to understand the performance trade-offs. To simplify the presentation, we focus on two variants of VQR, differing by the set of employed analyzers:

VQR-4. Utilizes a set of arguably the most basic analyzers (phonetic, full phonetic, 4-grams, and MLT) that spans the 3 families of similarities (word-level, character-level, and phoneme-level). In this setting, all random forest classifiers achieved the best results when the maximal tree depth was 6 and the number of estimators in the forest was bounded by 130.

VQR-6. Uses 6 analyzers (those of VQR-4 appended with 3-grams and phonetic 4-grams). As the two additional analyzers have higher retrieval latency, this variant provides some insight regarding latency-performance trade-offs. In this setting, all random forest classifiers achieved the best results with a maximal tree depth of 6 and number of estimators in the forest bounded by 170.

5.4 Evaluation Metrics

We evaluate all approaches using 3 key metrics:

- Coverage – The ratio between the number of null queries for which an underlying method generates an alternative, and the overall number of null queries.
- Precision (P@1) – The ratio between the number of alternative queries that were identified as suitable replacements and the overall number of alternative queries.
- Effectiveness (E@1) – The multiplication between the coverage and precision. This metric essentially captures the impact on the customer, namely, the rate of null queries that were changed for the better.

For the evaluation of the approaches on the web null queries dataset, we also consider few metrics that measure the quality of the resulting ranked offers. These metrics include $P_{max}@3$, which captures the ratio of alternative queries that have at least one relevant offer within their top 3 offers, and the corresponding effectiveness metric $E_{max}@3$. We also consider $nDCG_3$, which uses a graded relevance scale, and its corresponding effectiveness metric $EnDCG_3$.

5.5 Experimental Results

We report our results on the test sets of both the voice and web null queries datasets. For confidentiality reasons, we report the evaluation metrics relative to the RQR baseline, omitting absolute metrics numbers. We begin by noting that although the effectiveness measure captures the direct impact on the customer, there is a subtle point here. Optimizing the effectiveness measure can be achieved by providing alternative queries whenever possible. However, from a customer point of view, it may be better that the system will not select an alternative when that query results in irrelevant offers. This observation is especially true on voice, when the presentation of irreverent offers create much higher friction than a graceful failure. This is one of the reasons for the existence of a controller component (recall Figure 1), safeguarding the quality. Consequently, although the best effectiveness is achieved by not setting any safeguard, we restrict our attention to solutions whose precision is around that of the RQR baseline.

Based on performance analysis on the validation set, we chose a conservative quality threshold of 0.4 on the prediction score of the VQR-6 model. Note that although we concentrate on a specific operating point, VQR-6 model has a wide-range of operating points that have higher precision, coverage and effectiveness than the RQR baseline. Figure 2 illustrates the effectiveness and precision trade-off curves of VQR-6 model (relative to RQR) as a function of the threshold given by the prediction score of the VQR-6 model (x-axis). Notice that the shape of the curve suggests that the model can adjust for different business use-cases by a careful selection of the threshold, e.g., one can optimize for higher-precision customer-facing scenarios. The selected operating point where the prediction score (threshold) equals 0.4 is visualized upon the knee of the effectiveness curve, and as claimed, its achieved precision is higher than the baseline RQR.

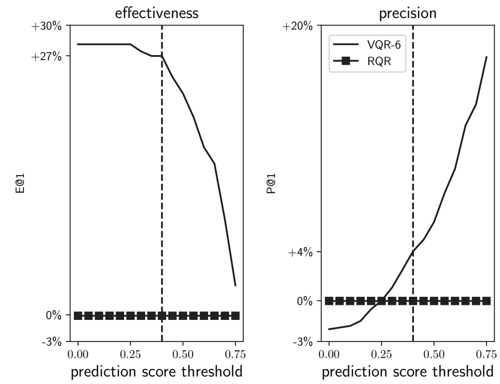

Figure 2: The effectiveness and precision of VQR-6 and RQR as a function of the quality threshold.

Figure 3 provides an alternative view, exhibiting the trade-offs of precision and effectiveness (y-axis) against the coverage (x-axis). The curves are generated by different thresholds on the prediction score of the VQR-6 model. It is evident that the VQR model has a wide range of operating points that have higher precision, coverage and effectiveness than RQR. Note that the precision of VQR-6 decreases as the coverage increases, indicating that high prediction scores of the model correspond to more confident query rewritings.

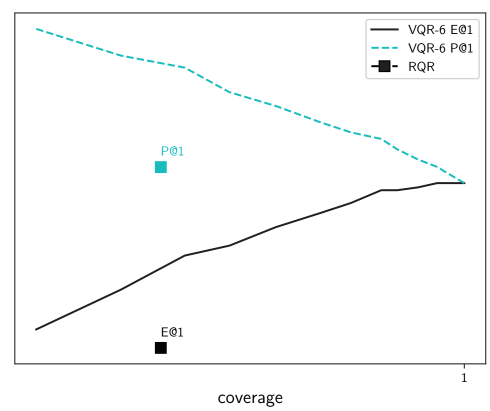

Figure 3: The effectiveness and precision of VQR-6 and RQR as a function of the coverage. Note that effectiveness and precision are equal when the coverage is 1.

Figure 4 illustrates the cumulative distributions of the prediction scores of the selected alternatives by the VQR-6 model over the validation set. One curve corresponds to the selected alternative queries labeled as relevant, and the second curve corresponds to less-relevant selected alternative queries. We notice that setting a quality threshold at 0.4 blocks many less-relevant alternatives while keeping almost all relevant alternatives. This trend can also be inferred from Figure 2, where the precision is consistently rising while the threshold increases towards 0.4, but the effectiveness hardly decreases. It seems important to note that each of the cumulative distribution curves is self-contained in the sense that it only takes into account either the number of relevant cases or the less-relevant cases.

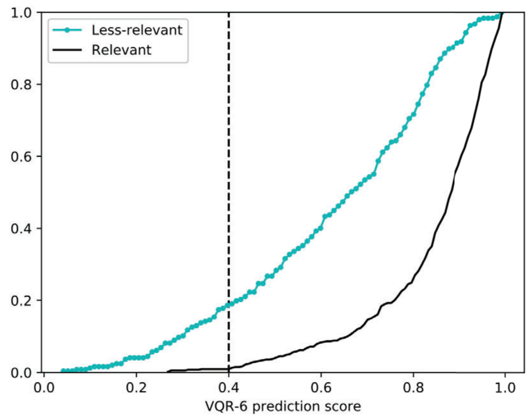

Figure 4: The cumulative distribution of prediction scores for relevant and less-relevant alternatives.

Table 5 presents an evaluation of the different methods on the voice null queries test set. Reported values are statistically significant with $p < 0.05$. As can be observed, the baseline RQR approach significantly outperforms the effectiveness of TDQR baseline. While the coverage of TDQR is higher than RQR by more than 22%, its precision is significantly lower, resulting in an overall decrease in effectiveness. Both VQR-4 and VQR-6 models outperform the effectiveness of RQR by more than 22% and 26%, respectively. Note that the improvements in the effectiveness and coverage are statistically significant at the 0.05 level using a student's t-test. The fact that the improvement in precision is not statistically significant is by our design decision, restricting our attention to an operating point that achieves similar precision to that of RQR. Note that there are operating points in which our models improve over the RQR baseline in both precision, coverage and effectiveness in a statistically significant way.

Table 5: Performance of QR models on voice queries relative to the RQR baseline. The best performance is boldfaced.

Method	E@1	P@1	Coverage
TDQR	-43.4%	-56.6%	$\geq$ **+22%**
VQR-4	+22.6%	+3.3%	+18.6%
VQR-6	**+26.5%**	**+3.6%**	+22%

When we dive deeper into the contribution of each analyzer towards the output of VQR-6, we notice that in 26.0% of the cases the analyzer whose alternative had the highest score was phonetic 4-grams, in 21.7% it was MLT, in 18.8% it was 3-grams, in 17.2% it was phonetic, in 12.3% it was 4-grams, and only in 4.0% it was full phonetic. From a precision point of view, the analyzer with the lowest precision is 3-grams. We consider its precision as a reference point. Then, full phonetic has a 2.5% better precision, MLT has a 6% better precision, phonetic has a 12.8% better precision, 4-grams has a 13.2% better precision, and phonetic 4-grams has a 19.6% better precision. Notice that the phonetic 4-grams alternatives have the highest traffic share and precision. This immediately raises the question how VQR-4 still maintains such high precision and effectiveness. It turns out that in many cases, the alternative with the highest score is also identified by other analyzers (with lower score). In VQR-4, the ranking model adjusts so that most of the traffic that phonetic 4-grams and 3-grams handled is spread between 4-grams (34.5%), MLT (32.0%), and phonetic (28.3%) with relatively small changes in precision.

Table 6 presents an evaluation of the TDQR and VQR-4 methods relative to the RQR baseline on the web null queries test set. Reported values are statistically significant with $p < 0.05$. Again, RQR significantly outperforms the effectiveness of TDQR in-spite of the significant coverage increase, and VQR-4 outperforms RQR in all metrics with a statistically significant improvement in the coverage and various effectiveness measures at the 0.05 level. We decided not to report the results for VQR-6 as they follow the same trends presented for the voice dataset, and thus, carry a light conceptual message.

One especially interesting observation relates to the performance differences of TDQR on voice and web data. We first note that the performance of RQR on the voice and web test sets is roughly the same, having no difference in coverage and only small increase of 3.9% in precision and effectiveness for the web data. So, essentially, our reference performance does not change between voice and web. However, we observe that the degradations in precision and effectiveness of TDQR over the voice test set is much more significant than over the web test set. We see a -43.4% and -56.6% decrease in E@1 and P@1 over voice compared to -15.3% and -34.8% decrease over web. This is a statistically significant difference, hinting that web queries are considerably different, and apparently easier to fix, than voice queries. This important insight adds to previous line of research [15, 17, 20, 22], highlighting differentiating factors between the voice and web domains, and supporting the need for specialized mechanisms for voice.

6 ONLINE EVALUATION

Variants of the proposed QR framework were evaluated for their effectiveness in handling null queries as part of an online A/B test on a commercial voice assistant. In that test, the control group experienced a RQR-like approach, while the treatment group experienced a VQR-4-like approach. The experiment ran for about a month. The A/B test demonstrated positive impact on customers, reducing null queries by 40.5%. While increasing the overall traffic coverage and providing customers with product offers, the quality metrics also demonstrated an improvement. The relevance of the queries handled by the QR system (P@1) increased by about 20%, as evaluated by internal annotators. Furthermore, the rate of positive actions (e.g., purchases) across the entire traffic increased by 1.7%, although the fraction of null queries was relatively small. All those

Table 6: Performance of QR models on web null queries relative to the RQR baseline. The best performance is boldfaced.

Method	E@1	P@1	Coverage	E_{max}@3	P_{max}@3	$EnDCG_3$	$nDCG_3$
TDQR	-15.3%	-34.8%	$\geq$ **+22.0%**	-13.2%	-33.2%	-14.6%	-34.3%
VQR-4	**+23.7%**	**+1.6%**	+22.0%	**+24.6%**	**+2.2%**	**+24.3%**	**+2.0%**

results were shown to be statistically significant at the 0.05 level using a student's t-test.

7 CONCLUSIONS

We presented a new framework for pre-retrieval query rewriting of voice shopping null queries. Our approach takes the characteristics of the voice-user interface into consideration. We conducted experiments with both voice and web data of a commercial voice assistant and an e-commerce website. Those experiments demonstrated that our approach outperforms several baselines by large margins in both offline and online settings. We also provided empirical evidence for a fundamental difference between voice null queries and web null queries, substantiating the use of specialized mechanisms for the voice domain. We believe that our proposed framework, mapping tail to head queries, is of independent interest as it can be extended and applied to other domains beyond voice shopping. As future work, we plan to explore more advanced mapping techniques. For example, one natural approach is to apply sequence-to-sequence deep learning techniques to learn a mapping from null queries to alternatives. It can be particularly valuable to work with phoneme representations of queries on top of textual ones. Another intention is to identify better ranking techniques. Since voice interfaces are new and not yet widely adopted, there is still a data sparsity issue that limits the ability to tackle those plans. There is a need to research ways to bypass this issue. One such approach may be to massively produce semi-supervised weak labels, e.g., use the relevance score that a search engine assigns to a (query, offer) pair to create labeled data for the ranking. Another promising approach is to use transfer learning from web to voice.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues Natali Arieli, Sagi Bernstein, Anna Chalupowicz, Anna Farberman, David Shamouilian, and Yochai Zvik for their fruitful collaboration, discussions, and help in implementation.

REFERENCES

[1] 2020. What is Elasticsearch? https://www.elastic.co/what-is/elasticsearch. [Online; accessed 5-January-2020].

[2] Cyril Allauzen, Edward Benson, Ciprian Chelba, Michael Riley, and Johan Schalkwyk. 2012. Voice query refinement. In *Thirteenth Annual Conference of the International Speech Communication Association.*

[3] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 1999. *Modern Information Retrieval.* ACM Press / Addison-Wesley.

[4] Michael Bendersky and W. Bruce Croft. 2008. Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 491–498.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[6] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. 2008. The Query-flow Graph: Model and Applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management.* 609–618.

[7] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. 2009. Query Suggestions Using Query-flow Graphs. In *Proceedings of the 2009 Workshop on Web Search Click Data.* 56–63.

[8] Francesco Bonchi, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. 2012. Efficient Query Recommendations in the Long Tail via Center-Piece Subgraphs.

[9] Daniele Broccolo, Lorenzo Marcon, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. 2012. Generating suggestions for queries in the long tail with an inverted index. *Information Processing & Management* 48, 2 (2012), 326 – 339.

[10] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 243–250.

[11] David Carmel and Elad Yom-Tov. 2010. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2, 1 (2010), 1–89.

[12] Fernando Diaz. 2016. Pseudo-Query Reformulation. In *Advances in Information Retrieval.* Springer International Publishing, 521–532.

[13] Sharad Goel, Andrei Broder, Evgeniy Gabrilovich, and Bo Pang. 2010. Anatomy of the Long Tail: Ordinary People with Extraordinary Tastes. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining.* 201–210.

[14] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context- and Content-aware Embeddings for Query Rewriting in Sponsored Search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 383–392.

[15] Ido Guy. 2016. Searching by Talking: Analysis of Voice Queries on Mobile Web Search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 35–44.

[16] Mohammad Al Hasan, Nish Parikh, Gyanit Singh, and Neel Sundaresan. 2011. Query Suggestion for E-commerce Sites. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining.* 765–774.

[17] Ahmed Hassan Awadallah, Ranjitha Gurunath Kulkarni, Umut Ozertem, and Rosie Jones. 2015. Characterizing and Predicting Voice Query Reformulation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* 543–552.

[18] Ben He and Iadh Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors. In *String Processing and Information Retrieval.* Springer Berlin Heidelberg, Berlin, Heidelberg, 43–54.

[19] Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to Rewrite Queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.*

[20] Amir Ingber, Arnon Lazerson, Liane Lewin-Eytan, Alexander Libov, and Eliyahu Osherovich. 2018. The Challenges of Moving from Web to Voice in Product Search. In *Proc. 1st International Workshop on Generalization in Information Retrieval.*

[21] Amir Ingber, Liane Lewin-Eytan, Alexander Libov, Yoelle Maarek, and Eliyahu Osherovich. 2018. Offline vs. Online Evaluation in Voice Product Search. In *1st International Workshop on Generalization in Information Retrieval.*

[22] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How Do Users Respond to Voice Input Errors? Lexical and Phonetic Query Reformulation in Voice Search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 143–152.

[23] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 133–142.

[24] Rosie Jones and Daniel C. Fain. 2003. Query word deletion prediction. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 435–436.

[25] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating Query Substitutions. In *Proceedings of the 15th International Conference on World Wide Web.* 387–396.

[26] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (Valencia, Spain). Association for Computational Linguistics, 427–431.

[27] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, and Wangyou Zhang. 2019. A Comparative Study on Transformer vs RNN in Speech Applications. *CoRR* abs/1909.06317 (2019).

[28] Giridhar Kumaran and Vitor R. Carvalho. 2009. Reducing long queries using query quality predictors. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 564–571.

[29] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 120–127.

[30] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, and Ravi Teja Gadde. 2019. Jasper: An End-to-End Convolutional Neural Acoustic Model. *CoRR* abs/1904.03288 (2019).

[31] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Ricardo Baeza-Yates, and Hongyuan Zha. 2017. Exploring Query Auto-Completion and Click Logs for Contextual-Aware Web Search and Query Suggestion. In *Proceedings of the 26th International Conference on World Wide Web*. 539–548.

[32] Saurav Manchanda, Mohit Sharma, and George Karypis. 2019. Intent Term Weighting in E-Commerce Queries. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2345–2348.

[33] Donald Metzler and W. Bruce Croft. 2007. Latent Concept Expansion Using Markov Random Fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 311–318.

[34] Donald Metzler, Susan Dumais, and Christopher Meek. 2007. Similarity Measures for Short Segments of Text. In *Advances in Information Retrieval*, Giambattista Amati, Claudio Carpineto, and Giovanni Romano (Eds.).

[35] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-Oriented Query Reformulation with Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 574–583.

[36] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *CoRR* abs/1904.08779 (2019).

[37] Lawrence Philips. 1990. Hanging on the Metaphone. *Computer Language Magazine* 7, 12 (December 1990), 39–44.

[38] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

[39] Milad Shokouhi, Rosie Jones, Umut Ozertem, Karthik Raghunathan, and Fernando Diaz. 2014. Mobile Query Reformulations. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1011–1014.

[40] Gyanit Singh, Nish Parikh, and Neel Sundaresan. 2011. User behavior in zero-recall ecommerce queries. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 75–84.

[41] Gyanit Singh, Nish Parikh, and Neel Sundaresan. 2012. Rewriting Null e-Commerce Queries to Recommend Products. In *Proceedings of the 21st International Conference on World Wide Web*. 73–82.

[42] Ning Su, Jiyin He, Yiqun Liu, Min Zhang, and Ma Shaoping. 2018. User Intent, Behaviour, and Perceived Satisfaction in Product Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 547–555.

[43] Zehong Tan, Canran Xu, Mengjie Jiang, Hua Yang, and Xiaoyuan Wu. 2017. Query Rewrite for Null and Low Search Results in eCommerce. In *Proceedings of the SIGIR Workshop On eCommerce*.

[44] Rong Xiao, Jianhui Ji, Baoliang Cui, Haihong Tang, Wenwu Ou, Yanghua Xiao, Jiwei Tan, and Xuan Ju. 2019. Weakly Supervised Co-Training of Query Rewriting andSemantic Matching for e-Commerce. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 402–410.

[45] Jinxi Xu and W. Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 4–11.

[46] Xiaobing Xue, Samuel Huston, and W. Bruce Croft. 2010. Improving verbose queries using subset distribution. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*. 1059–1068.

[47] Le Zhao and Jamie Callan. 2010. Term necessity prediction. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*. 259–268.