European Research Studies Journal
Volume XXVII, Special Issue 2, 2024

pp. 150-159

Product Knowledge Graphs - Creating a Knowledge System for Customer Support

Submitted 18/02/24, 1st revision 16/03/24, 2nd revision 20/04/24, accepted 16/05/24

Bartosz Przysucha[1], Paweł Kaleta[2], Artur Dmowski[3], Jacek Piwkowski[4], Piotr Czarnecki[5], Tomasz Cieplak[6]

Abstract:

Purpose: *This article explores developing and integrating a product knowledge graph within an e-commerce customer support system to improve product discovery and recommendation processes.*
Methodology: *The methodology involves a structured development process for the knowledge graph, utilizing natural language processing (NLP) to extract relevant entities from product data and machine learning algorithms to establish and categorize relationships between products. The approach integrates data from multiple sources, including vendor catalogs, online reviews, and customer interactions, ensuring a comprehensive data set.*
Findings: *The research resulted in the creation of a dynamic, scalable knowledge graph that significantly enhances the accuracy and personalization of product recommendations. The graph's ability to link seemingly disparate data points allows for a nuanced understanding of user behavior and preferences, improving customer satisfaction and sales performance.*
Practical Implications: *The presented method has significant implications for retailers looking to enhance their online presence and customer interaction. By implementing this knowledge graph, retailers can expect to streamline their product recommendation processes and gain deeper insights into customer trends, which can inform broader marketing and inventory decisions.*
Value: *This study's novelty lies in applying a comprehensive knowledge graph tailored explicitly for e-commerce systems. This graph integrates abstract and concrete entities to offer a richer, more interconnected dataset than traditional relational databases.*

Keywords: *Knowledge graphs, E-commerce optimization, Natural Language Processing (NLP), Machine Learning Algorithms, product recommendations, data integration.*

JEL codes: *C45, C61, M31, L8, D83.*

Paper type: *Research article.*

---

[1]*Corresponding Author: Management Faculty, Lublin University of Technology, Lublin, e-mail: b.przysucha@pollub.pl;*
[2]*WSEI University, Lublin, Poland, e-mail: Paweł.Kaleta@wsei.lublin.pl;*
[3]*WSEI University, Lublin, Poland, e-mail: Artur.Dmowski@wsei.lublin.pl;*
[4]*WSEI University, Lublin, Poland, e-mail: Jacek.Piwkowski@wsei.lublin.pl;*
[5]*Wyższa Szkoła Biznesu - National Louis University, e-mail: pczarnecki@wsb-nlu.edu.pl;*
[6]*Management Faculty, Lublin University of Technology, Lublin, Poland, e-mail: t.cieplak@pollub.pl;*

1. Introduction

In the age of global digitization, information growth is almost exponential. The accumulation of ever-larger databases and information resources requires the automation of sales and business systems (Rymarczyk *et al.*, 2018). In a world of virtual assistants and automated recommendation systems, knowledge graph-based systems are gaining importance as a critical tool for data management and analysis.

A significant problem with e-commerce and other applications is the variety and complexity of information and data stored (Xu *et al.*, 2019). In such cases, knowledge graphs offer unique opportunities. Not only do they allow data to be efficiently stored in a way that reflects its natural relationships, but also to be dynamically populated and modified, which is invaluable in rapidly changing business environments.

With knowledge graphs, companies can analyze and extract valuable insights from the connections between data in real-time. The advantage of such a design is the ability to personalize offerings and increase the speed and convenience of service.

Such graph-based knowledge opens up new perspectives in knowledge management, enabling the creation of advanced recommendation systems that respond intelligently to users' needs, adapting to their individual preferences and purchase history.

A knowledge graph (Kejriwal *et al.*, 2021) is a data structure that uses graph models to organize and represent information in the form of nodes (or vertices) and edges. Nodes in a knowledge graph represent various objects, such as people, places, things, categories, and concepts. Graph edges, on the other hand, describe the relationships between these objects (Zampeta and Chondrokoukis, 2023).

Knowledge graphs are mainly used in systems that require complex analysis of relationships and dependencies between diverse data. Each graph element (node or edge) has a specific meaning and is often labeled with metadata describing its properties and relationships. The main advantage of knowledge graphs is the ability to perform complex queries that can discover connections between distant elements.

Knowledge graphs are particularly effective in integrating information from different sources. Knowledge graphs are often used in e-commerce systems based on artificial intelligence (Shyna and Monga, 2020; Thalassinos *et al.*, 2015). They can improve models' understanding of speech context and semantics. For example, knowledge graphs in speech recognition or product recommendation systems help interpret queries in the context of previous interactions and user preferences.

For instance, suppose a graph contains information that users who buy product A are also often interested in product B. In that case, the system can recommend B to other customers who buy A. Knowledge graphs can significantly improve the quality of search results on e-commerce platforms by better understanding the context of queries and product relationships.

This can include understanding synonyms, category relationships, and user preferences. Integrating knowledge graphs with sentiment analysis allows real-time monitoring of product and brand opinions, which can influence marketing and sales strategies (Anjam *et al.,* 2020).

Knowledge graphs in e-commerce are a tool that improves customer interactions and internal business processes, from logistics to marketing (Heist *et al.,* 2020). With the ability to create deep, semantic connections between data, these graphs enable more intuitive and responsive online sales systems.

An example is the use of knowledge graphs at Amazon, where the product recommendation system is based on extensive knowledge graphs that integrate information about products, users, their buying behavior, and preferences. Such a system allows recommendations to be precisely matched to individual users' needs and history. Another example is eBay, which uses knowledge graphs to understand better the relationship between products and categories, which improves search results and product suggestions.

E-Commerce catalogs are created by obtaining data from vendors (3P) and suppliers/brands (1P). Data provided by partners (vendors, suppliers, brands) is often incomplete, sometimes missing critical information customers seek. Although partners adhere to the specifications (established format for uploading product data), a vast amount of data is hidden in the title, description, and images.

In addition to the data provided by partners, there is a lot of unstructured data on the Internet in the form of product manuals, product reviews, blogs, social networks, etc. Unstructured data, such as product reviews, social media posts, or product descriptions, can provide valuable insights into customer preferences and expectations.

Natural language processing (NLP) technologies play a crucial role in extracting useful information from this data, which can then be integrated into knowledge graphs. For example, sentiment analysis of product reviews can automatically determine which product features are most valued by consumers, which can be directly used to optimize product offerings. Additionally, NLP techniques can help identify and categorize concepts, improving knowledge graphs' management and updating (Yan *et al.,* 2018; Tyagi *et al.,* 2023).

Implementing knowledge graphs brings many benefits, such as better personalization, more accurate recommendation systems, and a deeper understanding of customer behavior and needs. However, it also comes with challenges, including scaling systems as data volumes increase. Scaling knowledge graphs requires advanced technology solutions, such as graph databases, to effectively manage large data sets and provide high query performance (Iglesias *et al.*, 2023). Integrating data from different sources is also challenging, requiring complex data extraction and transformation methods to ensure consistency and usability in the graph.

This paper presents the concept and implementation of a product knowledge graph for an eCommerce customer support system to improve product discovery and recommendation. This graph, which connects products to their associated entities, both abstract and concrete, enables better management of product data and supports customers and employees in efficiently browsing sales catalogs.

The study describes the steps in creating this graph, including extracting entities, linking them to products, and managing the data, which are vital to improving the buying and operational experience.

2. Building a Knowledge Graph for an E-Commerce Customer Support System

Product Knowledge Graph captures the complex connections between products and the various entities in the retail world (Dong, 2018; 2019). These entities can be objects, things, concepts, or abstractions, such as interior styles or specific life situations.

Focusing on two main types of entities - abstract ("kid-friendly" statement, for example) and concrete (e.g., the color red") - the knowledge graph allows answering a variety of questions, from those concerning the general shopping context (e.g., 'living room furniture in a country house') to specific product features ("blue jeans pants").

The arrangement of products in the graph also takes into account their relationship to each other, dividing them into segments of substitutes (semantic equivalents of meanings) and complements (compatible products). This approach allows for a more comprehensive analysis of the relationships between products and increases the effectiveness of recommendation systems.

Several challenges were encountered while working on the product catalog and had to be dealt with. One of the main problems was the lack of a single, reliable source of product data. In addition, the catalog contained much misinformation from partners.

Thus, several steps had to be taken when creating a knowledge graph, including building a bipartite graph - products on one side and interchangeable products on the other, using the existing taxonomy and effectively linking products to entities.

An entity is an abstract representation of a data unit that can be stored in a graph as a node (or vertex). Entities can be represented by different entities, such as products, brands, features, trends, preferences, or categories, depending on the context of the analysis.

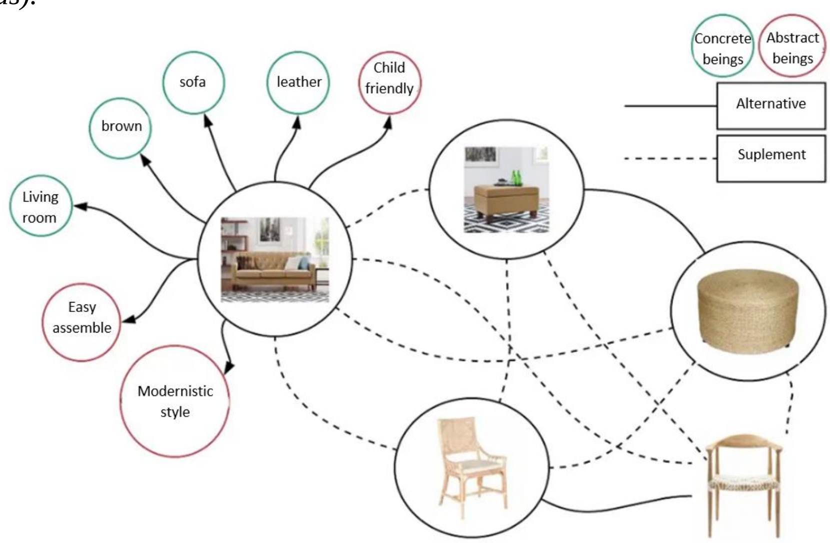
The constructed knowledge graph addresses this need by capturing not only product data but also related entities to facilitate customer and employee product discovery in sales catalogs. This product knowledge graph can be a crucial tool to support semantic search and recommendation systems in the retail context.

An essential step in creating a product knowledge graph was effectively identifying and extracting entities from the content describing the products. The identified entities were then linked to the corresponding products, creating a structure composed of triples (product, relationship, entity), which are the fundamental elements of the knowledge graph.

An additional management layer was necessary to ensure that only those triples that exceeded a certain confidence threshold were accepted for further use in the graph. This approach aimed to maintain high data quality in the graph, which is crucial for the effectiveness of recommendation systems and semantic search engines.

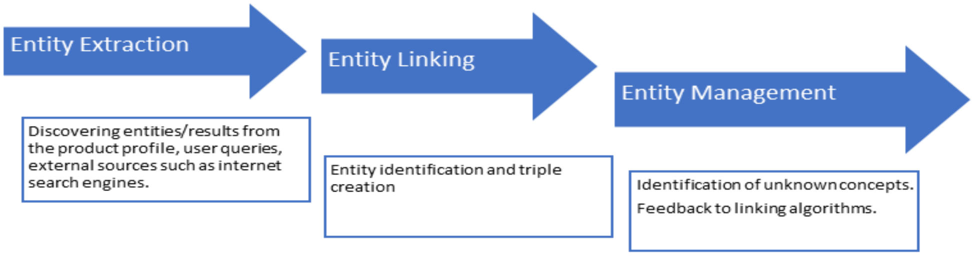
The second important aspect was establishing Product <-> Product relationships, where products were classified as substitutes or complements.

Figure 1. *Conceptual representation of the retail graph (edge labels are in the legends).*



Source: *Own creation.*

Figure 2. *The process of building a Product<->Enteration graph.*



Source: *Own creation.*

2.1 Entity Extraction

The purpose of the entity extraction module is to extract "entities" from product titles and descriptions, which can vary widely depending on the type of assortment. Product descriptions are diverse; some are long-winded, while others may contain short phrases written in bullet point form. For these reasons, two algorithms for extracting entities from product content have been developed to deal with the variety and specificity of texts effectively.

- NLP-based model

The entity extraction process began by analyzing product titles, descriptions, and metadata. A linguistic model was built using the Stanza library, provided by Stanford Core NLP (https://stanfordnlp.github.io /stanza/installation_usage.html) to accomplish this task. This model proved highly effective because product titles and descriptions are often in bulleted form, presenting critical product information in a bulleted list rather than in complete sentences.

- heuristic model

An alternative approach that yielded positive results was using heuristic rules to analyze product descriptions. Vendors and suppliers often use specific formats, such as HTML tags, to highlight key product features. Accordingly, heuristic rules were developed to analyze and extract critical information from such formatted descriptions.

2.2 Combining Entities

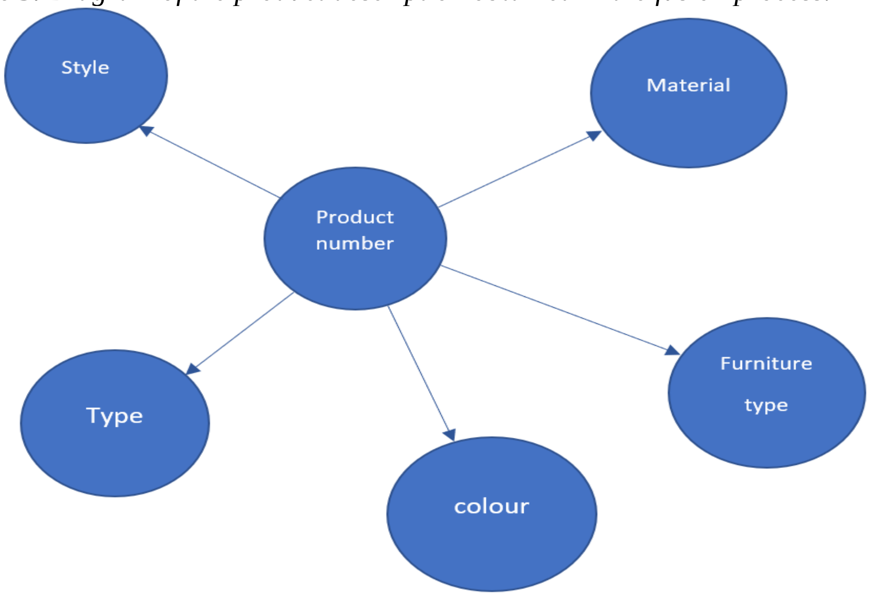
After extracting the entities, it is necessary to determine what they represent and their relationship to the product (defined by, for example, the part number). Let us consider the entity "mid-century sofa." It would be best to determine what "mid-century" means in the context of the sofa. This process, called entity linking, involves linking the extracted entity to the corresponding product, using the catalog number as a reference.

In addition, the entity linking module performs the vital function of contextual disambiguation. For example, the word "cherry" can have different meanings depending on the context - it can mean fragrance in the context of a candle, flavor in the case of juice, finish in furniture, color in textiles, or fruit in a grocery store. A category or product type usually defines the context in question.

The consolidator inputs the product type and entity during the merging process, forming a trinity as entity-predicate-object. Due to the lack of a uniform, accurate source of product data, the task of entity merging is complex and challenging. The process begins with creating a dictionary that includes product type, attribute name, and attribute values based on a set of best-selling products.

It is assumed that the data on these products is more accurate. The first step is to use this dictionary to identify potential candidates, regardless of context. Then, a second model is run to rank the selected candidates based on their context. These results are presented as the output of the consolidator in Figure 3.

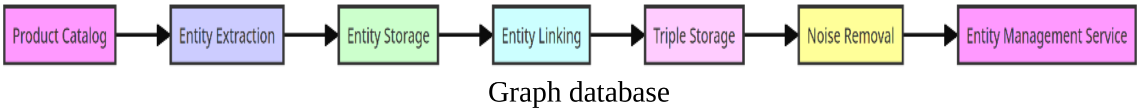
Figure 3. *Diagram of the product description obtained in the fusion process.*



Source: *Own creation.*

The identification of substitutes for specific products uses both textual and graphic data. Visual similarity plays a crucial role in identifying substitutes in some product categories, such as furniture or clothing. Image and text embedding methods are used for analysis, and the resulting data is entered into FAISS (Faiss is a library developed by Facebook for efficient similarity search and dense vector clustering) (Welcome to Faiss Documentation).

Figure 4. *Data flow in the Product<->Enteration pipeline.*



Source: *Own creation.*

For each product, a set of k-nearest neighbors (KNN) is created from textual and visual embeddings to generate candidate sets. Then, category-specific ranking logic is applied to determine the final substitutes. For example, in the furniture category, the "style of home decor," such as mid-century, coastal, or country house, plays a crucial role in determining substitutes, influencing the ranking logic used.

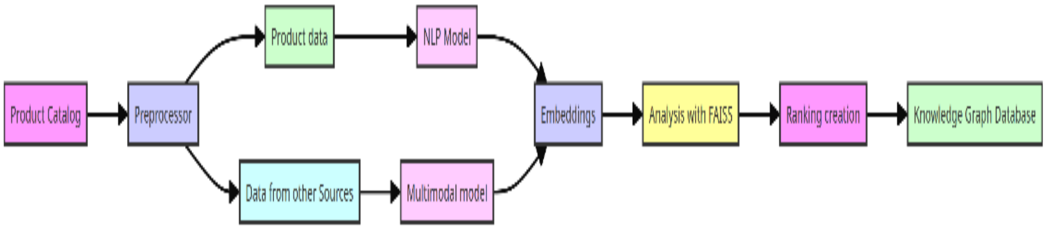
The system's architecture for creating product knowledge graphs is based on two main data processing pipelines - one responsible for generating Product <-> Entity relationships and the other for Product <-> Product. Both of these pipelines are cycled through the local Kubernetes cluster. Once the processing is complete, APIs are used to process the candidates and store the results in the Neo4j database. Figure 4 shows a very general diagram of these processing pipelines.

In finding replacements and additions, various data sources are used, such as product catalogs and external sources, resulting in the division of the data flow into at least two streams. The first stream uses product data collected in the catalog, which is then subjected to embedding.

Data is pulled from various online sources, such as websites and online catalogs, to expand the available assortment to include products from other suppliers or manufacturers. In this case, the data can be textual and include graphics and multimedia materials (Hu and Chen, 2022). Then, the data is embedded (Xu *et al.*, 2020 Choudhary *et al.*, 2021).

Identifying similarities between products is implemented using the FAISS library. Since the set of similarities is usually very large, establishing a cutoff threshold is necessary to decide which products can be treated as complements or substitutes and which are significantly different from the case under consideration. A diagram of this process can be found in Figure 5.

Figure 5. *Data flow in Item <-> Object pipeline generating substitutes, product additions.*



Source: *Own creation.*

Among modern technologies, NoSQL databases (such as Neo4j) offer numerous advantages, including flexibility, scalability, performance, and handling large amounts of unstructured, semi-structured, and structured data. To improve the efficiency of handling user queries to the knowledge graph, it is crucial to continuously replenish the knowledge store with new entities and relationships in the graph database and optimize the graph structure.

3. Conclusions

The study highlights the importance of using knowledge graphs in e-commerce, demonstrating an approach to managing complex data interactions. Implementing knowledge graphs makes it possible to connect wide-ranging data points that traditional relational databases may not be able to connect effectively.

Using natural language processing (NLP) techniques and machine learning algorithms, this study demonstrates the ability of knowledge graphs to not only consolidate distributed data sources, but also transform them into actionable insights that drive predictive analytics and personalized recommendations.

The study demonstrates the dynamic adaptability of knowledge graphs in e-commerce environments where customer preferences and market trends are constantly evolving. The ability to quickly integrate and analyze new data ensures that companies remain agile, responding deftly to consumer demands and market changes.

In addition, the knowledge graph's role in synthesizing information from multiple data sources improves decision-making by offering a detailed understanding of customer behavior patterns and potential market opportunities. The challenges of scaling and integrating disparate data sources into a coherent model underscore the need for continued innovation in graph technology and machine learning methodologies.

Applying knowledge graphs is a critical asset in maintaining competitive advantage and supporting sustainable growth.

References:

Anjam, M., Khan, H., Ahmed, S., Thalassinos, E.I. 2020. The Antecedents of Consumer Eco-Friendly Vehicles Purchase Behavior in United Arab Emirates: The Roles of Perception, Personality Innovativeness and Sustainability. International Journal of Economics & Management, 14(3).

Choudhary, S., Luthra, T., Mittal, A., Singh, R. 2021. A Survey of Knowledge Graph Embedding and Their Applications. Doi:10.48550/arxiv.2107.07842.

Dong, X.L. 2018. Challenges and Innovations in Building a Product Knowledge Graph, 2869-2869. doi:10.1145/3219819.3219938.

Dong, X.L. 2019. Building a broad knowledge graph for products, Proc Int Conf Data Eng, 25. doi:10.1109/ICDE.2019.00010.

Heist, N., Hertling, S., Ringler, D., Paulheim, H. 2020. Knowledge Graphs on the Web - an Overview. ArXiv, abs/2003.00719.

https://stanfordnlp.github.io/stanza/installation_usage.html.

Hu, H., Chen, Y. 2022. Progress and Analysis of Multimodal Knowledge Graphs. 2022 7th International Conference on Intelligent Computing and Signal Processing, ICSP 2022, 1097-1101. doi: 10.1109/ICSP54964.2022.9778820.

Iglesias, E., Jozashoori, S., Vidal, M.E. 2023. Scaling up knowledge graph creation to large and heterogeneous data sources. Journal of Web Semantics, Volume 75, 100755. https://doi.org/10.1016/j.websem.2022.100755.

Kejriwal, M., Knoblock, C.A., Szekely, P. 2021. Knowledge Graphs: Fundamentals, Techniques, and Applications. MIT Press.

Rymarczyk, T., Cieplak, T., Kłosowski, G., Rymarczyk, P. 2018. Design of Data Analysis Systems for Business Process Automation. Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska, 8(3), 43-46. https://doi.org/10.5604/01.3001.0012.5283.

Shyna, K., Monga, V.A. 2020. Study on Artificial Intelligence in E-Commerce. Welcome to Faiss Documentation — Faiss documentation. https://faiss.ai.

Thalassinos, E.I., Venediktova, B., Zampeta, V. 2015. Applications of M-GARCH Model for the Selection of Securities of Banks' Investment Portfolio. Applied Economics and Finance, 2(2), 1-13.

Tyagi, P., Grima, S., Sood, K., Balamurugan, B., Özen, E., Thalassinos, E.I. (Eds.). 2023. Smart analytics, artificial intelligence and sustainable performance management in a global digitalised economy. Emerald Publishing Limited.

Xu, D., Ruan, C., Körpeoglu, E., Kumar, S., Achan, K. 2019. Product Knowledge Graph Embedding for E-commerce. Proceedings of the 13th International Conference on Web Search and Data Mining.

Yan, J., Wang, C., Cheng, W. 2018. A retrospective of knowledge graphs. Front. Comput. Sci., 12, 55-74. https://doi.org/10.1007/s11704-016-5228-9.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K. 2020. Product knowledge graph embedding for E-commerce. WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining, 672-680. doi: 10.1145/3336191.3371778.

Zampeta, V., Chondrokoukis, G. 2023. A Comprehensive Approach through Robust Regression and Gaussian/Mixed-Markov Graphical Models on the Example of Maritime Transportation Accidents: Evidence from a Listed-in-NYSE Shipping Company. Journal of Risk and Financial Management, 16(3), 183.