Query Rewrite for Low Performing Queries in E-commerce Based On Customer Behavior

Meng Zhao* Morgan White* Faizan Javed*

Abstract

Search query rewriting refers to the technology that generates alternative search queries which retrieve more results pertaining to the original user intent. In the domain of e-commerce in the home improvement industry, the lexical gap between customer query and indexed product information is among the top pain points impacting revenue perspective and hurting customer experience. To reduce customer friction and improve search relevancy, in this paper, we propose a novel query rewrite system that leverages large-scale in-session user behavior which increases recall without compromising the original query intent. The proposed system has two major components: 1) an embedding system at query level that generates query reformulations based on fast and precise semantic representation, 2) a token similarity system at character level that improves the robustness and practicality of the embeddings. Evaluations based on the stratified sampling of search queries show that the top-3 query rewrites generated by the system significantly improve customer engagements by an average of 40%. The beta version of the system has been applied in various natural language and query understanding projects with large potential revenue impact.

1 Introduction

An intriguing challenge in the home improvement domain is the vocabulary mismatch between product information and customer queries. Pertaining to operational standards, vendors use professional nomenclature to describe product titles and descriptions in our catalog. Customers, on the contrary, are commonly accustomed to unoriginal nomenclatures or queries with inadequate domain-specific information. For instance, the query *Tide pen* most commonly refers to the product *Tide To Go stain remover*. Since there is no *Tide pen* in the product catalog, products retrieved by our search engine will likely be either limited or irrelevant. See Table 1 for examples of lexical differences between customer query and product titles from the home improvement domain.

Based on our analysis, the vocabulary mismatch between customer queries and product terms could impact as many as 30% of our search traffic resulting in, as is called, *low performing queries* (LPQ). Specifically, we regard LPQ as queries for which click-through rate (CTR) is less than 20%, where the click-through rate is defined as

$$CTR = \frac{\#\text{ of clicks from impression}}{\#\text{ of unique sessions where query is searched}}.$$

Addressing LPQs increases CTRs and consequently has a positive impact on metrics such as ATC (Add-to-Carts) and Gross Demand (GD) which are important business metrics for online e-commerce operations.

Customer Query	**Product Title**
the thing under the chair	furniture slides
clog remover	drain auger
faucet connector	supply line
elbow	90-degree pipe fitting
sheetrock mud	joint compound

Table 1: Examples of Customer Queries And Corresponding Product Titles

While enriching ontologies or knowledge bases can close this gap [3], this approach may not be sustainable in practice as millions of products are onboarded and billions of unique search terms are conducted every year. Query rewriting techniques such as query relaxation [10] and query expansion [6] have been used to handle such vocabulary mismatch issues. These techniques remove or add tokens or phrases from a query. However, we argue that such operations are under the assumption that the original query is self-explanatory, or in other words, *correct*. To revisit the *Tide pen* example, it would be reasonable to presume that query relaxation would likely prioritize search engine executions on *pen*, or query expansion would tend to promote the term *Tide*. Therefore, the products retrieved would be either ballpoint pens or laundry detergents (or a combination), none of which could meet customer expectations.

It is worth noticing that customer behavior observed on our website shows that customers tend to focus on a single home remodeling project within a

*The Home Depot .Inc

given session. This is partly because home remodeling projects are generally non-trivial, both physically and financially, and thus require substantial product research and decision-making efforts. For example, a customer working on fixing a running toilet could easily spend hours searching and browsing products related to the objective. The concentration of customer intents within the same session produces a unique search characteristic such that in-session search queries are likely to share significant semantic homogeneity. Such characteristic is the primary motivation for the proposed methodology because modeling semantic relations of in-session queries could produce query rewrite candidates and at the same time, train domain-specific word embeddings. In this paper, we propose a novel solution of generating query rewrite candidates based on in-session customer behaviors. In what follows, we present more algorithmic details in terms of language models with corresponding mathematical justifications. Our main contributions are the following:

1. Quantifying search logs at query level in lieu of token level as in previous works,
2. Constructing neural network language models based on in-session query reformulations,

2 Related Works

Query rewrite systems can be traced back to the late 90s .COM boom when both Google and Yahoo started to experiment automated spell correction systems [30]. These early query rewrite systems adopted heuristic-based techniques that involve synonym replacement, hyper/hypo-nym extraction and/or morphology normalization [1,24,29]. See [4] for a comprehensive survey. In recent years, techniques that utilize probabilistic language models have become mainstream [9, 17, 26, 31].

Lately, query rewrite systems that tackle low or empty result queries for E-commerce applications have been on the rise. [30] focused on dropping and replacement at the token level based on probabilistic scoring models. [28] add taxonomy constraints to sub-queries and retrieve items from collections of historical purchases to solve a unique use case of information retrieval from ephemeral documents at eBay. [13] proposed query suggestions by their semantic similarities. [22] described a query segmentation method based on domain-specific frequency models. However, most existing literature imposes methodological concentrations on lexicon or token level, or focus on marketplace use cases. However, to the best of our knowledge, no previous work has addressed query rewrite challenges by modeling semantic relations of in-session queries for e-commerce search in the home improvement industry.

3 Methodology

3.1 The Query-Session Model. As aforementioned, we have observed significant homogeneity of customer intents within the same session. In fact, in the inspection of a random sample of 2K sessions of 2018, we noticed more than 60% of them contained at least two queries that could belong to the same intent. This is largely because most home remodeling projects research, estimation, comparison, and even validation, before final purchases. Consequently, due to the complexity of the decision-making process that is usually comprised of multiple steps or even phases, customers typically are only able to focus on one specific step or goal within a session. For example, to shop for a project to install wall tiles in a bathroom, customers would usually start by understanding specifications of tiles, followed by researching the designs and patterns of the tiles, then trying to collect other tools and materials for the installation before lastly studying related how-to guides. Each individual step or phase will most likely occupy the entire session during each visit. This is the key motivation that drives the modification of the standard language model to model queries based on in-session behavior, in lieu of tokens and sentences.

More specifically, given a search session, we jointly model the remaining tokens of each query after standard text cleaning procedures as documented in popular natural language processing (NLP) pipelines [19,20]. Lemmatization is only applied to high volume search terms to avoid false positive suggestions. All punctuation is excluded except for quotation marks (that represent foot and inch). Sessions thus can be modeled similarly as in a standard language model, only that words become queries in this case. We herein denote the method as the *query-session model*. There are two major and one minor advantage to the approach.

- Queries in each session can be regarded independent from each other, as opposed to tokens in n-gram or query segmentation researches [12] . Thus a uni-gram language model could be enabled for computational simplicity without sacrificing predictability.
- Since the topic of a given session is relatively identical, each individual query would be closely related to the context (or other queries) of the sessions, and hence can be used to predict other queries within the same sessions. This assumption enables methodological legitimacy of learning semantic relations by the skip-gram model [21].
- As a minor advantage, in contrast to queries that could be arbitrarily long, sessions are much shorter

in terms of the number of queries. This could result in several folds of magnitudes of complexity reduction for back-propagation as splits in Huffman trees [25] could be reduced significantly. Practically, we have noticed a 20-30% improvement in training time.

Formally, we adopt in our approach a *query-session* model that models the probability of an array of queries in a session. Denote a *clickstream* sample as $\mathcal{C}$, $s_j(q_i)$, or s_j for short, as session j from $\mathcal{C}$ where q_i has been searched for. By the chain rule of conditional probability [27], we know

$$P(q_1, \ldots, q_n) = \prod_{k=1}^{N} P(q_k | q_0, \ldots, q_{k-1}). \tag{3.1}$$

where $N = |\{q_k : q_k \in s_j\}|$. By independence, Eq. (3.1) simplifies to

$$P(q_1, \ldots, q_n) = \prod_{i=1}^{n} P(q_i).$$

Hence,

$$P(q_1, \ldots, q_n | q_i) = \prod_{k \neq i} P(q_k),$$

or,

$$P(s_j(q_i) | q_i) = \prod_{k \neq i} P(q_k). \tag{3.2}$$

Eq. (3.2) defines the query-session model as part of our query rewrite system.

3.2 Query Embedding. In order to generate query rewrite candidates, we start by training query embedding vectors such that similar queries will have similar vectors based on the query-session model. The proposed approach is more robust than in-session query reformulation [14] because cross-session customer behaviors are jointly modeled through neural-network language models [2]. We argue that an effective query rewrite system can be built by approximating the population of query variations because customer behaviors stabilize as *clickstream* [16] samples become large enough.

We employ the skip-gram model and follow standard treatments to parameterize the query-session model. Consider conditional probabilities $P(s_j | q_i; v_i)$, where v_i denotes the vector representation of query i in the search space. Combining [11] and [25], write

$$\arg\max_{v_i} \prod_{(s_j, q_i) \in \mathcal{C}} P(s_j | q_i; v_i), \tag{3.3}$$

as the objective function for the embedding process. By Eq. (3.2), Eq. (3.3) simplifies to

$$\arg\max_{v_i} \prod_{k=1}^{V} \frac{\exp(v_k^T \cdot v_i)}{\sum_{k'=1}^{N} \exp(v_{k'}^T \cdot v_i)}.$$

Or,

$$\arg\max_{v_i} \sum_{k=1, k \neq i}^{V} \left[v_k^T \cdot v_i - \log\left(\sum_{k'=1}^{N} \exp(v_{k'}^T \cdot v_i) \right) \right], \tag{3.4}$$

after log-transformation, where N is defined as in Eq. (3.1) and $V = |\{q_{k'} : q_{k'} \in \mathcal{C}\}|$, respectively.

To improve training efficiency and allow negative sampling [21], we modify Eq. (3.4) as

$$\arg\max_{v_i} \left[\sum_{k=1, k \neq i}^{V} \log \sigma\left(v_k^T \cdot v_i\right) + \sum_{k'=1, k' \neq i}^{V'} \log \sigma\left(-v_{k'}^T \cdot v_i\right) \right], \tag{3.5}$$

where V' is the size of negative samples of $(s_{j'}, q_i)$, and $\sigma(x)$ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

3.3 Low Performing Query Improvement. Given an LPQ, top 3 query rewrite candidates are generated based on the multiplicative combination objective as suggested by [18]. Results are pre-calculated and cached in an in-memory database such as Redis [5] with daily re-indexing. For production viability and latency thresholding, we only process queries that are searched 10 times or more on an annual basis, excluding product ID or model number searches. The vector size is limited to 200. On average, we have observed a 200% increase in CTR before and after suggested rewrites in an evaluation dataset hand picked by business stakeholders. See Table 2 for an example of recommended query rewrites.

LPQ	**Rewrite**	**Percent of Increase in CTR**
big fans	heavy duty fans	300%
weed eater part	gas trimmer parts	800%
kitchen area rug	kitchen runners	500%
coffee pots	coffee makers	300%
wireless outdoor speaker	bluetooth outdoor speaker	200%

Table 2: Examples of Query Rewrites for LPQ

4 Evaluation

Evaluation of the proposed query rewrite system is conducted using domain expert validation and random sampling. Moreover, we evaluate the query rewrites for selected categories of high traffic and revenue impact for prioritization purposes. This strategy is mainly for the purpose of business prioritization, as revenue contributions of product categories vary substantially. Our evaluation plan in a nutshell is stratified sampling of queries by business impact targeting mean average precision (MAP).

4.1 Evaluation Metric. We choose **Precision@k** or **P@k** as the primary evaluation metric, given a threshold θ_0. Given an LPQ, for all query rewrites with similarity distance greater than or equal to the presumed threshold, we calculate the proportion of the top k rewrites that are relevant, where relevancy is generally considered as retaining the same customer intent as the original query. For instance, for the query *lantern oil*, rewrite candidate *lantern fuel* is considered relevant, whereas *fuel* or *lantern* are not. Relevancy is judged by a combination of expert opinions and expert systems.

Mathematically, **P@k** is defined as

$$\textbf{P@k} = \frac{\text{\# of relevant query rewrites @k}}{\text{\# of query rewrites @k}},$$

where k is set as 3 to balance recall and latency.

4.2 Test Data Preparation. Input queries with CTR of 20% or less are first allocated into M categories based on an in-house query classifier. A simple random sampling is applied to each category to estimate **P@3**. For each category, MAP is used as the final metric.

Denote p_i as the population MAP of category i, and $\hat{p}_i$ as the corresponding sample MAP, we introduce error rate h_i and confidence level C_i, such that

$$P\left(\left|\frac{\hat{p}_i - p_i}{p_i}\right| < h_i\right) = C_i. \tag{4.6}$$

In our design, we set $h_i \in [0.01, 0.05]$ and $C_i \in [0.95, 0.99]$, depending on the revenue impact of the individual category. Note that the sampling approach is designed to ensure that C_i of the time the sampled proportion is not more than h_i different from the population proportion [7].

To determine sample size, let N_i be the size of category i, we calculate sample size n_i as

$$\begin{cases} n_{i0} = \dfrac{z^2_{\alpha_i}(1-p_i)}{h_i^2 p_i} \\ n_i = \dfrac{n_{i0}}{1 + (n_{i0}-1)/N_i} \end{cases}, \tag{4.7}$$

where $\alpha_i = (1 - C_i)/2$, and z_{α_i} is the upper $\alpha_i/2$ quantile of the standard normal distribution such that

$$P(Z < z_{a_i}) = 1 - \frac{\alpha_i}{2}, \quad Z \sim N(0,1).$$

4.3 Evaluation Results. To conduct the evaluation, we regard one year of search log as the targeted population. To make the evaluation more commercially valuable, we separated the population into three groups based on search volume and revenue impact. Those groups are called herein Head, Medium and Tail. For each group, we increase sample size by 1K at a time until the desired confidence level is reached. Given a query, the suggested query rewrite is considered positive if

- The rewrite is semantically similar.
- The rewrite has at least 40% more click-through than the original.

Specific population, sample sizes for each group, and corresponding metrics are presented in Table 3. Results shown are for selected high revenue categories. Note that population sizes are estimated based on aggregation of search logs of the past two years. Moreover, threshold for similarity distance is set at 0.85. Category D is of a relatively lower average precision mostly because that it is a newly introduced category with much less customer behavior data. Otherwise, MAP values of around or over 0.9 from the system generate statistically and commercially viable results.

Category	Population Size	Sample Size	MAP@3	Conf.
A	5M	1.6K	0.94	95%
B	6.8M	1K	0.95	95%
C	7M	1K	0.89	95%
D	10M	1K	0.85	95%

Table 3: LPQ Rewrite Evaluation Results

5 Conclusions and Future Work

We developed and evaluated a novel query rewrite system for generating semantically close alternatives of user input for improved precision and recall to promote click-through and eventually sales. The system is able to handle the vocabulary mismatch issue and improve LPQs and CTR metrics. By training the session-query embedding properly to maintain similar (intent) queries within closer proximity.

The system is designed and fine-tuned toward high volume and high revenue queries that are mostly short,

broad and ambiguous. Because of the correct assumptions made regarding the query-session embedding component, the system is able to support the massive majority of search traffic and search conversions (actual percentages excluded from the paper) of the e-commerce division of the home remodeling business that we support. By stratified sampling based evaluation of major categories based on revenue impact, the system is able to achieve close to 90% precision on average on MAP@3.

We exposed the query rewrite system as both RESTful service and standalone word embeddings to support multiple NLP tasks across major lines of business here from query understanding to customer service. As the search space expands, we plan to handle the growing vocabulary size by a similar hashing trick as in fastText [15]. In addition, the domain-specific word embedding trained from the second component of the proposed system needs more evaluation and benchmarking against popular embeddings like ELMo [23] or BERT [8]. We will address those tasks in more details in future work.

References

[1] J. Bai, J.-Y. Nie, and G. Cao, *Context-dependent term relations for information retrieval*, in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, Stroudsburg, PA, USA, 2006, Association for Computational Linguistics, pp. 551–559.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, *A neural probabilistic language model*, J. Mach. Learn. Res., 3 (2003), pp. 1137–1155.

[3] J. Bhogal, A. Macfarlane, and P. Smith, *A review of ontology based query expansion*, Inf. Process. Manage., 43 (2007), pp. 866–886, https://doi.org/10.1016/j.ipm.2006.09.003.

[4] J. Bhogal, A. MacFarlane, and P. J. Smith, *A review of ontology based query expansion*, Inf. Process. Manage., 43 (2007), pp. 866–886.

[5] J. L. Carlson, *Redis in Action*, Manning Publications Co., Greenwich, CT, USA, 2013.

[6] C. Carpineto and G. Romano, *A survey of automatic query expansion in information retrieval*, ACM Comput. Surv., 44 (2012), pp. 1:1–1:50, https://doi.org/10.1145/2071389.2071390.

[7] W. G. Cochran, *Sampling Techniques, 3rd Edition.*, John Wiley, 1977.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[9] F. Diaz, B. Mitra, and N. Craswell, *Query expansion with locally-trained word embeddings*, ArXiv, abs/1605.07891 (2016).

[10] T. Gaasterland, *Cooperative answering through controlled query relaxation*, IEEE Expert: Intelligent Systems and Their Applications, 12 (1997), pp. 48–59, https://doi.org/10.1109/64.621228.

[11] Y. Goldberg and O. Levy, *word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method.*, CoRR, abs/1402.3722 (2014).

[12] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam, *Query segmentation revisited*, in Proceedings of the 20th International Conference on World Wide Web, WWW '11, New York, NY, USA, 2011, ACM, pp. 97–106, https://doi.org/10.1145/1963405.1963423.

[13] M. A. Hasan, N. Parikh, G. Singh, and N. Sundaresan, *Query suggestion for e-commerce sites*, in Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, New York, NY, USA, 2011, ACM, pp. 765–774, https://doi.org/10.1145/1935826.1935927.

[14] J. Huang and E. N. Efthimiadis, *Analyzing and evaluating query reformulation strategies in web search logs*, in Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, New York, NY, USA, 2009, ACM, pp. 77–86, https://doi.org/10.1145/1645953.1645966.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, *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, Association for Computational Linguistics, April 2017, pp. 427–431.

[16] R. Kohavi, *Mining e-commerce data: the good, the bad, and the ugly*, in KDD, 2001.

[17] S. Kuzi, A. Shtok, and O. Kurland, *Query expansion using word embeddings*, in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, New York, NY, USA, 2016, ACM, pp. 1929–1932, https://doi.org/10.1145/2983323.2983876.

[18] O. Levy and Y. Goldberg, *Linguistic regularities in sparse and explicit word representations*, in In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL, 2014.

[19] E. Loper and S. Bird, *Nltk: The natural language toolkit*, in Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, ETMTNLP '02, Stroudsburg, PA, USA, 2002, Association for Computational Linguistics, pp. 63–70, https://doi.org/10.3115/1118108.1118117.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, *The Stanford CoreNLP natural language processing toolkit*, in Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.

[21] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in Proceedings of

the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, USA, 2013, Curran Associates Inc., pp. 3111–3119.

[22] N. Parikh, P. Sriram, and M. A. Hasan, *On segmentation of ecommerce queries*, in CIKM, 2013.

[23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep contextualized word representations*, in Proc. of NAACL, 2018.

[24] Y. Qiu and H.-P. Frei, *Concept based query expansion*, in Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '93, New York, NY, USA, 1993, ACM, pp. 160–169, https://doi.org/10.1145/160688.160713.

[25] X. Rong, *word2vec parameter learning explained*, arXiv preprint arXiv:1411.2738, (2014).

[26] D. Roy, D. Paul, M. Mitra, and U. Garain, *Using word embeddings for automatic query expansion*, ArXiv, abs/1606.07608 (2016).

[27] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd ed., 2009.

[28] G. Singh, N. Parikh, and N. Sundaresan, *Rewriting null e-commerce queries to recommend products*, in Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion, New York, NY, USA, 2012, ACM, pp. 73–82, https://doi.org/10.1145/2187980.2187989.

[29] M. Song, I.-Y. Song, X. Hu, and R. B. Allen, *Integration of association rules and ontologies for semantic query expansion*, Data Knowl. Eng., 63 (2007), pp. 63–75, https://doi.org/10.1016/j.datak.2006.10.010.

[30] Z. Tan, C. Xu, M. Jiang, H. Yang, and X. Wu, *Query rewrite for null and low search results in ecommerce*, in eCOM@SIGIR, 2017.

[31] H. Zamani and W. B. Croft, *Embedding-based query language models*, in Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR '16, New York, NY, USA, 2016, ACM, pp. 147–156, https://doi.org/10.1145/2970398.2970405.