Specification and Identification of Relationships between Products in the Food Segment

Sian Brumm, Rolf Krieger, and Christoph Brosch

Trier University of Applied Sciences, Umwelt-Campus Birkenfeld
s.brumm@umwelt-campus.de
r.krieger@umwelt-campus.de
c.brosch@umwelt-campus.de

Abstract. There are numerous relationships between products in retail. Understanding these relationships is particularly valuable for optimizing e-commerce, master data management, and marketing, among other applications. However, these relationships are often not explicitly known. In this work, we address the problem of automatically detecting a wide range of product relationships within the food segment. To achieve this, we provide a detailed specification of relationships specific to food products. Additionally, we describe multi-class classification models for automatically determining these relationships.

Keywords: Product Relations, Product Matching, Machine Learning

1 Introduction

Retail companies often manage data for hundreds of thousands, and sometimes millions, of products in their information systems. Understanding the relationships between different products is crucial for maintaining and utilizing this product data across various functional areas of the company. There are many types of relationships. For example, two products may be considered related if they share the same brand and content but differ in package size, or if they share the same brand, content type, and quantity but come in different flavors. Many other types of relationships are also of interest.

Knowledge about product relationships can be leveraged in various ways to support retail business processes. For example, certain relationships may be useful in sales to recommend complementary products, such as accessories or those that present an up-sell opportunity. Product recommendations are a key tool in online stores for boosting sales. Competitive pricing analysis, on the other hand, requires a detailed comparison of product prices between competitors. To perform such an analysis, one must first identify the products to compare, which involves determining the relationships between products in the assortments of both competitors. In master data management, these relationships can simplify or even partially automate the maintenance of master data. For instance, the process of creating new product records can be facilitated by suggesting attribute values from similar products, and significant discrepancies in these attributes might indicate potential errors. Thus, understanding product relationships can enhance the quality of product data. The relationships are also important when constructing product graphs. A product graph is a knowledge graph that represents products and their relationships in an organized manner. It is typically stored in a graph database.

Many existing information systems, such as those for enterprise resource planning, product information management, space planning, and online shopping, already offer

functionality to maintain such relationships. In some cases, these relationships are represented through specialized product types, such as bills of materials, sales sets, displays, generic articles, or product variants. However, maintaining these relationships often requires manual input in the systems, which is prone to errors.

In this paper, we propose a machine learning (ML)-based approach to automatically determine relationships between products. Due to the variety of products and the resulting complexity of the problem, we focus solely on food products. The paper is structured as follows: In the next section, we provide an overview of recent work on identifying product relationships. Afterwards, we define several types of relationships. In Section 4, we introduce various multi-class classification models for their automatic determination. The subsequent section presents experiments comparing the performance of these models. Finally, we summarize key findings and offer suggestions for future research.

2 Related Work

Data matching involves comparing data sets from different sources to identify identical entities. In product matching, the focus is specifically on comparing product information to recognize the same product despite differences in representations. Christen's book [1] provides a general overview of the topic and explains key procedures that are also used in this work.

The research field of product matching has developed considerably in recent years, particularly through the use of machine learning. While older works, such as Bezu et al. [2], determine the similarity between products using string-based methods or by comparing word frequencies, ML models have proven to be much more powerful. Especially Neural Networks, such as Convolutional Neural Networks and Siamese Networks, have shown promising results, as demonstrated by Shah et al. [3] and Vilcek et al. [4] who used latter for duplicate detection. In addition to these advances, the use of multimodal data is becoming increasingly important. Wilke and Rahm [5] demonstrated that combining text and image data can enhance product-matching tasks. The WDC product dataset, developed by Primpeli et al. [6], is a key resource in this area of research. They also evaluate various machine learning models using this dataset in their study.

Another key development in product matching is using text-based embeddings specially adapted to the problem domain of products. Work such as that of Tracz et al. [7] and Peeters et al. [8] showed that the generalization performance can be significantly increased by fine-tuning BERT models with domain-specific product data. Peeters et al. [8] achieved an F1 score of over 90% by additionally training BERT with large amounts of product data. The potential of Large Language Models (LLMs) for product matching, is also currently being explored. Peeters and Bizer [9] found that LLMs achieve strong zero-shot performance in product matching, comparable to language models trained on thousands of examples.

While most research focuses on recognizing identical products, other approaches focus on identifying similar products. Zuo et al. [10] developed a product recommendation system based on a Siamese Neural Network that divides similar products into different categories.

This work aims to enhance the performance of product relationship classification and to provide a more detailed definition of these relationships compared to previous studies, with a focus on relationships tailored to the food sector. The Schema.org [11] and GS1 [12] ontologies define several attributes for modeling relationships between products. Schema.org defines relationships such as *isSimilarTo*, *isRelatedTo*, *IsAccesscoryOrS-*

parePart, *IsConsumableFor*, *PredecessorOf* and *SuccessorOf*. However, the meanings of these relationships are specified only in broad and abstract terms, which allows for flexibility but also introduces ambiguity. GS1 provides a similar yet different set of relationships within its GS1 Web Vocabulary [12]. While Schema.org tends to define more abstract relationships, GS1 offers a more refined structure focused on the specifics of supply chain and B2B operations. Relationships such as *equivalentProduct*, *primaryAlternateProduct*, *dependentProprietaryProduct*, *replacedProduct*, and *replacedByProduct* support more specific use cases, particularly in the context of retail and manufacturing.

3 Specification of Product Relationships

Based on the specifications provided by Schema.org, we now give an extended and more detailed specification of the relationships between food products.

Given a set of food products P, each product $p \in P$ is described by a set of properties A. Each product relation is a subset $R \subseteq P \times P$. The problem considered in this paper is to decide whether $(p, q) \in R$ for two products $p, q \in P$, $p \neq q$ based on various product properties. Our specification of product relationships is based on the following properties: brand, type, content, packaging, refill bag, and use. The type of a product corresponds to the category to which the product is assigned. In addition, the Global Trade Item Number (GTIN) is a property that can be used to determine identical products. All in all, we distinguish the following relations:

SameAs/SameAs:DiffPackaging The symmetric relation *SameAs* is used to group identical products together. The relationship p *SameAs* q is valid if the data records p and q describe the same product. Such a relationship can be identified using the GTIN, if available. The brand, type, content, and packaging of products must also match. This means that $p.Brand = q.Brand$, $p.Type = q.Type$, $p.Content = q.Content$ and $p.Packacking = q.Packacking$. If two products p and q only differ in their packaging size or packaging type, it holds p *SameAs:DiffPackacking* q.

IsConsumableFor The asymmetric relation *IsConsumableFor* describes which products serve as consumables for other products. Although this relationship is rare in the food sector, it can be used for products such as refill packs. If p *IsConsumableFor* q, we have p *SameAs:DiffPackacking* q and $p.IsRefillBag = TRUE$.

isVariantOf/isVariantOf:DiffPackaging The symmetric relation *IsVariantOf* is used if products of the same brand and type slightly differ in content, e.g. concerning flavor, consistency, or formulation. The packaging of both products only differs in print. The shapes and sizes are the same. This means that $p.Brand = q.Brand$, $p.Type = q.Type$, $p.Content \approx q.Content$, $p.Packacking \approx q.Packacking$. If products differ in packaging or size in addition to the variant it holds p *isVariantOf:DiffPackacking* q.

IsRelatedTo/IsRelatedTo:Brand:DiffType/IsRelatedTo:Usage The relation *IsRelatedTo* specifies product relationships independent of brand, type, and content. A specialization of this symmetric relation is *IsRelatedTo:Usage*, which relates products frequently used together. In this work, we checked ingredient lists of recipes to determine these product relationships automatically. If ingredients p and q are often listed together, we assume p *IsRelatedTo:Usage* q. This approach only considers a part of all possible relationships based on the usage of products. If two products p and q have the same brand but different content ($p.Type \neq q.Type$) it holds p *IsRelatedTo:Brand:DiffType* q.

IsSimilarTo/IsSimilarTo:DiffPackaging The symmetric relation *IsSimilarTo* defines relationships between products of different brands. But related products have the

same type and similar content. The content only differs in flavor, consistency, or formulation. The volume of the packaging is nearly the same. Therefore, if $p.Brand \neq q.Brand$, $p.Type = q.Type$, $p.Content \approx q.Content$ and $p.Packacking \approx q.Packacking$ it holds p *IsSimilarTo* q. If the packagings differ significantly, it holds p *IsSimilarTo:DiffPackaging* q.

Other product relationships are also important but not considered here. For example, the replacement of a product p by a product q in the manufacturer's assortment.

In the following, we focus on the relationships *SameAs*, *IsVariantOf*, and *IsSimilarTo*. The specification discussed in this section is the foundation for our label policy, which we used when annotating the training data.

4 Model Development

Based on the specification in Section 3 a procedure for the automated determination of the product relationships was developed. It consists of three steps: 1. Data Preparation, 2. Blocking, and 3. Multi-Class Classification of Product Relationships. These steps are explained in more detail in the next subsections. Both text data, such as name, description, and categories, and image data were utilized.

The primary data source is a web crawler that extracts products and their corresponding attributes from German online stores, focusing on those selling food products. After applying filters for attribute completeness and ensuring only food products were included, a total of 107 568 products from seven online shops were retained. The relationships between products should be identified independently of the online stores, which led to challenges due to the varying structures and taxonomies of the stores, such as differences in naming conventions and product categories. Table 1 shows an example of the text data used for the model development. Additionally, data from the ERP system of a retail company about internal product relations (such as displays and price groups) was available for this work.

Attributes	**Product p (Shop A)**	**Product q (Shop B)**
name	Coca-Cola Zero Sugar 0,33l	Coca-Cola Zero 0,33L
description	Coca-Cola Zero Sugar. Keine Kalorien. Null Zucker. Für alle Coke Liebhaber ...	Kalorienfreies, koffeinhaltiges Erfrischungsgetränk mit Pflanzenextrakten, mit Süßungsmitteln ...
categories	Getränke & Genussmittel, Soft Drinks, Cola	Startseite, Lebensmittel, Getränke
brand	Coca Cola	Coca Cola

Table 1. Attribute values of a product from different online shops.

4.1 Data Preparation

Text-based attributes were processed by removing special characters and stop words. We segmented the product names to extract attributes such as brand, packaging size, and packaging volume. For this purpose, we trained a Named Entity Recognition (NER) model using data labeled by ChatGPT 3.5, followed by manual validation. To support

further processing, we performed feature extraction to vectorize both text and image data. For text data, we tested three different embedding models: the BERT model bert-base-german-uncased [13], the SBERT model distiluse-base-multilingual-cased-v2 [14], and the OpenAI model text-embedding-3-small [15]. Image data was vectorized using a pre-trained ResNet50 model [16]. All embeddings were stored in a vector database, and their impact on performance is discussed in Section 3.

4.2 Blocking Strategy

Our objective is to identify all specified product relationships for a given product $p \in P$. In a naive approach, p must be compared with $|P|-1$ other products. For some relationships, the number of comparisons can be heuristically reduced. Therefore, we implemented a blocking algorithm as proposed in [1]. Products are categorized by their *Type* and *Brand* attributes, utilizing the Global Product Classification (GPC) [17] system to map the product type. We utilize a fine-tuned pre-trained BERT model that assigns products to their respective GPC brick codes with an F1 score of 0.91. The training data for this model was provided by our retail partner. If the result set of the blocking algorithm contains more than 100 items, an Approximate Nearest Neighbor [18] algorithm is applied to narrow it down based on feature vectors. With this approach, we reduce the average number of products in the candidate set for a product considerably, while retaining 80% of all ground truth matches evaluated based on product pairs having the same GTIN.

4.3 Multi-Class Classification of Product Relationships

The product relationships *SameAs*, *IsVariantOf*, and *IsSimilarTo* are determined automatically using machine learning models, as well as a model based on attribute similarities. If the relationship between the two products does not belong to one of the three relationships mentioned before, it is assigned to the class *NotSpecified*.

The attribute-based method uses vector representations of the product attributes name, description, categories, and image. Based on these vectors the cosine similarity of the attribute values is calculated. The total similarity is determined by weighting the attribute similarities according to

$$sim(p,q) = \frac{\sum_{i=1}^{n} w_i \cdot sim(p.A_i, q.A_i)}{\sum_{i=1}^{n} w_i}.$$

A weight $w_i \in [0;1]$ expresses the relevance of attribute A_i, $1 \leq i \leq n$. Threshold values were determined to assign the product pairs to their corresponding relationships.

Random Forest (RF) [19] and Siamese Neural Network (SNN) [20] were considered for the ML-based classification of product relationships, receiving the concatenated vector representations of all text attributes as input. In addition, a hybrid approach using image data was explored. In the case of SNN, two text embeddings from different products are processed through a shared dense layer with 1 024 neurons to extract features. The two outputs are combined for the classification using a softmax function in the last layer, with categorical cross-entropy as the loss function.

With rule-based algorithms, we can further determine the sub-relations. The *Diff-Packaging* specification is applied after identifying the corresponding general product relationships by comparing the packaging sizes extracted from the product names. For the product relationship *IsConsumableFor*, the *SameAs* relationship must first be determined, and the product name is then checked for indicators of refill packs. The product relationship *IsRelatedTo:Brand:DiffType* is established by analyzing products of the same brand that are of a different type.

5 Experiments

As described in Section 4.3 we implemented several multi-class classification models for the determination of the four main product relationships *SameAs*, *IsVariantOf*, *IsSimilarTo*, and *NotSpecified*. The models were evaluated with a series of experiments. The performance of the blocking procedure and the rule-based determination of the relationships such as *IsConsumableFor* are not considered in the experiments.

5.1 Datasets

Product pairs from the web crawler dataset were manually labeled to train the models. The used labeling policy is based on a subset of the product relationships described in Section 3. The *SameAs* relationship was labeled automatically by comparing GTINs, yielding 45 093 product pairs. The relationships *IsVariantOf* and *isSimilarTo* were labeled 3 502 and 1 751 times, respectively. The data for the *NotSpecified* class was generated by randomly selecting product pairs with different GPC brick codes.

Dataset	SameAs	IsVariantOf	IsSimilarTo	NotSpecified	Total
small_ds	1 700	1 700	1 700	1 700	6 800
middle_ds	3 000	3 000	1 751	3 000	10 751
large_ds	5 000	3 502	1 751	5 000	15 253

Table 2. Dataset statistics regarding sizes and distribution of product relationships

Three datasets of varying sizes were created from the labeled data to evaluate the impact of training data size on model performance. Due to the high manual effort required to label the *isVariantOf* and *IsSimilarTo* relationships, a smaller amount of data is available for these classes, leading to unevenly distributed datasets. An analysis of the GPC classes in the largest dataset, which contains 21 245 unique products, revealed that most products are categorized under alcoholic beverages (17%). Other frequently represented categories include sweets (11%), non-alcoholic ready-to-drink beverages (10%), herbs/spices/extracts (5%), and sauces/spreads/dips/seasoning sauces (5%). In total, 67 distinct GPC classes were identified.

5.2 Results

Our models for classifying the four main product relationships were evaluated in a series of experiments. The results are presented in Table 3.

Our baseline is a simple model based on attribute similarities (AS). It serves as a starting point for our analysis. As shown by experiment E1 it achieves an F1 score of 0.69 on the classification task. Identical weights were used for the attributes. In model AS_w, the name and image attributes were weighted five times more heavily than the description and categories leading to improvements in classification. For both models, the threshold values were 0.8 for the *SameAs* relationship and 0.5 for *IsVariantOf* and, if the brand is different, 0.5 for the *IsSimilarTo* relationship.

In addition, various ML models with different parameters were examined. All models were trained with 80% of a dataset and evaluated with the remaining 20%. The first experiments used Random Forest models configured with 300 trees, with a minimum

	Parameters				Weighted			Macro		
Exp.	**Model**	**Embedding**	**Data**	**Attributes**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
E1	AS	openai	small_ds	ndci	0.71	0.68	0.69	0.71	0.68	0.69
	AS_w	openai	small_ds	ndci	0.78	0.74	0.75	0.78	0.74	0.75
E2	RF	bert	small_ds	ndcb	0.79	0.78	0.78	0.79	0.78	0.78
	RF	bert	middle_ds	ndcb	0.82	0.81	0.81	0.82	0.80	0.81
	RF	bert	large_ds	ndcb	0.86	0.85	0.85	0.87	0.81	0.83
E3	RF	sbert	large_ds	ndcb	0.86	0.86	0.85	0.86	0.82	0.83
	RF	openai	large_ds	ndcb	0.86	0.86	0.86	0.87	0.82	0.84
E4	RF	openai	large_ds	nd	0.77	0.76	0.76	0.79	0.71	0.73
	RF	openai	large_ds	ndcbi	0.85	0.84	0.84	0.85	0.79	0.81
E5	SNN	openai	large_ds	ndcb	0.84	0.83	0.84	0.82	0.81	0.81
	SNN	sbert	large_ds	ndcb	0.84	0.84	0.84	0.83	0.81	0.82

Table 3. Experimental results concerning the classification of product relationships. For each experiment, the metrics precision, recall, and F1 score are calculated. The column *Model* shows the model types (AS: model based on (weighted) cosine similarities of attributes, RF: Random Forest, SNN: Siamese Neural Network). The column *Embeddings* specifies the used embedding model. The column *Data* shows the used dataset. The column *Attributes* lists the attribute combination used in an experiment (n: name, d: description, c: categories, b: brand, i: image).

size of 1 for leaves and 5 for splits. In the experiment E2, the influence of the dataset size was investigated. It was shown that increasing training data is directly related to improved classification performance. In the experiment E3, the impact of different embedding models on classification was investigated. The best result was achieved by the Openai model in combination with the largest dataset with an F1 score of 0.86. The influence of different combinations of attributes as input to the model was then tested in experiment E4. Here it was shown that using all text attributes delivered the best results, while adding image data did not bring any general improvement. In the final experiment, Siamese Neural Networks were evaluated for the classification task. The models were trained with an Adam optimizer over 12 epochs with a batch size of 32. Compared to the RF models, they achieved slightly weaker results with F1 scores of 0.84.

6 Conclusion

In this work, various relationships between food products were defined. Three approaches were implemented and tested to classify these product relationships: an attribute-based method, a Random Forest classifier, and a Siamese Neural Network. The machine learning models achieved F1 scores of approximately 0.85 for classifying the product relationships *SameAs*, *IsVariantOf*, and *IsSimilarTo*. Initial experiments on converting the multi-class classification problem into a binary problem, by using one classification model for each relationship type, showed that this method could further improve the recognition of individual product relationships.

Future research could explore leveraging large language models to classify the defined product relationships more effectively. Expanding the training dataset, either by increasing its size or incorporating additional attributes such as ingredient lists, presents a promising direction for improving the system's accuracy and generalizability. Moreover, assessing the performance of the overall process - particularly the reduction of the candidate set through the blocking procedure - remains an important area for further experimental investigation.

7 Acknowledgements

Part of the research presented in this paper was funded by the German Ministry of Education and Research under grant FKZ 01IS23060. AI tools were used to improve the writing style and for grammar-checking.

References

1. Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
2. Bezu, R., et al.: Multi-component similarity method for web product duplicate detection. (April 2015) 761–768
3. Shah, K., et al.: Neural Network based Extreme Classification and Similarity Models for Product Matching. In Bangalore, S., Chu-Carroll, J., Li, Y., eds.: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), New Orleans - Louisiana, Association for Computational Linguistics (June 2018) 8–15
4. Vilcek, A., et al.: Transformer-Based Deep Siamese Network for At-Scale Product Matching and One-Shot Hierarchy Classification. (2021)
5. Wilke, M., Rahm, E.: Towards Multi-Modal Entity Resolution for Product Matching. In: GvDB. (2021)
6. Primpeli, A., Peeters, R., Bizer, C.: The WDC Training Dataset and Gold Standard for Large-Scale Product Matching. In: Companion Proceedings of The 2019 World Wide Web Conference, San Francisco USA, ACM (May 2019) 381–386
7. Tracz, J., et al.: BERT-based similarity learning for product matching. In Zhao, H., et al., eds.: Proceedings of Workshop on Natural Language Processing in E-Commerce, Barcelona, Spain, Association for Computational Linguistics (December 2020) 66–75
8. Peeters, R., Bizer, C., Glavas, G.: Intermediate Training of BERT for Product Matching. In: DI2KG@VLDB. (2020)
9. Peeters, R., Bizer, C.: Entity Matching using Large Language Models (June 2024)
10. Zuo, Z., et al.: A flexible large-scale similar product identification system in e-commerce. https://www.amazon.science/publications/a-flexible-large-scale-similar-product-identification-system-in-e-commerce (2020)
11. Schema.org: Homepage. https://schema.org/
12. GS1-AISBL: Web Vocabulary. https://ref.gs1.org/voc/
13. Huggingface: Word embedding bert-base-german-uncased. https://huggingface.co/dbmdz/
14. Huggingface: Word embedding distiluse-base-multilingual-cased-v2. https://huggingface.co/sentence-transformers/
15. Openai: Embeddings OpenAI API. https://platform.openai.com/docs/guides/embeddings/
16. PyTorch: resnet50 Torchvision. https://pytorch.org/vision/main/models/resnet.html
17. GS1-AISBL: How GPC works - Standards | GS1. https://www.gs1.org/standards/gpc/
18. Milvus: Milvus Documentation. https://milvus.io/docs/overview.md
19. Parmar, A., et al.: A Review on Random Forest: An Ensemble Classifier. In Hemanth, J., et al., eds.: International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018. Volume 26. Springer International Publishing, Cham (2019) 758–763 Series Title: Lecture Notes on Data Engineering and Communications Technologies.
20. Ilina, O., Ziyadinov, V., Klenov, N., Tereshonok, M.: A Survey on Symmetrical Neural Network Architectures and Applications. Symmetry **14**(7) (July 2022) 1391