

Decompose Before You Match: Mapping Complex Terms to FoodOn by Their Parts

Tadeusz Price¹, Jens Ruhof¹, Ido Toxopeus¹, Marga Ocké¹ and Anna Fensel²

¹The National Institute for Public Health and the Environment, Bilthoven, the Netherlands

²Wageningen University & Research, Wageningen, the Netherlands

Abstract

FoodOn is increasingly the reference ontology for making food data interoperable and for grounding food knowledge graphs that support reasoning. Both uses depend on matching dataset terms to ontology classes, and both turn out to be harder than expected. Reusing meat-alternative ingredients data from a comprehensive multi-dataset knowledge graph (NICE-Food KG), we find two problems. Coverage is poor for complex terms, those naming a base food together with a process, state, or form (such as “rehydrated soy protein”), because no food ontology can pre-coordinate every such combination. And even where a class is found, it is weakly axiomatised, rarely carrying more than `rdfs:subClassOf`, leaving a reasoner little to work with. We show that decomposing a complex term into a base plus modifiers and matching the parts improves matching rate, and the matched classes are better axiomatised. This shifts food knowledge graph construction from pre- to post-coordination, and we present the first two modules of a pipeline, a decomposer and a matcher, that automate the step with an Large Language Model (LLM) and ontology search.

Keywords

ontology matching, complex term mapping, knowledge graph construction, FoodOn, large language models

1. Introduction

Food data is fragmented. Information about products, ingredients, and chemicals lives in many separate datasets and domain perspectives, from nutrition and chemical safety to sustainability and life-cycle assessment, collected for different purposes at varying granularities. Integrating these datasets would let us study food as a whole and answer questions of clear societal importance, such as how a shift toward plant-based diets affects nutrition, safety, and environmental impact together. Shared ontologies are gaining traction as the means to achieve interoperability. Food ontologies broadly have two major roles: (1) providing a standardised semantic layer for interoperability across heterogeneous food data sources, and (2) serving as the schema/foundation for food knowledge graphs that support richer representation and reasoning-driven applications (for example Haussmann et al. [1]) [2]. In the food domain, FoodOn [3] is the major general-purpose ontology.

Linking a dataset’s terms to ontology classes is an ontology matching problem, and most matching seeks one-to-one equivalences. Recent work to build a cross-domain food knowledge graph, NICE-Food KG [4], found this harder than expected and ran into two limits at once. Coverage was low, with many dataset terms finding no adequate FoodOn class. And the classes that were matched were thinly axiomatised, carrying little beyond `rdfs:subClassOf`, so their weak axiomatisation limited the reasoning and inference the graph could support. The first limit constrains interoperability, the second constrains reasoning.

Both limitations are most pronounced for complex terms: entries that name a base food together with one or more modifiers giving a process, state, fraction, or form, for example “rehydrated soy protein”. Complexity here is a property of the term, not the food: “lasagna” is a simple term for a complex dish, while “fried onion” is a complex term for a simple food. FoodOn does enumerate some compounds as pre-coordinated classes, but very unevenly, with dozens of variants for a few foods and none for most. Representing a complex term faithfully then means either finding one pre-coordinated class for the

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

✉ tadeusz.price@rivm.nl (T. Price)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

whole term, or post-coordinating it from a class for the base food plus classes for its modifiers, related by axioms (cf. Cimino [5] for a clear discussion from the medical domain). The second is a complex, non-equivalence correspondence [6]: more general, but harder to produce automatically. This tension is not specific to food, and complex correspondences are a recognised hard case in ontology matching [6].

Reusing the meat-alternative ingredients that NICE-Food KG matched to FoodOn, we measure coverage and semantic richness for simple and complex terms under direct one-to-one matching, and then test whether decomposing each complex term and matching its parts does better on both. We then present an early prototype that automates this decompose-then-match step with an LLM and ontology search, and outline a path toward post-coordinated, semantically rich food knowledge graphs building on these findings.

2. Matching complex terms: two measurements

2.1. Measurement 1: matching whole terms

In previous work for NICE-Food KG [4], 132 ingredients of meat alternatives on the Dutch market were matched directly to FoodOn classes, using embeddings-based retrieval with manual selection. This produced a list of ingredients with their closest matching class where found.

We took 118 (after deduplication and scoping) of those ingredients and annotated them with:

- is this a complex term, describing a base food and some modifier to it (e.g. rehydrated pea protein)
- an informal axiomatic richness score:
 - poor: the class is primitive, asserting only `rdfs:subClassOf` parents that themselves carry no relevant axioms; nothing beyond taxonomy a reasoner could use
 - medium: at least one logical relationship beyond subsumption (e.g. a `derives_from` or quality restriction)
 - rich: approaches a full definition, carrying most expected relationships (organismal origin, parts, qualities)

Additionally, for 29 of the ingredients a better match was selected due to updates to FoodOn and manual search instead of automated retrieval in the original mappings.

Table 1 shows the results of analysing the annotated data, showing the connection richness and coverage for simple and complex terms.

Table 1

Coverage and semantic richness of FoodOn matches for simple and complex terms (column-normalised proportions, n=118).

semantics	simple	complex	All
poor	0.37	0.38	0.38
medium	0.27	0.26	0.26
rich	0.34	0.08	0.23
no adequate match	0.01	0.28	0.13

The table shows two gaps between simple and complex terms. On coverage, complex terms have no adequate class 28% of the time compared to 1% for simple terms. On richness, only 8% of complex terms match to a rich class against 34% of simple ones. Coverage (how many terms get a match) is important for interoperability applications, while when using the ontology as a schema for knowledge graph construction, the differences in axiomatic richness are also important.

2.2. Measurement 2: matching the parts

The complex terms for foods were then selected for further analysis: each was separated into two or more terms, most commonly a base food term and a modifier implying some state or processing of

the base food. These terms were then individually mapped to FoodOn, and the semantic richness of the matched terms was once again recorded. Table 2 is analogous to Table 1, and shows coverage and semantic richness split by base versus modifier term.

Table 2

Coverage and semantic richness of FoodOn matches for decomposed terms, by term type (base vs. modifier), after de-duplication (column-normalised proportions, n=53 unique terms).

semantics	base	modifier	all
poor	0.12	0.46	0.30
medium	0.36	0.14	0.25
rich	0.52	0.29	0.40
no suitable match	0.00	0.11	0.06

The aggregated numbers (all) show a similar distribution to non-complex terms in Table 1. Grouping by type of term (base/modifier) clearly shows that the base terms have much better (perfect in this sample) coverage than the modifier terms, and are also more semantically rich than modifier terms or the complex terms themselves. In total, 92% of the original complex terms were fully matched through the matching of their component (base+modifiers) terms, compared to 72% matched directly to pre-coordinated classes in the previous experiment. The failed 4 matches are caused by 3 terms not being matched to FoodOn classes: protein isolate, starch modification and caramelisation.

These results have implications for using FoodOn both for reasoning and for interoperability. For interoperability, the gain is more matches, and fewer additional IRIs need be created given that out of the 107 terms extracted by decomposition, only 53 were unique. There are vastly more combinations of base foods and modifiers than the cardinality of the combined set. This means that curating these base terms would be more sustainable than maintaining all complex terms as pre-coordinated classes. The base terms of FoodOn are already better axiomatised than its pre-coordinated complex classes.

However, a major issue is that this complicates automatic term-matching. Much work has been done so far on one-to-one matching, and complex matching is recognised as a harder problem [6]. The matching of a complex term decomposed into sub terms has three challenges. First, there are multiple ways to decompose a term, second, for the more primitive terms there are often many suitable candidates, and choosing between them requires understanding the ontology’s modelling conventions, which is difficult to automate, and finally, the right object properties / relations must be chosen between the terms. We intend to work in the direction of developing an automated KG construction pipeline that decomposes, matches and adds the right connections to the terms. Here we show initial results of the first two modules of this pipeline, and outline ideas for further work.

3. Automating decompose-then-match

Due to the size of the datasets involved, automatic or semi-automatic pipelines are a necessity, and are common for the term-to-class matching problem (for example [7]). For post-coordinated matching like the decompose-then-match method described above, the problem is different. Molfetta et al. [8] recently presented an approach for FoodEx2, a regulatory taxonomy of food which is made up largely of base foods and facets. When working with FoodOn, the problem is more general, as the space of possible bases and facets is not strictly bounded, as in FoodEx2.

We implement two modules of a pipeline: a term decomposer, and a term-matcher, which work together to automatically perform the task the annotator had in experiment 2 - decompose a complex ingredient-term into a base and modifiers, and match those to FoodOn classes. The two modules perform the following:

- Step 1: an LLM takes a term, and if it is a complex term it is decomposed into sub-terms following a set of rules,
- Step 2: Each decomposed term is matched separately to a FoodOn class in the following manner,

- 2a: A search using the OLS4 API is carried out [9], to find search results in FoodOn for the term that is being matched. Search-term expansion is used: for a term t (e.g. *soy protein*), three queries are issued: (1) the term suffixed with “food product” (*soy protein food product*), (2) the bare term (*soy protein*), and (3) the term in double quotes (“*soy protein*”), which targets exact matches,
- 2b: A list of candidate terms is assembled by taking the top 3 results of the first search, top 5 of the second and top 3 of the third, giving a list of up to 11 items,
- 2c: The term, its context, and the list of candidate classes (with synonyms, descriptions, labels and parent classes) are passed to an LLM (GPT-4.1) prompted with matching instructions to select the most appropriate class,
- 2d: The result for the original complex term is returned: its decomposition into base and modifiers, each matched to a FoodOn class.

The modules were evaluated separately against the annotated data from measurement 2, that is, the decompositions and matches the reference annotator produced when building the corpus for the earlier experiments. The decomposer was scored on whether it split a complex term into at least two sub-terms and gave each the correct base or modifier label, and the matcher on the FoodOn class it selected. Evaluation here is not straightforward, and Table 3 reports three measures: exact agreement with the reference, an ambiguity-aware judgement by the reference annotator, and an independent blind judgement by a second annotator. The strictest is agreement, exact alignment with the reference: the pipeline produced the same decomposition, or chose the same class, as the annotator had when the corpus was built. On this measure the decomposer agrees on 56% of terms and the matcher on 51%.

Table 3

Performance of the decomposer and matcher modules under three measures. Agreement is exact alignment with the reference annotator’s decompositions and matches. Judged accuracy additionally counts as correct the pipeline outputs that the reference annotator, on reviewing the disagreements, found acceptable despite differing from their own. The last two columns give the share of outputs a second, independent annotator judged acceptable when shown the pipeline’s and the reference’s answers blind and in randomised order.

Component	n	Agreement	Judged accuracy	Second rater	
				pipeline	reference
Decomposition	50	0.56	0.76	0.98	0.82
Matcher	53	0.51	0.75	0.77	0.87
base terms			0.80		
modifier terms			0.71		

Inspecting the disagreements showed that many were not necessarily errors. For a lot of terms there is more than one sensible way to split a base from its modifiers, and more than one defensible FoodOn class, so the pipeline and the reference often differ without either being wrong. To get a quantitative sense of how much of the exact-agreement gap is ambiguity rather than mistakes, the reference annotator went back over every disagreement (22 decompositions, 26 matches) and marked whether the pipeline’s answer was acceptable even where it differed from their own. Counting those as correct as well gives the Judged accuracy column: 76% for decomposition and 75% for matching, the latter splitting into 80% for base terms and 71% for modifiers.

Because the disparity between raw agreement and judged accuracy was large, a second annotator was brought in, and to keep the comparison fair they judged the reference annotator’s answers alongside the pipeline’s rather than the pipeline alone. On every disagreement both answers were shown side by side in randomised order with no indication of their source, and each was marked acceptable or not. Treating the items on which pipeline and reference coincided as accepted by both, the second rater judged the pipeline acceptable for 98% of decompositions and 77% of matches, and the reference for 82% and 87%. That even the reference is accepted only about four times in five confirms real ambiguity in the task and not a weakness of the pipeline: for many terms several answers are defensible, and the

reference is one of them rather than the only correct one. For decomposition the second rater accepted the pipeline more often than the reference corpus decomposition. For matching the pipeline lags behind the reference. Model abstentions are the main cause of the discrepancy: of the 12 components the second rater did not accept from the pipeline, 10 are abstentions and 2 are a class chosen wrongly.

Exact agreement understates the pipeline because the task is ambiguous: for a complex term there are usually several acceptable decompositions, and several plausible classes in an ontology as large as FoodOn. Stricter annotation guidelines would raise exact agreement and make the task easier to score, but forcing a single canonical decomposition is not obviously the right goal, and the ambiguity is not in itself an issue. A qualitative look at the residual errors suggests that tighter prompts could remove most of them, and iterating on both modules by examining failure cases would likely give strong performance. Finally, the matcher assigned no class rather than forcing one whenever no candidate was adequate, which makes its errors easy to locate and signals resilience against hallucination.

4. Conclusion and future work

Matching real datasets to FoodOn is problematic in two ways that impact the ontology’s applied utility: complex terms often have no adequate pre-coordinated class, which limits interoperability, and the classes that do match are thinly axiomatised, which limits reasoning. On a set of meat-alternative ingredients we showed that decomposing complex terms and matching their parts addresses both. Coverage rises and the base classes carry more usable axioms. This is a move from pre- to post-coordination which will enable the use of food ontologies across the granularity spectrum.

The cost is harder automation, since decomposition, part matching, and relation assignment each admit several valid answers. We reported a decomposer and a matcher as the first two modules of a pipeline, and plan two more: a relation selector that picks the object property linking each modifier to its base (e.g. `derives_from`, `output_of`), and an entity constructor that builds the graph node for the term: a data individual typed to the matched base class and linked by the selected relations to its modifier classes and processes, without minting a new class. We expect that generative steps will need pairing with hard constraints such as SHACL for output reliable enough to build on, and see this as a realistic path toward semantically richer food knowledge graphs, with processing as a particular focus.

A limitation of the current work is the small and domain-specific evaluation set. As a next step we will deploy and evaluate this approach on a large random sample from a branded food dataset, providing a broader and less curated test of the pipeline. This will also serve as the first step toward constructing a knowledge graph that integrates multiple datasets in the food domain.

Acknowledgements

This work was part of the BIGFOOD project and funded by the Strategic Program of the National Institute for Public Health and the Environment (RIVM), grant number S/133030/01.

Online resources

The pipeline modules (decomposer and matcher), including the LLM prompts with decomposition and matching rules, and the annotated evaluation data are available at <https://github.com/rivm-syso/matching-complex-food-terms>.

Declaration on Generative AI

The authors used ChatGPT for assistance in programming and help with LaTeX formatting. All code was checked and its behaviour validated by the authors. The authors take full responsibility for the integrity of the submitted work.

References

- [1] S. Haussmann, O. Seneviratne, Y. Chen, Y. Ne’eman, J. Codella, C.-H. Chen, D. L. McGuinness, M. J. Zaki, Foodkg: a semantics-driven knowledge graph for food recommendation, in: International Semantic Web Conference, Springer, 2019, pp. 146–162.
- [2] W. Min, C. Liu, L. Xu, S. Jiang, Applications of knowledge graphs for food science and industry, *Patterns* 3 (2022).
- [3] D. M. Dooley, E. J. Griffiths, G. S. Gosal, P. L. Buttigieg, R. Hoehndorf, M. C. Lange, L. M. Schriml, F. S. Brinkman, W. W. Hsiao, Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration, *npj Science of Food* 2 (2018) 23.
- [4] F. Bindt, M. Ocké, R. de Jonge, I. Toxopeus, A. Fensel, Nice-food kg: A knowledge graph for the analysis of the nutritional, ingredient, contaminant, and environmental characteristics of food for food system research, *Future Foods* (2026) 100935.
- [5] J. J. Cimino, Desiderata for controlled medical vocabularies in the twenty-first century, *Methods of information in medicine* 37 (1998) 394–403.
- [6] R. Amini, S. S. Norouzi, P. Hitzler, R. Amini, Towards complex ontology alignment using large language models, in: International Knowledge Graph and Semantic Web Conference, Springer, 2024, pp. 17–31.
- [7] R. Stojanov, I. Kocev, S. Gramatikov, G. Popovski, B. K. Seljak, T. Eftimov, Toward robust food ontology mapping, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 3596–3601.
- [8] L. Molfetta, A. Cocchieri, S. Fantazzini, G. Frisoni, L. Ragazzi, G. Moro, Feast: Retrieval-augmented multi-hierarchical food classification for the foodex2 system, *arXiv preprint arXiv:2603.03176* (2026).
- [9] J. McLaughlin, J. Lagrimas, H. Iqbal, H. Parkinson, H. Harmse, Ols4: A new ontology lookup service for a growing interdisciplinary knowledge ecosystem, 2025. URL: <https://arxiv.org/abs/2501.13034>. *arXiv:2501.13034*.