Wisdom of the Sellers: Mining Seller Data for eCommerce Knowledge Graph Generation

Petar Ristoski, Sathish Kandasamy, Aleksandr Matiushkin, Sneha Kamath, and Qunzhi Zhou

eBay Inc., San Jose, USA
{pristoski,satkandasamy,amatiushkin,snkamath,qunzhou}@ebay.com

Abstract. Query understanding is a fundamental part of an e-commerce search engine, and it is crucial for correctly identifying the buyer intent. To perform a semantic query understanding, in this work we introduce a relationship-rich product Knowledge Graph (KG), mined from seller provided data, which captures entities and relationships to model the whole product inventory, allowing us to identify the buyer intent more accurately.

Keywords: Knowledge Graphs · e-Commerce · Query Understanding

1 Introduction

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, which is the main cause for customer churn and loss of revenue. To bridge this gap, plethora of query understanding approaches have been introduced [1]. However, generating a precise knowledge base with high coverage for semantic query understanding remains a main challenge. In this work we mine seller provided information, to generate a high-quality KG covering the whole product inventory. To assure data quality, all the knowledge in the KG must be confirmed by a number of sellers, i.e., "wisdom of the sellers". The KG contains entities and relations describing millions of products, e.g., brands, colors, materials, sizes etc. To perform semantic query understanding, we perform entity linking using the KG [5]. Through the identified entities 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 "Oyster Bracelet Submariner", we first identify the query category in our inventory, and we pull the corresponding KG for that category, i.e., "Wristwatches". As shown in Figure 1, we are able to link each text mention to the corresponding KG entity, e.g., "Submariner" is linked to an entity of type "model". Along with the entities, we can retrieve the number of listings associated with the given entity. This allows us to capture the buyer intent, i.e., the buyer is interested in "luxury" and "classic" watches, and we can identify similar models that they might be interested in.

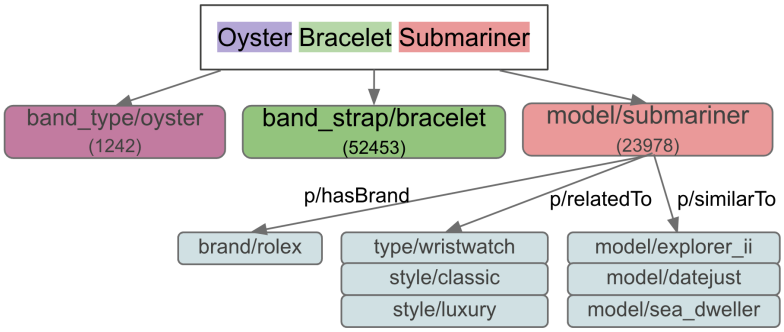

Fig. 1: KG-based semantic query understanding

In this work, we use the product KG in several query understanding applications, which significantly improve the buyer experience.

2 Approach

We build a directed weighted graph based on co-occurring aspect-value pairs in listings provided by sellers partitioned by category. Formally, we represent the product inventory as a set of product listings $L = \{l_1, l_2, \ldots, l_n\}$, where each listing l_n is represented as a set of aspect-value pairs $AV_n = \{av_1, av_2, \ldots, av_n\}$, where a is the aspect name, and v is the aspect value. Each aspect name a is converted to an *rdf:Class*, *class/A*, and the corresponding value v is converted to an instance of the class *class/A*. For example the aspect value pair *Brand:Apple* is converted to the following triple: `kg:brand/Apple rdf:type kg:class/Brand`. Each pair of co-occurring aspect-value pairs within at least one product listing, av_i and av_j is converted into 2 triples as follows: $kg : a_i/v_i \;\; kg : p/a_j \;\; kg : a_j/v_j$ and $kg : a_j/v_j \, kg : p/a_i \, kg : a_i/v_i$, where the predicates are derived from the class of the object entity. For each pair av_i and av_j, we calculate the co-occurrence frequency c_{ij}, as well as the total frequency of each aspect-value pair, c_i and c_j respectively. Then the predicate e_{ij} between these two pairs is assigned a weight $w_{ij} = c_{ij}/c_i$. Similarly, we set a weight $w_{ji} = c_{ji}/c_j$ on the symmetric edge e_{ji}. Such weights give higher relevance to aspect-value pairs that co-occur more often together, normalized by their global popularity. This results in a directed weighted graph, which is generated separately for each category.

For example, for the co-occurring aspect-values *Brand:Apple* and *Color:Sierra Blue*, we will generate the following quadruples:

```
kg:brand/apple kg:p/color kg:color/sierra_blue 0.01
kg:color/sierra_blue kg:p/brand kg:brand/apple 0.99
```

The fourth element is the weight of the triple, 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.

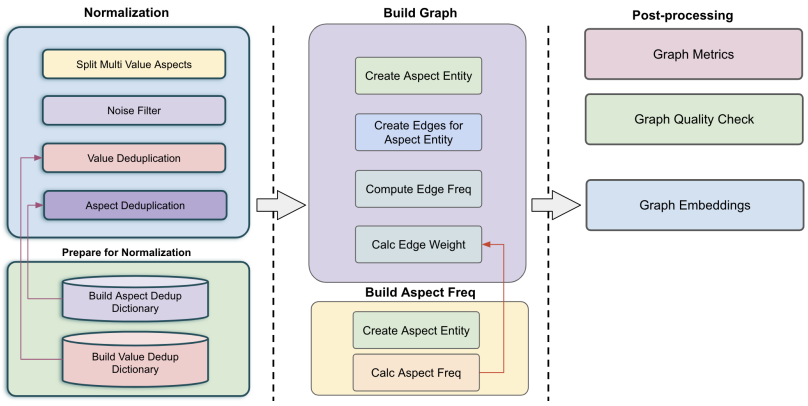

Fig. 2: KG Generation Pipeline

To assure high-quality data, we filter out noisy entities based on their frequency, consolidate entities appearing under different surface forms, and prune edges with low weight. The KG generation pipeline, shown in Figure 2 runs weekly, using Spark jobs, automated through Apache Airflow. The resulting KG contains tens of millions of entities, and hundreds of millions of relations.

To ease the use of such KG in downstream tasks, we use the graph embedding approach using biased walks [4]. 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 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 Applications

The knowledge graph is available for exploration and querying within the enterprise, as shown in Figure 3. Internal users regularly use this tool to perform exploratory data analysis, and scope new opportunities.

We use the product KG in a handful fundamental e-Commerce applications. Each application has been evaluated offline or online, i.e., A/B tests with millions of users. The tests showed a statistically significant drop in search abandonment rate and decrease in low recall search sessions, as well as a significant increase in purchased products. The applications include:

Semantic Query Expansion: Identify synonyms, hyponyms and subtype relations for semantic query expansions, for colors, materials, models, brands, etc. [3].

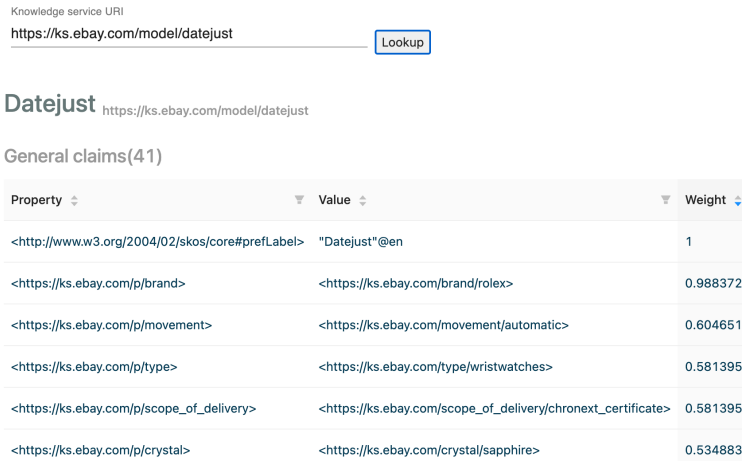

Fig. 3: KG data exploration tool

KG-enhanced Query Reformulation: Neural generative query rewriting model using KG embeddings, trained on user search logs. We build a KG-enhanced token dropping model, which is able to identify and remove least significant tokens in a query in order to increase relevant recall. Furthermore, we train a generative model for end-to-end query rewriting, which is able to identify entity substitutes or increase the query abstraction in order to increase the recall [2].

Multi-Faceted Item Recommendation: Recommend diverse items related to the initial buyer intent, by expanding on different entities. We identify entity substitutes and allow the user to pivot on different aspects of the query in order to easier identify the products they are interested in.

Listing Autocomplete and Validation: Infer missing aspect values and remove inconsistent aspect values to assist sellers when listing new items on the platform.

References

1. Chang, Y., Deng, H.: Query understanding for search engines. Springer (2020)
2. Farzana, S., Zhou, Q., Ristoski, P.: Knowledge graph-enhanced neural query rewriting. In: Companion Proceedings of the Web Conference, The 2nd Workshop on Interactive and Scalable Information Retrieval Methods for eCommerce (May, 2023)
3. Liang, L., Kamath, S., Ristoski, P., Zhou, Q., Wu, Z.: Fifty shades of pink: Understanding color in e-commerce using knowledge graphs. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 5090–5091 (2022)
4. Ristoski, P., Paulheim, H.: Rdf2vec: Rdf graph embeddings for data mining. In: International Semantic Web Conference. pp. 498–514. Springer (2016)
5. Zhou, Q., Wu, Z., Degenhardt, J., Hart, E., Ristoski, P., Mandal, A., Netzloff, J., Mandalam, A.: Leveraging knowledge graph and deepner to improve uom handling in search. In: ISWC (Posters/Demos/Industry) (2021)