Comparative Analysis of Entity Matching Approaches for Product Taxonomy Integration

Michel Hagenah and Michaela Kümpel[a]
Institute for Artificial Intelligence, University of Bremen, Am Fallturm 1, 28359 Bremen, Germany

Keywords: Entity Matching, Knowledge Engineering, Comparative Analysis, Word Embeddings, Large Language Models, WordNet, Lemmatization.

Abstract: This work examines different approaches to solving the entity matching problem for product categories by converting the GS1 Global Product Categorization (GPC) published by GS1 as an ontology and linking it to the Product Knowledge Graph (ProductKG). For the implementation, methods were developed in Python for word embeddings, WordNet, lemmatization, and large language models (LLMs), which then link classes of the GPC ontology with the classes of the ProductKG. All approaches were carried out on the same source data and each provided an independent version of the linked GPC ontology. As part of the evaluation, the quantities of linked class pairs were analyzed and precision, recall, and F1 score for the Food / Breakfast segment of the GS1 GPC taxonomy were calculated. The results show that no single approach is universally superior. LLMs achieved the highest F1-score due to their deep semantic understanding but suffered from lower precision, making them suitable for applications requiring broad coverage. Lemmatization achieved perfect precision, making it ideal for use cases where false matches must be avoided, though at the cost of significantly lower recall. WordNet offered a balanced trade-off between precision and recall, making it a reasonable default choice. Word embeddings, however, performed poorly in both metrics and did not outperform the other methods.

1 INTRODUCTION

Organizing products into standardized taxonomies is essential for e-commerce, supply chain management, and data integration (Aanen et al., 2015). However, different classification systems, such as the Global Product Categorization (GPC) by GS1[1] and the Product Knowledge Graph (ProductKG) (Kümpel and Beetz, 2023), use distinct structures, naming conventions, and levels of granularity. These discrepancies create challenges in aligning product categories, making interoperability between datasets difficult. In the context of the Semantic Web, where data from diverse sources should be meaningfully connected, aligning product taxonomies is crucial for enabling seamless data exchange and integration (Aanen et al. (2015), Christen (2012)).

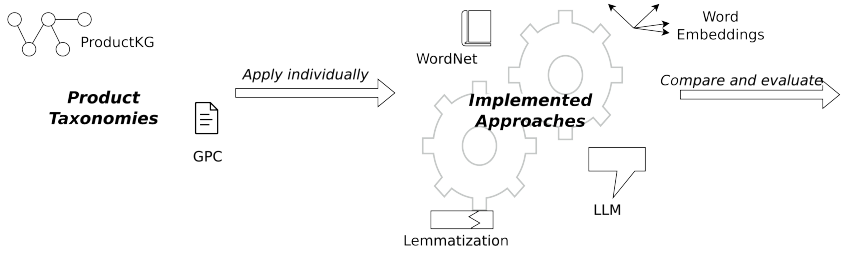
The challenge of aligning product taxonomies is a particular instance of the broader problem of *entity matching*, the task of identifying and linking records that refer to the same real-world entity (Barlaug and Gulla, 2021; Christen, 2012; Elmagarmid et al., 2007; Köpcke and Rahm, 2010), which has existed for as long as databases have been in use. As soon as new datasets, tables, or ontologies are created, organizations face the recurring need to integrate them with existing ones. This challenge has been recognized for decades Elmagarmid et al. (2007); Christen (2012), and despite extensive research, no universal solution has emerged. Entity matching remains a highly relevant problem because every new data source potentially introduces terminological differences, schema variations, or domain-specific nuances that require resolution before meaningful integration is possible.

In product classification, entity matching involves matching categories across different taxonomies, even when they use varying terminologies or hierarchical structures. Ontologies, which define shared conceptualizations of a domain, can provide structured knowledge to support this task by making relationships between product categories more explicit. However, despite significant research in ontology-based and data-driven entity matching, existing approaches still face notable limitations:

[a] https://orcid.org/0000-0002-0408-3953
[1] GS1 website: https://www.gs1.org/

Hagenah, M. and Kümpel, M.
Comparative Analysis of Entity Matching Approaches for Product Taxonomy Integration.
DOI: 10.5220/0013711700004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 2: KEOD and KMIS*, pages 40-51
ISBN: 978-989-758-769-6; ISSN: 2184-3228



	Precision	Recall	F1
Lemmatization	**1.000**	0.263	0.417
LLM	0.653	**0.410**	**0.504**
W. Emb. (x=0.80)	0.882	0.161	0.273
W. Emb. (x=0.75)	0.895	0.183	0.304
WordNet (x=0.33)	0.931	0.290	0.443
WordNet (x=0.40)	0.935	0.312	0.468

Figure 1: Overview of the evaluation workflow: (1) Preparation of source data, (2) Execution of method implementations on same source data.

- Rule-based methods, while interpretable, struggle with complex variations in naming conventions and require extensive manual effort (Cohen et al., 2003).
- WordNet-based approaches leverage predefined lexical relationships but may lack coverage for domain-specific terminology (Agirre et al., 2009).
- Word embeddings and Large Language Models (LLMs) offer more flexibility by capturing semantic similarities, yet they remain prone to errors in cases where product categories have subtle distinctions (Narayan et al., 2022).

These shortcomings highlight the need for a systematic evaluation of different techniques to determine the most effective approach for aligning product taxonomies in a structured and scalable manner. While other works have shown how these approaches can be used for entity matching (e.g. Narayan et al. (2022); Zhang et al. (2024); Peeters et al. (2023); Aanen et al. (2015); Zhu and Iglesias (2018); Jatnika et al. (2019), a lack of a broad comparative analysis is still apparent.

This work presents a comparative analysis of four entity-matching techniques: lemmatization, WordNet-based similarity, word embeddings, and LLMs. Each approach is applied to aligning product categories between GPC and ProductKG, assessing their accuracy and effectiveness in resolving naming and structural inconsistencies. Some of the methods analyzed, such as lemmatization and WordNet, are based on older or more foundational techniques. However, they still offer valuable insights into the core challenges of entity matching. Although more modern and sophisticated approaches exist, evaluating them in depth would exceed the scope of this work. The goal is to establish a solid baseline by comparing a range of fundamental methods, including a more recent LLM-based approach, and to highlight their strengths and weaknesses as a basis for future research.

Based on the current trajectory of research in natural language processing, we hypothesise that

Hypothesis 1.1. *LLMs will perform best in the experiment.*

The broad coverage of LLMs and their ability to capture nuanced semantic relations are likely to give them an advantage, although tendencies to overgeneralize and to generate hallucinations will certainly be limiting factors.

Hypothesis 1.2. *Word embeddings are expected to follow closely.*

Since word embeddings are also trained on large text corpora and can model semantic similarity effectively, they are expected to perform almost as good as LLMs.

Hypothesis 1.3. *WordNet is anticipated to provide decent performance.*

As WordNet encodes structured lexical relationships, it is expected to also perform well. However, its coverage of domain-specific terminology is limited, which leads us to assume that it performs worse than LLMs and word embeddings.

Hypothesis 1.4. *Lemmatization is expected to yield the weakest results.*

Lemmatization merely normalizes words and checks for string matches without deeper semantic reasoning, leading us to assume the worst experiment performance.

The contributions of this work are

1. A systematic evaluation of entity-matching techniques in the context of product classification and the Semantic Web.
2. An analysis of the strengths and weaknesses of the four different approaches in handling naming variations, hierarchical differences, and ontology-based relationships.
3. An experimental validation using real-world product taxonomies to assess matching accuracy and practical applicability.

2 RELATED WORK

The usage of various technologies to solve the Entity Matching problem has been a topic of discussion in various previous works. While prior work has applied these methods in isolation, no systematic head-to-head comparison has been made in the context of product taxonomies. This motivates our comparative study.

2.1 Obtaining Semantic Similarity from WordNet

WordNet (Fellbaum, 2010) has been used to obtain semantic similarity of different words in multiple works. Gurevych and Strube (2004) propose a spoken dialogue summarization method using semantic similarity metrics from WordNet. Their system extracts key utterances by computing the similarity between an utterance and the entire dialogue. Although various works have analyzed the effectiveness of WordNet-based approaches to obtain semantic similarity (Gurevych and Strube, 2004; Meng et al., 2013; Farouk, 2018; Agirre et al., 2009) they do not compare its effectiveness to approaches based on other methods such as word embeddings or large language models. Aanen et al. (2015) present an algorithm that uses WordNet as one of its core components to map different product taxonomies to each other to aggregate product information from different sources. Although their algorithm is specifically made for product taxonomies, they also do not provide any broader comparison of other methods.

2.2 Calculating Semantic Similarity from Word Embeddings

Semantic vector representations of words, referred to as word embeddings, can also be used to obtain semantic similarity of words. By calculating the distance between two vectors using the cosine function, such a value can be retrieved (Farouk, 2018; Zhu and Iglesias, 2018; Jatnika et al., 2019; Kenter and De Rijke, 2015).

Kenter and De Rijke (2015) proposed an alternative way to calculate the similarity of text using word embeddings rather than employing approaches such as lexical matching, syntactical analysis, or handmade patterns.

While Farouk (2018) compares the ability to semantically compare sentences of WordNet and word embeddings using standardises datasets, it does not directly compare their ability to resolve two different identifiers for the same actual entity, which represents the core issue of the entity matching problem.

2.3 Entity Matching with Transformer Based Models

The field of research that analyses the usage of data integration capabilities of language models, whether large or small, has gained a lot of traction in recent years through the rise in popularity of generative AI.

Narayan et al. (2022) presents how general purpose transformer models (Vaswani et al., 2017), in this case GPT-3, can be used for data integration tasks, including entity matching. Despite the fact that such models do not have any task-specific fine-tuning done beforehand, it is found that for each analysed data integration task, the language model outperforms the state-of-the-art solutions for each task.

Peeters et al. (2023) specifically presents the capabilities of large language models when solving the entity matching problem. They compare different large language models against pre-trained language models (PLMs). By using a range of various prompts in zero-shot and few-shot scenarios, the work reveals that there is no single best prompt for a given model or dataset, but rather for a specific model and dataset combination. Furthermore, the quality of the results is very sensitive to prompt variation. They concluded that LLMs outperform PLMs in entity matching tasks in certain scenarios, despite the fact that the PLMs were trained on task-specific data.

Zhang et al. (2024) shows how even small language models can be used to solve the entity matching problem. In order to show that entity matching can be performed while using a significantly lower amount of resources, they created a GPT-2 based model called *AnyMatch*. In order to keep the amount of required resources low, they carefully selected the data they used to train the model for the entity matching task at hand. They then evaluated AnyMatch in a zero-shot environment and found that the resulting F1-Score was only 4.4% worse than the GPT-4 based MatchGPT, despite it using significantly lesser resources.

These works presented how transformer-based language models can be used to perform entity matching. While they do, among other things, also present a comparison to other models, they do not compare the effectiveness to other entity matching approaches that are based on WordNet or word embeddings.

Table 1: Overview of selected related work on entity matching and semantic similarity.

Domain / Focus	Techniques Used
Product taxonomy mapping (Aanen et al., 2015)	WordNet + Rule-based
Dialogue summarization (Gurevych and Strube, 2004)	WordNet similarity
Semantic similarity measures (Meng et al., 2013)	WordNet similarity
Sentence similarity (Farouk, 2018)	WordNet vs. Word embeddings
Entity disambiguation in KGs (Zhu and Iglesias, 2018)	Word embeddings
Word similarity (Jatnika et al., 2019)	Word embeddings (Word2Vec)
Short text similarity (Kenter and De Rijke, 2015)	Word embeddings
Data integration tasks (Narayan et al., 2022)	Transformer-based LLM (GPT-3)
Entity matching (Peeters et al., 2023)	LLMs (GPT-family) vs. PLMs
Resource-efficient entity matching (Zhang et al., 2024)	Small LLM (AnyMatch, GPT-2 based)

3 PRODUCT TAXONOMIES

In this section we introduce the product taxonomies employed in our analysis, detailing their structures, purposes, and relevance to our research objectives. Specifically, we focus on the Global Product Classification (GPC) for its usage as a global standard and its broad coverage of product categories and the Product Knowledge Graph (ProductKG) for its practical applications. These taxonomies serve as the source and target of the different entity matching methods explored in this work.

3.1 The Global Product Classification

The GPC (GS1, 2024a), developed by GS1, is an internationally recognized standard for the systematic categorization of products. Its primary purpose is to provide a common language that enables companies, marketplaces, governmental bodies and other stakeholders to classify products in a consistent and unambiguous manner, thereby facilitating interoperability across supply chains and ensuring that product information can be exchanged without semantic conflicts. By establishing such a standardized framework, GPC reduces inefficiencies in trade processes, data alignment, and regulatory reporting, all of which would otherwise be prone to inconsistencies if organizations relied solely on internal product taxonomies (GS1, 2025b).

GPC is designed as a hierarchical taxonomy consisting of four distinct levels. At the most general level, the **Segment** represents a broad industry sector, such as “Food/Beverage/Tobacco” or “Electronics”. Each Segment is subdivided into **Families**, which group products of similar nature within that sector. Families are then divided into **Classes**, providing further specificity, and the most detailed level is the **Brick**, which clusters closely related products and serves as the operational unit for product identification. Bricks are associated with so-called Brick Attributes, which describe key characteristics of the products within them, such as screen size for laptops or roast type for coffee. For example, a bag of roasted coffee beans would be categorized in the Segment “Food/Beverage/Tobacco,” within the Family “Beverage,” in the Class “Coffee,” and finally in the Brick “Roasted Coffee,” with attributes specifying features such as caffeine content, roast level, or packaging type. This hierarchical and attribute-based structure ensures that trading partners in different countries can describe and exchange information about the same product with precision and without ambiguity (GS1, 2015).

Unlike static taxonomies, GPC is continuously updated through a governance process coordinated by GS1. Industry stakeholders, including manufacturers, retailers, distributors, and regulators, can submit change requests when gaps, ambiguities, or emerging product categories are identified. These proposals are reviewed and discussed in the Global Standards Management Process (GSMP), where consensus-based decisions ensure that modifications are both relevant and implementable (GS1, 2024b). Each release contains a complete XML schema, representing the current state of the taxonomy, as well as an XML delta file that highlights the changes relative to the previous version (GS1, 2015). These updates are then distributed across the GS1 system, including the Global Data Synchronisation Network (GDSN), to guarantee alignment between stakeholders (GS1, 2015).

The integration of GPC into the GDSN highlights its significance in global data exchange. The GDSN is a network of interoperable GS1-certified data pools that allows companies worldwide to exchange standardized and trusted product information in real time (GS1, 2025a). It operates on a publish–subscribe model: a brand owner enters product data once into

a data pool, and all subscribed trading partners automatically receive the same data, eliminating duplication and inconsistencies (Wikipedia, 2024). GPC plays a crucial role within this framework by acting as the categorical backbone against which product data is structured and validated (GS1, 2025c). In practical terms, this ensures that a newly introduced product, will be categorized consistently across the network, rather than being subject to differing local taxonomies.

In sum, the GS1 Global Product Classification provides a universal, hierarchical, and attribute-enriched framework for product categorization that evolves in line with market innovations. Through its integration into the GDSN, it enables seamless, real-time, and semantically coherent product data exchange, underpinning the reliability and efficiency of global commerce.

For the purpose of this work, the GPC dataset was transformed from its XML format into an OWL ontology to support ontological linking and to facilitate its use in the different matching tasks. Each product category from the original dataset was mapped to an `owl:Class`. To preserve the hierarchical structure of the taxonomy, the `rdfs:subClassOf` property was used. Starting from the segment level, every family was linked to its corresponding segment via this property, and the same approach was applied recursively down to classes and bricks.

3.2 The Product Knowledge Graph (ProductKG)

ProductKG(Kümpel and Beetz, 2023) is an open-source product knowledge graph that integrates product information obtained from the Web with environment data provided by a semantic digital twin (semDT) Kümpel et al. (2021). The purpose of this system is to combine abstract product data, such as taxonomies, ingredients, nutritional values, and labels, with information about the physical world in which products exist in specific locations and quantities. In this way, ProductKG enables intelligent applications to reason about products in ways that are both semantically rich and contextually grounded.

The semDT component of ProductKG represents environment information in the form of a spatial and relational model of a physical setting, for example a retail store. Its standardised representation enables omni-channel applications Kümpel and Dech (2025). It represents shelf layouts, product placements, stock levels, and prices, which are often derived from robotic perception systems as described in Beetz et al. (2022). As a result, the semDT provides knowledge about where products are located and how many are available, linking physical product instances to their digital representations.

On the other hand, ProductKG integrates diverse sources of product information. It includes product taxonomies, ingredient classifications that are connected to allergens, product labels such as brands and packaging details, nutritional data and physical dimensions. Structurally, ProductKG is modular and consists of interconnected ontologies that focus on different aspects of product knowledge. Products are interlinked across these ontologies by standardized identifiers, most prominently the Global Trade Identification Number (GTIN), which ensures reliable alignment between web-based product data and environment observations.

The integration of these two knowledge domains makes ProductKG suitable for applications that require both semantic understanding and contextual awareness. Examples include shopping assistants that highlight suitable products on store shelves according to user preferences Kümpel et al. (2023), dietary recommenders that suggest products or recipes based on nutritional information and personal profiles, and cooking assistants that relate available products in a household to recipe steps. ProductKG is exposed through a SPARQL endpoint, which allows external applications to query and combine product characteristics, nutritional attributes, or product hazard information.

By combining web-based product information with a semDT, ProductKG advances the concept of context-aware product reasoning. It allows applications not only to identify that a product is vegan or gluten-free, but also to verify whether such a product is present in the immediate environment, whether suitable alternatives are available, and how it can be incorporated into a dietary plan or recipe. This ability makes ProductKG a valuable resource for intelligent assistants in domains such as retail and household environments.

One of the central components of ProductKG is the *product taxonomy*, which defines a variety of product categories. These categories were extracted from online sitemaps of supermarket websites and are further enriched through integration with existing ontologies such as *FoodOn* (Dooley et al., 2018). Due to its broad coverage of everyday consumer products and its native availability in OWL format, the product taxonomy could be directly reused in this work without the need for further transformation or preprocessing.

4 ENTITY MATCHING

To evaluate the different semantic similarity techniques, each matching approach was implemented as an independent module. All methods operate on the same input data: the set of ontology classes of the converted GPC dataset and the ontology classes obtained from the product taxonomy. The modules produce their output independently, enabling an objective comparison of their performance. Despite their independence, all methods adhere to some predefined rules:

- **At Most One Matched Class:** A GPC class should have, at maximum, only one link to one single class when matching. If there are multiple valid found matches, the best one should be selected.
- **Subclass Inheritance:** If a GPC class is matched to a taxonomy class, all its subclasses are assumed to belong to the same category.
- **Conflict Resolution with Superclasses:** If a superclass has already been matched to a taxonomy class, its subclasses are not allowed to match to that same class. In such cases, the system selects the next-best available and valid match.

Once a match is confirmed, the ontology class is linked to the matched taxonomy class using the `oboInOwl:hasDbXref` annotation property. This ensures that external taxonomy references are properly encoded in the resulting ontology structure.

4.1 Word Embeddings

Word embeddings are vector representations of words in a high-dimensional space that capture semantic relationships based on their usage in large text corpora (Kusner et al., 2015; Almeida and Xexéo, 2023). Training algorithms such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) leverage the distributional hypothesis (Sahlgren, 2008) to learn these representations so that semantically similar words are located near each other in the vector space. This allows for the computation of semantic similarity using cosine similarity (Farouk, 2018; Jatnika et al., 2019)

In this implementation, the pre-trained `glove-twitter-200` model, available through the python package `gensim`, was used (Řehůřek and Sojka, 2010). For product class names consisting of multiple words, embeddings were computed for each individual word and averaged to obtain a single vector representation per label. The similarity between ontology classes and taxonomy entries was then calculated using cosine similarity between these average vectors.

Since cosine similarity ranges from -1 to 1, a threshold must be defined to determine when two product class names are considered a semantic match. To analyze the effect of this parameter, two runs were performed using thresholds of 0.75 and 0.80 respectively. If multiple valid matches are found, then the one with the highest cosine similarity will be selected. The algorithm is detailed in the following:

Algorithm 1: Entity Matching with Word Embeddings.

Require: GPC classes C_{GPC}, ProductKG classes C_{PKG}, pre-trained GloVe model, threshold t

for each $c_g \in C_{GPC}$ **do**
  Compute embedding v_g as mean of word vectors of c_g
  for each $c_p \in C_{PKG}$ **do**
    Compute embedding v_p as mean of word vectors of c_p
    Compute cosine similarity $s = \cos(v_g, v_p)$
    if $s \geq t$ **then**
      Add (c_g, c_p, s) to candidate matches
    end if
  end for
  Select candidate with maximum s for c_g (if any)
  Apply subclass inheritance and conflict resolution rules
  Link $c_g \rightarrow c_p$ with `oboInOwl:hasDbXref`
end for

4.2 WordNet

WordNet is a lexical database that organizes words into sets of cognitive synonyms called *synsets*, which represent distinct concepts. Each synset is linked to other synsets through various semantic relations, including antonymy, hyponymy (subclass), hypernymy (superclass), meronymy (part-whole), and holonymy (whole-part), forming a graph-like structure of semantic relationships (Fellbaum, 2010).

This structure allows the computation of semantic similarity between words based on the shortest path or other distance metrics between synsets (Agirre et al., 2009; Meng et al., 2013). In this implementation, similarity scores were derived from such path-based measures.

As with word embeddings, a similarity threshold must be defined to determine whether two terms are considered a match. To analyze the effect of this threshold, two runs were conducted with values of 0.33 and 0.40. Similarly, if multiple valid matches are detected, the one with the highest value will be

selected. WordNet is accessible as a database and through various programming libraries such as NLTK (Bird et al., 2009).

Similar to Algorithm 1, we compute the WordNet similarity as described in the following:

Algorithm 2: Entity Matching with WordNet Similarity.

Require: GPC classes C_{GPC}, ProductKG classes C_{PKG}, threshold t

for each $c_g \in C_{GPC}$ **do**
  for each $c_p \in C_{PKG}$ **do**
    Compute WordNet path-based similarity $s(c_g, c_p)$
    if $s \geq t$ **then**
      Add (c_g, c_p, s) to candidate matches
    end if
  end for
  Select candidate with maximum s for c_g (if any)
  Apply subclass inheritance and conflict resolution rules
  Link $c_g \rightarrow c_p$ with `oboInOwl:hasDbXref`
end for

4.3 Lemmatization

The lemmatization-based approach performs lexical matching by first normalizing the names of the taxonomy classes. Lemmatization reduces inflected or derived words to their base or dictionary form (e.g., *"running"* → *"run"*), which is particularly useful for improving consistency in string comparison (Khyani et al., 2021).

After lemmatizing all terms, the resulting lemmas are compared to each other. If two terms yield exactly the same lemma, they are considered a match. This method extends simple string matching by making it robust against grammatical variations such as plural forms or verb conjugations (Khyani et al., 2021).

While it does not capture deeper semantic similarity, lemmatization provides an efficient and linguistically grounded baseline for identifying equivalent concepts based on surface forms.

4.4 LLMs

The LLM-based approach leverages the *emergent abilities* of large-scale language models, which appear as models that are scaled with more parameters and data (Zhao et al., 2023; Wei et al., 2022). These models demonstrate a deeper semantic understanding of concepts, enabling them to match class labels based on meaning rather than surface similarity Zhang et al. (2024); Peeters et al. (2023).

Algorithm 3: Entity Matching with Lemmatization.

Require: GPC classes C_{GPC}, ProductKG classes C_{PKG}

Lemmatize all terms in C_{GPC} and C_{PKG}
for each $c_g \in C_{GPC}$ **do**
  for each $c_p \in C_{PKG}$ **do**
    if lemma(c_g) = lemma(c_p) **then**
      Match $c_g \rightarrow c_p$
      Apply subclass inheritance and conflict resolution rules
      Link $c_g \rightarrow c_p$ with `oboInOwl:hasDbXref`
    end if
  end for
end for

Due to the large number of classes in both the GPC dataset and the product taxonomy, computational and memory limitations had to be considered. Instead of prompting the model with every possible class pair, the entity matching task was executed as a bulk operation. We selected the remotely available **GPT-4o** (OpenAI, 2024) for its support of file uploads, allowing a more flexible input format

To simplify processing and reduce syntactic complexity, both ontologies were converted to plain XML. The resulting files were split into smaller chunks to fit within the model's processing limits. For each chunk, the following prompt was used:

- *Match GPC classes to ProductKG taxonomy classes based on semantic similarity, not string similarity*
- *Avoid code generation or syntactic reformulation*
- *Follow the same matching rules used across all implementations (e.g., single best match, subclass inheritance, conflict resolution)*
- *Ensure all matched classes exist in the input to avoid hallucinations*

This process was repeated for all chunks. The results were then merged, and class names were resolved back to their original ontology terms. Valid matches were linked using the `oboInOwl:hasDbXref` annotation property.

5 EVALUATION

This section presents and compares the results of the individual matching approaches.

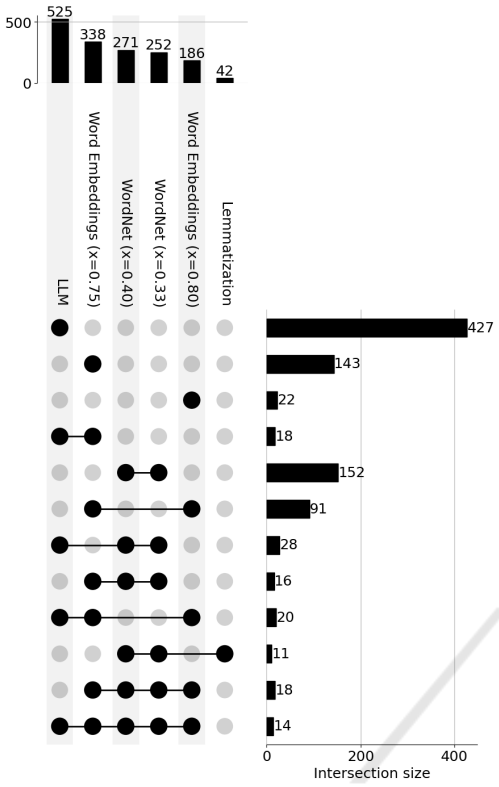


Figure 2: UpSet plot (Lex et al., 2014) illustrating the distribution and intersections of matched class pairs across the different entity matching approaches. Each bar represents the number of matches found either uniquely by a single method or jointly by multiple methods, indicated by the dots. This visualization highlights both agreement and divergence among the implemented techniques. Sets below the size of 8 are not shown here for the sake of readability.

5.1 Quantitative Evaluation

We analyze the matched class pairs from each method and evaluate their overlap using an UpSet plot (Lex et al., 2014). To assess performance more precisely, we calculate precision, recall, and F1 score on a focused subset of the dataset.

Figure 2 visualizes the class pairs matched by each method and their intersections. The LLM-based approach identifies many unique matches, suggesting a broader semantic range. The word embeddings method with a 0.75 threshold also finds distinct matches not shared with the stricter 0.80 setting or other methods.

The two WordNet-based variants yield nearly identical results, indicating low sensitivity to threshold changes (0.33 vs. 0.40). Interestingly, the 0.80 word embeddings configuration produces some unique matches not present at 0.75, likely due to higher similarity scores for more specific terms. Lemmatization, by contrast, contributes few unique matches and mostly overlaps with other methods, reflecting its reliance on surface-level similarity.

While Figure 2 highlights overlaps and differences, it does not indicate which method performs best. For that, we evaluate the approaches on a smaller, manageable subset of the dataset: the `Food/Beverages` segment of GPC. This domain allowed us to manually create a gold-standard mapping for reference.

Based on this reference, we compute standard evaluation metrics: precision, recall, and F1 score. These are derived from the true positives, false positives, and false negatives for each method, as shown in Table 2.

The results reveal clear performance differences. Lemmatization achieves the highest precision (1.0) but has low recall, resulting in a modest F1 score. It identifies only highly accurate matches but misses many valid ones.

The LLM-based method reaches the highest F1 score due to its much higher recall, though it also has the lowest precision (0.653), reflecting a larger number of false positives.

Both word embedding configurations achieve relatively high precision but very low recall, leading to the lowest F1 scores overall.

This reflects a conservative matching strategy that captures only a small subset of correct pairs.

WordNet-based methods offer a more balanced trade-off. With slightly lower precision but higher recall than lemmatization and embeddings, they outperform both in terms of F1 score. Nonetheless, the LLM-based method leads in overall coverage due to its superior recall.

5.2 Qualitative Evaluation

In addition to the quantitative results, we also provide a set of representative examples 3 to qualitatively illustrate the error patterns of the different approaches. These examples were not directly extracted from the evaluation dataset, which reported aggregated counts only, but were instead constructed to reflect the typical strengths and weaknesses observed in the quanti-

Table 2: Results for all approaches.

	Precision	Recall	F1-Score
Lemmatization	**1.000**	0.263	0.417
LLM	0.653	**0.410**	**0.504**
W. Emb. (x=0.80)	0.882	0.161	0.273
W. Emb. (x=0.75)	0.895	0.183	0.304
WordNet (x=0.33)	0.931	0.290	0.443
WordNet (x=0.40)	0.935	0.312	0.468

Table 3: Representative true positives, false positives, and false negatives for each entity matching approach.

Method	True Positive (Correct Match)	False Positive (Wrong Match)	False Negative (Missed Match)
Lemmatization	*"Coffees"* → *"Coffee"*	–	*"Roasted Coffee"* vs. *"Coffee Beans"*
WordNet	*"Bread"* ↔ *"Loaf"*	*"Oil"* ↔ *"Petroleum"*	*"Tofu"* vs. *"Soybean Product"*
Word Embeddings	–	*"Cake"* ↔ *"Biscuit"*	*"Skimmed Milk"* vs. *"Low-Fat Milk"*
LLM	*"Almond Milk"* → *"Plant-based Drinks"*	*"Energy Bar"* → *"Chocolate"*	*"Granola"* vs. *"Breakfast Cereals"*

tative analysis. For instance, lemmatization produces exact lexical matches such as Coffee → Coffee but fails in cases of synonymy, whereas LLMs demonstrate broader coverage (e.g., Almond Milk → Plant-based Drinks) at the cost of more frequent false positives.

The observed error tendencies also have important implications for practical applications of taxonomy alignment. False positives produced by LLMs, such as mapping Energy Bar to Chocolate, may be acceptable in consumer-facing scenarios like recommender systems, where broad semantic coverage is beneficial and occasional overgeneralization does not critically harm usability. However, in compliance-critical contexts such as allergen tracking or regulatory reporting, such overextensions could lead to serious misclassifications and must therefore be avoided. Conversely, the false negatives typical of lemmatization (e.g., Roasted Coffee vs. Coffee Beans) indicate that while this method ensures perfect precision, it risks omitting many valid mappings, limiting its suitability for applications where comprehensive coverage is essential. WordNet and word embeddings fall between these extremes, offering moderate trade-offs but still showing domain-specific weaknesses. Taken together, these qualitative patterns underscore that the choice of method should be guided not only by aggregate scores but also by the specific error tolerance of the intended use case.

5.3 Discussion

The comparative evaluation of the entity matching approaches confirms several aspects of the initial hypothesis while also revealing some unexpected outcomes.

Result 5.1. *The LLM-based method indeed outperformed the others in terms of coverage, identifying the highest number of matches, including unique ones not detected by alternative techniques.*

This demonstrates that large language models can capture subtle semantic relationships beyond lexical or structural similarity, as hypothesized. However, this strength comes at the cost of precision.

Result 5.2. *The results clearly show a tendency of LLMs to overgeneralize, leading to false positives, which aligns with the predicted challenge of hallucinations and overextension.*

Such behavior may still be advantageous in application contexts where broad semantic coverage is desired, for example in **shopping or recommendation systems**, but it introduces risks in scenarios requiring high reliability.

Result 5.3. *Lemmatization, as expected, performed the weakest overall in terms of coverage.*

Its perfect precision highlights that it is highly conservative and produces no false positives, but this comes at the expense of very limited recall.

Result 5.4. *This outcome supports the hypothesis that lemmatization is too restrictive to capture semantically related but lexically different terms.*

This is making it suitable only for use cases where absolute accuracy is more important than flexibility, such as **medical or allergen-sensitive applications**.

Result 5.5. *WordNet delivered results that aligned well with the hypothesis, providing decent performance and a balanced trade-off between recall and precision.*

WordNet consistently outperformed lemmatization by identifying semantically related terms while remaining robust against false positives. The minimal impact of changing the similarity threshold further indicates that WordNet-based similarity offers predictable and reliable behavior, though its limited coverage reflects its restricted lexical scope.

WordNet's predictable balance between recall and precision makes it attractive for lightweight applications where stable performance is more valuable than full coverage, for instance in **smaller-scale taxonomy integration tasks** or as an interpretable baseline in educational and research settings.

Result 5.6. *Contrary to the hypothesis, word embeddings did not closely follow LLMs in performance.*

Despite their potential to capture nuanced similarity through training on large text corpora, the results were significantly weaker than anticipated. Both tested thresholds resulted in low recall, and while the stricter threshold occasionally identified matches missed by the more lenient one, overall effectiveness remained limited.

This underperformance may be explained by domain mismatch, as pre-trained embeddings were not optimized for product taxonomies.

Result 5.7. *This suggests that in the specific domain of product taxonomies, pretrained word embeddings may not capture the necessary semantic granularity or domain-specific knowledge.*

Although pre-trained embeddings underperformed in this study, they may still prove useful in **scenarios where domain-specific retraining is feasible**, or as a candidate generation step in hybrid pipelines that rely on more expressive models for final matching.

Overall, the results partially validate our hypotheses. LLMs demonstrated the broadest coverage and strongest ability to capture semantic relations, though at the expected cost of precision. WordNet and lemmatization behaved largely as anticipated, with WordNet offering moderate effectiveness and lemmatization remaining overly restrictive. The unexpected underperformance of word embeddings indicates that their usefulness in this task may be constrained without domain-specific adaptation.

Result 5.8. *Ultimately, no single method emerges as universally optimal, and the choice of approach depends strongly on application requirements, particularly whether broader coverage or higher precision is prioritized.*

6 CONCLUSIONS AND FUTURE WORK

This work presented a comparative evaluation of four entity matching approaches for linking product categories between the GPC ontology and the ProductKG. Lemmatization, WordNet, word embeddings, and LLMs were implemented independently and assessed based on their ability to detect semantically equivalent classes.

The results show that each method has specific strengths and weaknesses. LLMs achieved the highest F1-score due to their ability to capture deep semantic relationships, but their lower precision indicates a tendency to overgeneralize. Lemmatization yielded perfect precision and is suitable for applications where accuracy is critical, though it struggled with semantically related but lexically different terms. WordNet offered a balanced trade-off, while word embeddings performed poorly in both recall and precision.

No single method proved best in all cases, suggesting that the optimal choice depends on the specific goals of the application.

Future work will include a more detailed analysis of threshold effects for WordNet and word embeddings, as well as an investigation into common patterns among false positives and false negatives. More advanced matching systems will also be explored. In addition, hybrid methods that combine the strengths of different approaches, such as pairing LLMs with lemmatization or WordNet filtering, may improve results. Evaluating the impact of such combinations on F1-score could lead to more effective and practical solutions. Furthermore, analysing the effectiveness of the techniques on other ontology-based datasets could also give more insight on the real world applicability.

ACKNOWLEDGEMENTS

This work was partially funded by the central research development fund of the University of Bremen as well as the German Research Foundation DFG, as part of CRC (SFB) 1320 “EASE - Everyday Activity Science and Engineering”, University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject P1 “Embodied semantics for everyday activities”.

REFERENCES

Aanen, S. S., Vandic, D., and Frasincar, F. (2015). Automated product taxonomy mapping in an e-commerce environment. *Expert Systems with Applications*, 42(3):1298–1313.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL ’09*, page 19, Boulder, Colorado. Association for Computational Linguistics.

Almeida, F. and Xexéo, G. (2023). Word embeddings: A survey. arXiv:1901.09069 [cs.CL].

Barlaug, N. and Gulla, J. A. (2021). Neural networks for entity matching: A survey. *ACM Trans. Knowl. Discov. Data*, 15(3).

Beetz, M., Stelter, S., Beßler, D., Dhanabalachandran, K., Neumann, M., Mania, P., and Haidu, A. (2022). *Robots Collecting Data: Modelling Stores*, pages 41–64. Springer International Publishing, Cham.

Bird, S., Loper, E., and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Christen, P. (2012). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Publishing Company, Incorporated. pages. 12–34.

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pages 73–78.

Dooley, D. M., Griffiths, E. J., Gosal, G. S., Buttigieg, P. L., Hoehndorf, R., Lange, M. C., Schriml, L. M., Brinkman, F. S., and Hsiao, W. W. (2018). Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. *npj Science of Food*, 2(1):23.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16.

Farouk, M. (2018). Sentence Semantic Similarity based on Word Embedding and WordNet. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 33–37.

Fellbaum, C. (2010). WordNet. In Poli, R., Healy, M., and Kameas, A., editors, *Theory and Applications of Ontology: Computer Applications*, pages 231–243. Springer Netherlands, Dordrecht.

GS1 (2015). *Global Product Classification (GPC) Development & Implementation Guide*. GS1. Issue 8, Final, December 2022.

GS1 (2024a). *Global Product Classification (GPC)*. GS1. https://www.gs1.org/standards/gpc.

GS1 (2024b). How is gpc developed and maintained? https://support.gs1.org/support/solutions/articles/43000734258-how-is-gpc-developed-and-maintained-. Accessed: 2025-09-15.

GS1 (2025a). Gs1 gdsn. https://www.gs1.org/services/gdsn. Accessed: 2025-09-15.

GS1 (2025b). How gpc works. https://www.gs1.org/standards/gpc/how-gpc-works. Accessed: 2025-09-15.

GS1 (2025c). How gs1 gdsn works. https://www.gs1.org/services/gdsn/how-gdsn-works. Accessed: 2025-09-15.

Gurevych, I. and Strube, M. (2004). Semantic Similarity Applied to Spoken Dialogue Summarization. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 764–770, Geneva, Switzerland. COLING.

Jatnika, D., Bijaksana, M. A., and Suryani, A. A. (2019). Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science*, 157:160–167.

Kenter, T. and De Rijke, M. (2015). Short Text Similarity with Word Embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420, Melbourne Australia. ACM.

Khyani, D., Siddhartha, B., Niveditha, N., and Divya, B. (2021). An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, 22(10):350–357.

Köpcke, H. and Rahm, E. (2010). Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, 69(2):197–210.

Kümpel, M. and Beetz, M. (2023). Productkg: A product knowledge graph for user assistance in daily activities. In *FOIS'23: Ontology Showcase and Demonstrations Track, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, 19-20 July, 2023, Sherbrooke, Québec, Canada*, volume 3637.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Kümpel, M. and Dech, J. (2025). Semantic digital twins for omni-channel localisation. In *Proceedings of the 11th IFAC MIM Conference on Manufacturing Modelling, Management and Control*.

Kümpel, M., Dech, J., Hawkin, A., and Beetz, M. (2023). Robotic shopping assistance for everyone: Dynamic query generation on a semantic digital twin as a basis for autonomous shopping assistance. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*, pages 2523–2525, London, United Kingdom.

Kümpel, M., Mueller, C. A., and Beetz, M. (2021). Semantic digital twins for retail logistics. In Freitag, M., Kotzab, H., and Megow, N., editors, *Dynamics in Logistics: Twenty-Five Years of Interdisciplinary Logistics Research in Bremen, Germany*, pages 129–153. Springer International Publishing, Cham.

Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., and Pfister, H. (2014). Upset: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992.

Meng, L., Huang, R., and Gu, J. (2013). A review of semantic similarity measures in wordnet. *International Journal of Hybrid Information Technology*, 6(1):1–12.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Narayan, A., Chami, I., Orr, L., Arora, S., and Ré, C. (2022). Can foundation models wrangle your data? *arXiv preprint arXiv:2205.09911*.

OpenAI (2024). Chatgpt (gpt-4o, may 2024 version). https://chat.openai.com. Large language model.

Peeters, R., Steiner, A., and Bizer, C. (2023). Entity matching using large language models. *arXiv preprint arXiv:2310.11244*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In

Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of linguistics*, 20:33–53.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Zhang, Z., Groth, P., Calixto, I., and Schelter, S. (2024). Anymatch–efficient zero-shot entity matching with a small language model. *arXiv preprint arXiv:2409.04073*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zhu, G. and Iglesias, C. A. (2018). Exploiting semantic similarity for named entity disambiguation in knowledge graphs. *Expert Systems with Applications*, 101:8–24.