Knowledge Graph-Enhanced Neural Query Rewriting

Shahla Farzana*
sfarza3@uic.edu
University of Illinois Chicago
Chicago, USA

Qunzhi Zhou
eBay Inc.
San Jose, USA
qunzhou@ebay.com

Petar Ristoski
eBay Inc.
San Jose, USA
pristoski@ebay.com

ABSTRACT

The main task of an e-commerce search engine is to semantically match the user query to the product inventory and retrieve the most relevant items that match the user's intent. This task is not trivial as often there can be a mismatch between the user's intent and the product inventory for various reasons, the most prevalent being: (i) the buyers and sellers use different vocabularies, which leads to a mismatch; (ii) the inventory doesn't contain products that match the user's intent. To build a successful e-commerce platform it is of paramount importance to be able to address both of these challenges. To do so, query rewriting approaches are used, which try to bridge the semantic gap between the user's intent and the available product inventory. Such approaches use a combination of query token dropping, replacement and expansion. In this work we introduce a novel Knowledge Graph-enhanced neural query rewriting in the e-commerce domain. We use a relationship-rich product Knowledge Graph to infuse auxiliary knowledge in a transformer-based query rewriting deep neural network. Experiments on two tasks, query pruning and complete query rewriting, show that our proposed approach significantly outperforms a baseline BERT-based query rewriting solution.

CCS CONCEPTS

• **Applied computing** → **Electronic commerce**;

KEYWORDS

eCommerce, Query Rewriting, Knowledge Graph

ACM Reference Format:
Shahla Farzana, Qunzhi Zhou, and Petar Ristoski. 2023. Knowledge Graph-Enhanced Neural Query Rewriting. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3543873.3587678

1 INTRODUCTION

Efficient and effective information retrieval is crucial for the success of any e-commerce platform. The main task is to retrieve the most relevant product listings among millions of listings that correctly match the user's intent. However, often the user intent cannot be directly matched to the product inventory, either caused by semantic mismatch or missing inventory. This is the main cause for customer churn and loss of revenue [Tan et al. 2017; Wang et al. 2021]. To address these challenges, Query Rewriting (QR) approaches are applied to reformulate the user query to increase the number of matched product listings and retain the relevance of the results with respect to the original user query. QR is of paramount importance to retain users on the platform, and increase conversion and click through rates even in the case of missing inventory.

In recent years, plethora of approaches for QR in e-commerce have been proposed. Majority of approaches are using Seq2Seq models to generate new query rewrites based on a source user query [Qiu et al. 2021]. However, as shown by Zhang et al. [Zhang et al. 2022] many of the existing approaches fail to correctly understand the shopping user intent, leading to sub-optimal query rewrites.

To address this challenge, we introduce a Knowledge Graph (KG)-enhanced neural query rewriting approach. Such an approach allows to perform full semantic understanding of the query, and correctly identify the user intent, leading to high-quality query rewrites. To do so, we are using a relationship-rich product KG, generated from millions of product listings in our inventory. The KG is a weighted directed graph, which models entities and relations describing product listings, e.g., brands, colors, materials, sizes etc. Such a KG allows us to perform semantic understanding of a query and correctly capture the user intent. We use a proprietary entity linking approach [Zhou et al. 2021]. Once the entities are identified, we can explore the graph to draw additional information about each entity and analyze the relations to other entities in the graph.

For example, given the query "Kobe Bryant Leather Sneaker Size 10", we first identify the query category in our inventory[1], and we pull the corresponding KG for that category, i.e., "Mens Athletic Shoes". As shown in Figure 1, we are able to link "Kobe Bryant" to the corresponding entity of the type "Product Line", the token "Leather" is linked to an entity of type "Material", etc. Furthermore, as our KG is built from our own inventory we are able to calculate the frequency for each of these entities, which is a direct signal of what is the maximal recall set we could expect for the given user query. In this example, the result set of the query is limited by the entity "product_line/kobe_bryant" to 136 product listings. A simple solution to increase the result set is to rewrite the query by either replacing or dropping this entity from the query. The KG allows us to easily identify a replacement for the entity by exploring the relations to other entities in the graph. For example, we can replace the product line with its brand, "brand/nike", or by the type of the shoes "type/athletic_shoes", or identify similar product line entities, such as "product_line/lebron_james". Such similar entities

*Work done during internship with eBay.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ISIR-eCom 2023, May 01, 2023, Austin, TX
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3543873.3587678

[1]Query categorization is preformed using proprietary model and it is out of scope for this work

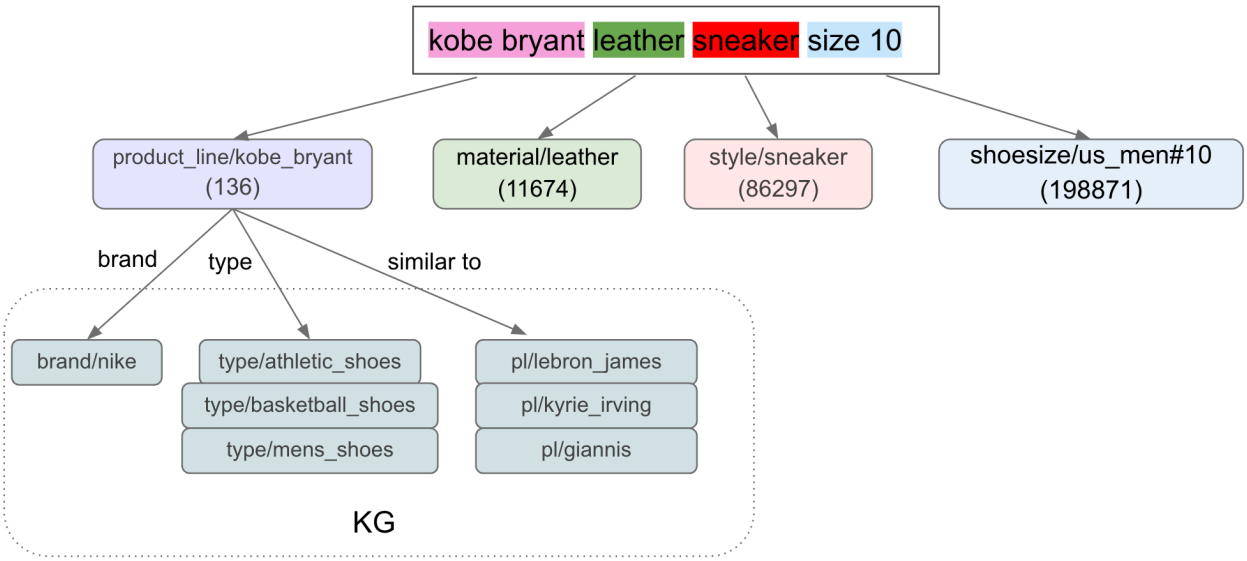

Figure 1: KG-based semantic query understanding.

could be identified by mining the relations in the graph, or using entity embedding approaches. Using the KG we are able to quickly generate many query rewrites, which retain the original user intent, and significantly increase the number of retrieved product listings.

To systematically incorporate such structured knowledge at scale, we enhance traditional query rewriting models by injecting auxiliary vectors extracted from the product KG. First, we introduce a KG-enhanced query pruning model, which performs only token dropping for query rewriting, i.e., identify the tokens in the user query with least importance and remove them in order to increase the recall set. Second, we introduce a KG-enhanced encoder-decoder model, which performs full query rewriting, by deleting, replacing and/or inserting tokens. Both models are enhanced with entity embedding vectors, generated from the product KG, entity types retrieved from Named-Entity Recognition model, category information, and entity frequency information. All these auxiliary vectors provide direct signals to the query rewriting models what is the best segment to be dropped or replaced, and what other segment should be replaced with.

Our contribution can be summarized as follows:

- KG-enhanced query pruning model, which is an enhanced version of a transformer-based token classification model by injecting auxiliary information extracted from a Product Knowledge Graph.
- KG-enhanced encoder-decoder complete query rewriting model, which is an enhanced version of a transformer-based encoder-decoder model by injecting auxiliary information extracted from a Product Knowledge Graph.

The rest of this paper is structured as follows. In Section 2, we give an overview of related work. In Section 3, we introduce our approach for building a product knowledge graph and two query rewriting approaches. In Section 4, we present in-depth evaluation of our models. We conclude with a summary and an outlook on future work.

2 RELATED WORKS

Query reformulation has been implemented to bridge the vocabulary mismatch in e-commerce search [Qiu et al. 2021] as well as in question answering (QA) systems to handle ambiguity of the follow-up questions by rewriting user query such that they can be processed by existing QA models as standalone questions outside of the conversation context [Vakulenko et al. 2021]. A subset of these works rely on two-phase system. In the first phase, candidate queries are generated using a combination of query token dropping, replacement and expansion. In the second phase, the candidates generated from the first phase are being ranked using a ranking model. Some research show that dropping unimportant terms in long queries [Chen and Zhang 2009; Tan et al. 2017] would be helpful to generate candidate queries. Other approaches include the utilization and creation of search datasets [He et al. 2016; Xiao et al. 2019; Zhang et al. 2022], and part-of-speech prediction [Tan et al. 2017]. The ranking task in second phase utilises TF-IDF weight, co-training [Xiao et al. 2019], and pre-trained language models [Lu et al. 2021; Tan et al. 2017]. Another type of approaches for query rewriting is end-to-end training using neural query generation models. This has been proved efficient and effective to generate query suggestion and rewritten queries, which usually leverages beam search, with top-n sampling decoding stage. For instance, sequence-to-sequence model for session-based query suggestion incorporating copy mechanism in decoding stage shows promising performance in generating query suggestions based on previous queries of the session [Dehghani et al. 2017].

In context of e-commerce search engine, studies show that 50% of the queries take part in a reformulation session and lead to a

rise in both click and purchase rates for the last query [Hirsch et al. 2020]. Although query rewriting is a crucial part for e-commerce search engine, rewriting involves many intrinsic difficulties, especially the lack of high quality query rewriting logs, contextual information for the query. To overcome the scarcity of high quality query rewriting logs in e-commerce search, end-to-end training of neural machine translation model leverages query titles from user click logs [Qiu et al. 2021] instead of query rewrite logs. Moreover, several approaches has been proposed regarding query reformulation in e-commerce search engine: enhancing query rewriting using query annotation (NER tags) as contextual knowledge [Wang et al. 2021], synonym dictionary [Mandal et al. 2019], entity tags, Parts of Speech (POS) tags and user behavior mining for query term dropping and replacement [Tan et al. 2017], contextual-term weighting for recognising important term aligned with query intent [Manchanda et al. 2019]. In voice search systems, retrieval based query reformulation has been proved effective mitigating the errors originating from Automatic Speech Recognition (ASR) system and Natural Language Understanding (NLU) pipeline [Chen et al. 2020; Yuan et al. [n.d.]].

Knowledge Graphs (KGs) constructed based on both input data and beyond the input query can be used to enhance text generation system like question answering, document summarisation, commonsense reasoning, and creative writing [Fan et al. 2019]. Incorporating KGs to generate answers with grounded facts in end-to-end diallog systems has been picked up in recent years. KG-copy [Chaudhuri et al. 2019] mechanism, a model that learns KG embeddings per dialog instead of globally, has shown promising performance both in goal and non-goal oriented dialog generation task producing knowledge grounded responses compared to the other memory network based encoder-decoder model and KG based generative models [Kassawat et al. 2019; Madotto et al. 2018]. A followup work [Chaudhuri et al. 2021] of KG-copy mechanism [Chaudhuri et al. 2019] leveraging the KG entities and relations along with pre-trained transformers outperformed the previous KG incorporated models [Chaudhuri et al. 2019; Madotto et al. 2018] in goal and non-goal oriented dialog generation in most automated and human generated metrics. In context of query rewriting, user interaction graph by mining their queries and learning query embeddings by leveraging the Graph Representation Learning has given performance boost in query rewriting over retrieval based systems [Yuan et al. [n.d.]]. A pre-trained model called Geo-BERT [Liu et al. 2021], combining geographic granularity knowledge graph of point of interests (POIs) with textual semantics shows strong performance in QR task for search in map service.

To the best of our knowledge, there is no end-to-end approach that is backed by a rich Product Knowledge Graph to perform query rewrites in the e-commerce domain. Our approach differs from the above methods in that we train transformer based query pruning and query rewriting models, which are capable of leveraging entity embeddings from industrial scale structured product knowledge graph.

3 METHODOLOGY

In this section, we present a KG-enhanced neural query reformulation framework that aims to provide a query-to-query reformulation solution by integrating auxiliary vectors extracted from domain knowledge. We present two solutions: (i) KG-enhanced query pruning model, and (ii) KG-enhanced Encoder-Decoder model for full query rewrites.

3.1 Problem Definition

Query rewriting is the process of modifying a source query consisting of n tokens $q_s = \{w_{s1}, w_{s2}, .., w_{sn}\}$, with result set recall r_s and result set relevance rel_s, to a target query consisting of m tokens $q_t = \{w_{t1}, w_{t2}, .., w_{tm}\}$, with result set recall r_t and result set relevance rel_t with respect to the source query, where the recall of the target query r_t is greater than the recall of the source query r_s, and the relevance of the target query result set rel_t is greater or lower, within acceptable threshold, than the relevance of the original query result set.

3.2 Product Knowledge Graph

External knowledge bases contain a plethora of cross-domain data which is highly precise, but lack coverage for specific e-commerce applications. To address this issue, an approach that leverages inventory data is employed. Most e-commerce platforms have rich data from sellers, such as item titles, descriptions, images, and aspect-value pairs. In this work, we use a product Knowledge Graph consisting of entities and relationships between them to model product inventory. The KG is mined from millions of product listings based on co-occurring aspect-value pairs in product listings, resulting in a directed weighted graph. More precisely, we generate a node in the graph for each aspect-value pair that occurs in product listings above a user specified threshold. To set the edge weights, for each co-occurring pair of aspect-value pair in at least one product listing, a normalized co-occurrence frequency is calculated. The co-occurrence frequency is normalized by the occurrence of both nodes in each direction, resulting in directed weights. Such weights give higher relevance to aspect-value pairs that co-occur more often together, normalized by their global popularity. For example, for the co-occurring aspect-values *Brand:Apple* and *Color:Sierra Blue*, we will generate the following edges in the graph:

```
Brand:apple p:color Color:sierra_blue 0.01

Color:sierra_blue p:brand Brand:apple 0.99
```

The fourth element is the weight of the edge, and in this case indicates that the color *Sierra Blue* is almost fully conditioned on the brand *Apple*, while the other direction is not significant.

To ease the use of such KG in downstream tasks, we use the approach of biased walks to embed all entities and relations. Using RDF2vec [Ristoski and Paulheim 2016], we perform biased walks on the weighted graph to flatten the graph in sequences that can later be embedded by any of the existing language models. This imparts a locality as well as some global contextual information to the nodes across the graph. This approach is able to capture the neighborhood of each entity in a single vector, which then can be

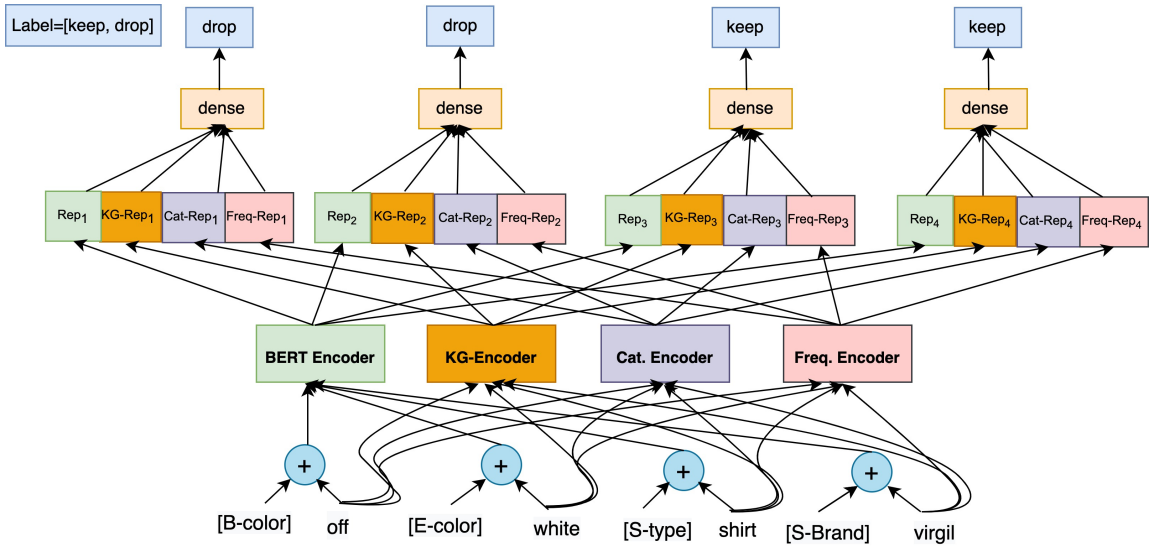

Figure 2: KG-enhanced query pruning model, with example input query "off white shirt virgil", labeling tokens "off" and "white" to be dropped.

used for similarity calculation or context inference. Such embeddings can then be ingested in various machine learning models to solve a variety of downstream tasks, in this case query rewriting.

3.3 KG-Enhanced Query Pruning Model

The objective of the KG-enhanced query pruning model is to identify and remove the tokens in the source query, which are least relevant and are causing low recall set. We define the task as a token-classification task, where each token in the source query is either *kept* or *dropped*. The KG-enhanced query pruning model is based on a standard BERT transformer token classification model, augmented with 4 additional inputs for each token: (i) NER vector N; (ii) KG n-dimensional entity embedding vectors K, which embeds the entity semantics. For multi-token entities, we assign the same entity vector to all tokens; (iii) category vector C of dimension 1, referring to the query category, and provides the category context for query rewriting; (iv) frequency vectors F of dimension 1 which refers to the frequency of each entity in the inventory and is a good indicator of the recall of an item in the inventory. The architecture of the KG-enhanced query pruning model is depicted in Figure 2.

Following the work on integrating keyword annotation with transformer embedding layer [Wang et al. 2021], we inject the NER vector in our BERT-encoder embedding layer, i.e., concatenating NER embedding with the token and positional embedding vectors, resulting in vector $X_i, i = 1, 2, ..., , n$, where n is the number of tokens in input query.

To be able to aggregate the rest of the entity auxiliary vectors with the BERT encoded vectors, we learn a transformation to project the vectors in the BERT vector space, following the approach presented in [Faldu et al. 2021]. More precisely, given an entity auxiliary vector, we train a two layer feed-forward network, using ReLU and TanH activation functions. The transformation functions for the entity auxiliary vectors are:

$$y_{x\_aux} = TanH(W_E^2 * ReLU(W_E^1 * x_{aux})) \quad (1)$$

where W_E^1 and W_E^2 are trainable weights for transforming the auxiliary entity vector. We apply the same transformation for all three auxiliary vectors, KG entity embedding K, category vector C, and the frequency vector F, which is normalized for each batch. The final token vector for token t_n and BERT vector X is the concatenation of all the vectors together:

$$\hat{X}_n = X_n \oplus K_n \oplus C_n \oplus F_n \quad (2)$$

The concatenated vector is fed into the token classification layer which is trained to predict the label *keep* or *drop* of each token. To train the model we label all tokens that are missing in the target user query as positive, i.e., labeled as *drop*.

3.4 KG-Enhanced Encoder-Decoder Query Rewriting Model

We introduce a KG-enhanced encoder-decoder architecture for complete query reformulation, including token dropping, replacement and insertion. The encoder model encodes an input sequence in a single vector, which is then passed to the decoder and used for generating new sequence conditioned on the input encoded vector. To train the model, we use the source query q_s on the input, and the target query q_t on the output, where the objective is to recreate the target query.

The model is based on the standard transformer-based encoder-decoder architecture [Rothe et al. 2020], extended with KG auxiliary vectors, as shown in Figure 3.

The embedding layer of the encoder block is coupled with the NER tags as explained in Section 3.3. Similarly as in the query pruning model, the encoder block takes the encoded representation of the category vector C, the frequency vector F, the KG embeddings

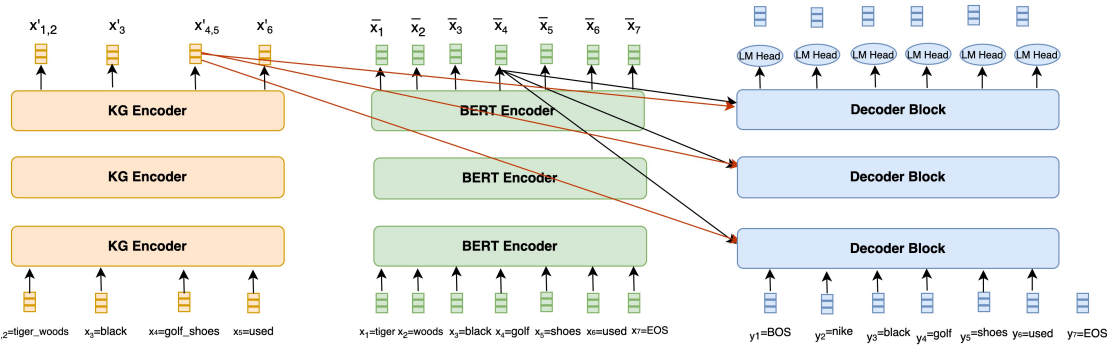

Figure 3: KG-enhanced encoder-decoder query rewriting model

vector K and concatenates them with the BERT encoded representation of each token. The conditional probability of entire target vector sequence by the decoder can be represented as follows:

$$p_{\theta_{dec}}(Y_{1:m}|X_{1:n}) = \prod_{i=1}^{m} p\theta_{dec}(y_i|Y_{0:i-1}, \hat{X})$$
$$\hat{X} = X \oplus K \oplus C \oplus F \tag{3}$$

After the model is trained, we use beam search to generate the top-N most probable target queries, conditioned on the encoded vector.

4 EXPERIMENTS

We evaluate both KG-enhanced models on 2 datasets, and compare the results to their baseline counterparts.

4.1 Datasets

We compile the datasets from user search logs from eBay Inc., one of the biggest e-Commerce platforms in the US. The datasets consist of source and target user query pairs. To identify such query pairs, we track user sessions in which a user first issued a query, the source query q_s, which matched less than 100 results, and the user didn't click on any item, then within the same session the user reformulated the query, the reformulated query q_t, and then clicked and/or purchased some of the results. Furthermore, we apply the following filters to improve the quality of the query pairs; (i) queries must be between 2 and 25 tokens; (ii) The token-based Jaccard distance between q_s and q_t must be above 20%, to ensure that the user retains the same shopping intent; (iii) The edit distance between q_s and q_t must be above 5, in order to avoid typographical error fixes;[2] (iv) The recall of q_t must be larger than the recall of q_s; (v) Both queries must belong to the same category; (vi) The user must engage with the result set of the target query, by either clicking or purchasing an item. For the query pruning model, we make sure that the target query is subsequence of the source query, where all the missing tokens are labeled as the positive class. For the encoder-decoder model, we allow token deletions, insertions and replacements. We generated 2 datasets for the task of query pruning, and 2 datasets for the task of full query rewriting. All datasets are generated from different time periods in the year 2022. The datasets have different sizes to analyze if users searching patterns change over time, and how well the models can adapt to those changes. The datasets statistics are shown in Table 1.

Table 1: Datasets statistics.

Dataset	Training	Test	Validation
$Query\ Pruning_{small}$	562,053	43,715	16,389
$Query\ Pruning_{large}$	2,863,026	222,679	95,434
$Query\ Rewrite_{small}$	511,768	39,804	17,060
$Query\ Rewrite_{large}$	1,038,088	80,740	34,604

The KG used in the experiments is built on top of around 50,000 categories of product listings, and contains tens of millions of nodes and edges.[3] To link the queries to the KG we use proprietary entity linking pipeline [Zhou et al. 2021]. The KG entity embeddings are trained using RDF2vec [Ristoski and Paulheim 2016], resulting in 100-dimensional entity embedding vectors. The NER tags are obtained from a proprietary NER model, which is part of the entity linking pipeline and is able to identify hundreds of named entity types and annotate them using BIOES tagging format [Zhou et al. 2021].

4.2 Evaluation Setup

Both the KG-enhanced query pruning and encoder-decoder model use a proprietary pre-trained BERT encoder model trained on several billion product titles from our inventory, user queries and public text corpora. We compare the KG-enhanced query pruning model to a baseline transformer-based token-based classifier using the same BERT encoder. Both models are trained for 5 epochs, and we tune the class weight in order to address the class imbalance between kept and dropped tokens, i.e., the dropped tokens are assigned higher weight, which is directly used in the loss function. We report Precision, Recall and F-score, and query accuracy, which measures how many of the predictions have exact overlap with the user target query. We perform the Wilcoxon signed-rank test to identify if there is statistical significant difference between the results of the baseline approach and our proposed approach [Demšar

[2]Spelling mistakes are handled by a separate proprietary spelling correction model.

[3]Exact statistics are company's proprietary information.

Table 2: Query Accuracy, Precision, Recall and F-Score results, for the baseline BERT model, and the KG-enhanced model, on the small and large Query Pruning datasets. * indicates a statistical significant increase over the baseline, for a confidence level $p < 0.01$.

Model	Query Accuracy	Precision	Recall	F-Score
$Baseline_{small}$	54.02%	66.21%	67.53%	68.36%
$KG-enhanced_{small}$	**55.43%***	**70.35%***	**69.59%***	**69.97%***
$Baseline_{large}$	44.00%	61.81%	59.27%	60.51%
$KG-enhanced_{large}$	**46.26%***	**62.50%***	**61.67%***	**62.09%***

Table 3: Recall@5, Jaccard@5 and BLEU@5 results, for the baseline BERT model, and the KG-enhanced model, on the small and large Token Replacement datasets.

Model	Recall@5	Jaccard@5	BLEU@5
$Baseline_{small}$	3.03%	43.82%	35.76%
$KG-enhanced_{small}$	**3.32%**	**44.19%**	**36.25%**
$Baseline_{large}$	1.60%	42.13%	34.95%
$KG-enhanced_{large}$	**1.79%**	**43.70%**	**35.51%**

Table 4: Query result set recall and relevance mean values, for the the source user query, target user query, the baseline model, and the KG-enhanced model. * indicates a statistical significant increase over the baseline, for a confidence level $p < 0.01$.

Model	Recall Mean	Recall Median	Rel. Mean
Source User Query	25.3	22	/
Target User Query	147.4	148	0.628
Baseline	225,227.3	37	0.543
KG-enhanced	**225,430.5***	**105**	**0.564***

2006]. We reject the null hypothesis if the results are the same for a confidence level $p < 0.01$.

The KG-enhanced encoder-decoder model is compared to a baseline encoder-decoder model using the same pre-trained encoder/decoder BERT model as before. Both models are trained for 5 epochs. We use beam search to generate the top-5 rewritten queries. Following the evaluation setup in [Wang et al. 2021] we use the following evaluation metrics to evaluate the encoder-decoder query rewriting model: (i) Recall@5: the proportion of test query pairs where the target query matches exactly (sentence-level) one of the top 5 predicted candidate queries by the model. (ii) Jaccard@5: quantifies the highest Jacard (token-level) query similarity (order agnostic) between the top-5 predicted candidates and the target query. (iii) BLEU@5: The highest BLEU score between the top-5 predicted candidates and the target query.

4.3 Results

Table 2 shows the results for the Query Pruning task, using the KG-enhanced model and the baseline token-based classification model on two datasets. We can observe that the KG-enhanced query pruning significantly outperforms the baseline model on both datasets.

The evaluation results for the full query rewriting models on two datasets are shown in Table 3. The KG-enhanced model outperforms the baseline model on all the evaluation metrics. However, we can observe that the difference is marginal. We have to note that a query rewriting model can potentially produce N valid rewrites for a source query q_s, which don't necessarily match the user target query. For example, the source query *"Kobe Bryant size 10"*, which the user reformulated to *Lebron James size 10*, could also be correctly rewritten to *"Nike shoes size 10"* or *"Kyrie Irving size 10"*.

To get better insights of the quality of the query rewrites of both models, we compare the source and rewritten queries result set recall and relevance on our whole product inventory. To do so, we selected 5,000 random queries from the dataset, for which we run the source query and the predicted queries by the KG-enhanced and baseline encoder-decoder model in our search engine and we compare the recall, and the relevance of the retrieved result set using proprietary item-to-query relevance model. In Figure 4 are shown the recall distributions of the user target query, the baseline and KG-enhanced encoder-decoder models top-1 predicted query. Visually, we can confirm that the KG rewrites produce higher recall compared to the baseline. Compared to the user rewrites, we can observe that the KG produces much longer tail, leading to significantly higher mean recall. In Figure 5 are shown the relevance distributions of the user target query, the baseline and KG-enhanced encoder-decoder models. As expected, the user target query produces results with highest relevance, followed by the KG-based rewrites. To statistically compare the Recall and Relevance distributions of the baseline and KG-enhanced models, we perform Welch's unequal variances t-test, for a confidence level $p < 0.01$. The results are shown in Table 4. For both tests, we reject the null hypothesis that the means of the samples are equal, i.e., the KG-enhanced model generates queries with significantly higher result set relevance with respect to the user target query, and significantly higher result set recall size compared to the baseline model.

4.4 Qualitative Analysis

In this section we perform qualitative analysis of the query rewrites generated by the KG-enhanced models, compared to their baseline counterparts.

In Table 5 are shown some examples query rewrites generated with the KG-enhanced and baseline query pruning model. Each query is run against our inventory and the result set size is added in brackets. As we can observe from the examples, the baseline model fails to correctly segment the query and often drops tokens that are part of an entity, leading to inconsistent and ambiguous

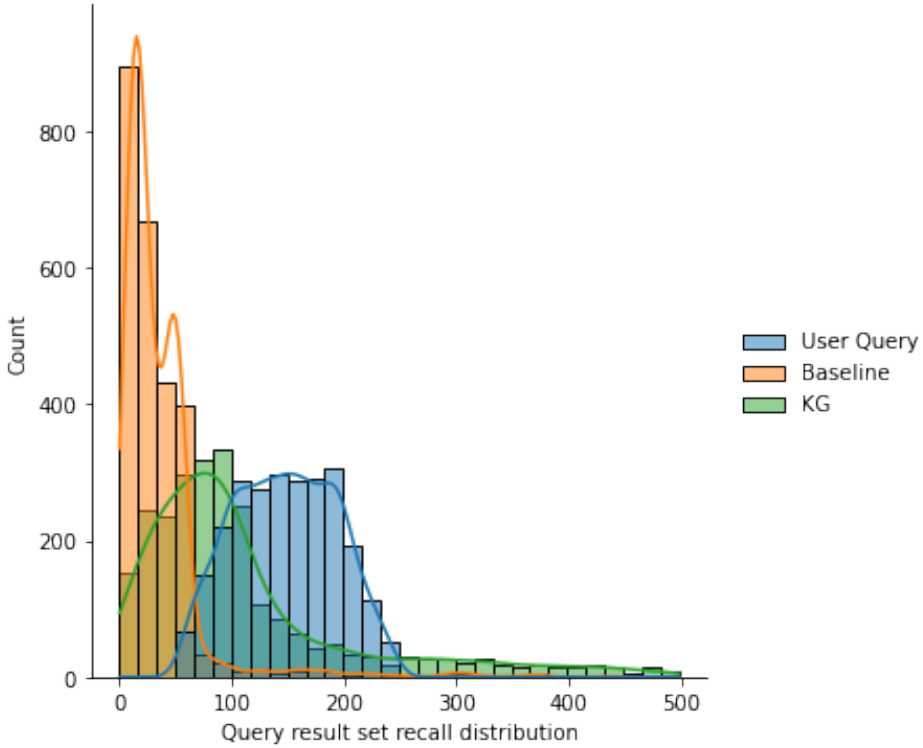

Figure 4: Query result set recall distribution.

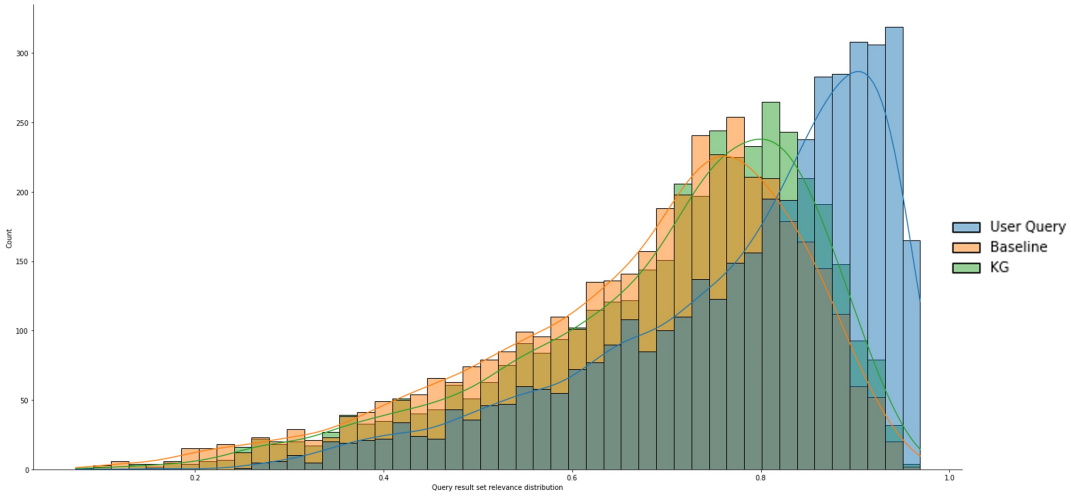

Figure 5: Query result set relevance distribution.

query rewrites. For example, in the first query example, the baseline model fails to identify "patent leather" as a complete segment, and drops the token "leather", while the KG-enhanced model correctly identifies this segment as a material entity and retains it in the predicted query. Similar behavior can be observed for the next 3 examples. For the last source query example, the frequency of the entity "Maruzen" is a crucial signal for dropping the token, as it is almost non-existent in our inventory. While the KG-enhanced model correctly removes this entity from the predicted query, as we incorporate the entity frequency directly in the model, the baseline model fails to drop the correct token.

In Table 6 are shown example query rewrites using the KG-enhanced encoder-decoder and the baseline model. We can observe that the KG-enhanced model can substitute an entity with similar

Table 5: Query rewriting examples using the query pruning baseline and KG-Enhanced model. The size of the query result set is shown in brackets next to each query.

Source Query	Target Query	Baseline Rewrite	KG-Enhanced Rewrite
ladies patent leather shoes thick heel (50)	ladies patent leather shoes (330K)	ladies patent shoes (370K)	ladies patent leather shoes (330K)
hot wheels fast and furious nissan skyline r-34 (82)	hot wheels fast and furious nissan skyline (1K)	hot wheels fast (22K)	hot wheels fast and furious nissan skyline (1K)
luka garza one and one (12)	luka garza (3.5K)	luka one (295)	luka garza (3.5K)
samurai rebellion dvd (28)	samurai rebellion (105)	samurai dvd (11K)	samurai rebellion (105)
sailor pen maruzen (5)	sailor pen (5K)	sailor maruzen (6)	sailor pen (5K)

Table 6: Query rewriting examples using the Encoder-Decoder baseline and KG-Enhanced models. The size of the query result set is shown in brackets next to each query.

Source Query	Target Query	Baseline Rewrite	KG-Enhanced Rewrite
nike dunk low marina blue (50)	nike dunk low panda (2K)	blue nike dunk low size 11 (650)	royal blue nike dunk (1.5K)
ivory prada sandals 38 (6)	cream prada sandals (130)	prada sandals 38 (400)	cream prada sandals 38 (130)
zyia leggings large (41)	leg leggings large (450K)	zyia leggings women (3.7K)	large fabletics leggings (1.3K)
lee extreme motion shorts grey 38 (24)	cargo shorts 38 (45K)	motion mens shorts 38 (299)	grey cargo shorts mens 38 (5.8K)
used rgb corsair mouse (100)	used rgb razer mouse (450)	used mouse (16K)	used gaming mouse (4.8K)

entities that fit the context of the query, or increase the scope of the query to increase the recall size. For example, in the query "ivory prada sandals 38", the KG-enhanced model correctly substitutes the color "ivory" with the visually similar color "cream". And while the rewritten query by the baseline model has a higher recall, the query is too abstract and retrieves results with lower relevance with respect to the user's intent. In the third example, the KG-enhanced model replaces the leggings brand "Zyia" with a similar brand "Fabletics" for which there are enough items in the inventory. In the 2 next examples, the KG-enhanced model correctly increases the query abstraction, to a level that offers a good trade-off between item relevance and recall size. Where the baseline model either retains tokens that are too specific and lead to low recall, or rewrites the query to a broader scope, which leads to very large recall but most of the items have low relevance for the original user's intent.

.

5 CONCLUSION

In this paper we presented two approaches for query rewriting, using auxiliary information from a Product Knowledge Graph. We embed the Product Knowledge Graph and use the entity embedding, together with NER, category and entity frequency auxiliary information to enhance two query rewriting models, i.e., query pruning model and encoder-decoder model for complete query rewriting.

In-depth quantitative and qualitative evaluation shows that the KG-enhanced query rewriting models significantly outperform their baseline counterparts. We showed that the auxiliary information extracted from the Product Knowledge Graph is a strong signal to identify the correct segments of the query to be dropped, replaced, and propose entities to be inserted.

So far, we have considered rather simple architecture for infusing the background knowledge, using linear transformation of the auxiliary vectors concatenated with the encoded tokens. In future work, we will explore more comprehensive architectures for encoding background knowledge, for example using more sophisticated graph neural networks to embed the Product Knowledge Graph [Wu et al. 2020].

Another future direction of research is building a query rewriting model on top of the KG-Copy models [Chaudhuri et al. 2019], commonly used for injecting background knowledge from KGs in question answering. In such models we can explicitly generate entity replacements based on entity similarity and relatedness calculated on the whole KG. Then using the KG-copy mechanism the model can generate top-N query rewrites, ranked by relevance in resepct to the source user query.

REFERENCES

Debanjan Chaudhuri, Md. Rashad Al Hasan Rony, Simon Jordan, and Jens Lehmann. 2019. Using a KG-Copy Network for Non-goal Oriented Dialogues. In *ISWC 2019*. Cham, 93–109.

Debanjan Chaudhuri, Md Rashad Al Hasan Rony, and Jens Lehmann. 2021. Grounding Dialogue Systems via Knowledge Graph Aware Decoding with Pre-trained Transformers. In *The Semantic Web*. Cham, 323–339.

Yan Chen and Yan-Qing Zhang. 2009. A Query Substitution-Search Result Refinement Approach for Long Query Web Searches. In *2009 IEEE/WIC/ACM*.

Zheng Chen, Xing Fan, and Yuan Ling. 2020. Pre-Training for Query Rewriting in a Spoken Language Understanding System. In *ICASSP*. 7969–7973.

Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. 1747–1756.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akabari. 2021. KI-BERT: Infusing Knowledge Context for Better Language and Domain Understanding. *arXiv preprint arXiv:2104.08145* (2021).

Angela Fan, Claire Gardent, Chloe Braud, and Antoine Bordes. 2019. Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to Rewrite Queries. 1443–1452. https://doi.org/10.1145/2983323.2983835

Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. 2020. Query Reformulation in E-Commerce Search. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA, 1319–1328.

Firas Kassawat, Debanjan Chaudhuri, and Jens Lehmann. 2019. Incorporating Joint Embeddings into Goal-Oriented Dialogues with Multi-task Learning. In *The Semantic Web*.

Xiao Liu, Juan Hu, Qi Shen, and Huan Chen. 2021. Geo-BERT Pre-training Model for Query Rewriting in POI Search. In *EMNLP*. 2209–2214.

Hanqing Lu, Yunwen Xu, Qingyu Yin, Tianyu Cao, Boris Aleksandrovsky, Yiwei Song, Xianlong Fan, and Bing Yin. 2021. Unsupervised synonym extraction for document enhancement in e-commerce search. In *Workshop on Knowledge Management in E-Commerce*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems. In *ACL*.

Saurav Manchanda, Mohit Sharma, and George Karypis. 2019. Intent term selection and refinement in e-commerce queries. *ArXiv* (2019).

Aritra Mandal, Ishita K. Khan, and Prathyusha Senthil Kumar. 2019. Query Rewriting using Automatic Synonym Extraction for E-commerce Search. In *eCOM@SIGIR*.

Yiming Qiu, Kang Zhang, Han Zhang, Songlin Wang, Sulong Xu, Yun Xiao, Bo Long, and Wenyun Yang. 2021. Query Rewriting via Cycle-Consistent Translation for E-Commerce Search. *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (2021), 2435–2446.

Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*. Springer, 498–514.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics* 8 (2020), 264–280.

Zehong Tan, Canran Xu, Mengjie Jiang, Hua Yang, and Xiaoyuan Wu. 2017. Query Rewrite for Null and Low Search Results in eCommerce. In *eCOM@SIGIR*.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question Rewriting for End to End Conversational Question Answering. https://arxiv.org/pdf/2004.14652.pdf

Yaxuan Wang, Hanqing Lu, Yunwen Xu, Rahul Goutam, Yiwei Song, and Bing Yin. 2021. QUEEN: NeuralQuery Rewriting in E-commerce.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

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 *WISDM*. New York, NY, USA, 402–410.

Siyang Yuan, Saurabh Gupta, Xing Fan, Derek Liu, Yang Liu, and Chenlei Guo. [n.d.]. Graph Enhanced Query Rewriting for Spoken Language Understanding System. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Mengxiao Zhang, Yongning Wu, Raif Rustamov, Hongyu Zhu, Haoran Shi, Yuqi Wu, Lei Tang, Zuohua Zhang, and Chu Wang. 2022. Advancing query rewriting in e-commerce via shopping intent learning. In *SIGIR 2022 Workshop on eCommerce*.

Qunzhi Zhou, Zhe Wu, Jon Degenhardt, Ethan Hart, Petar Ristoski, Aritra Mandal, Julie Netzloff, and Anu Mandalam. 2021. Leveraging Knowledge Graph and DeepNER to Improve UoM Handling in Search.. In *ISWC (Posters/Demos/Industry)*.