FoodTaxo: Generating Food Taxonomies with Large Language Models

Pascal Wullschleger[◇,†], **Majid Zarharan**[◇], **Donnacha Daly**[†]
Marc Pouly[†], **Jennifer Foster**[◇]

[◇] ADAPT Centre, School of Computing, Dublin City University
[†] Lucerne School of Computer Science and Information Technology (HSLU)
pascal.wullschleger@hslu.ch

Abstract

We investigate the utility of Large Language Models for automated taxonomy generation and completion specifically applied to taxonomies from the food technology industry. We explore the extent to which taxonomies can be completed from a seed taxonomy or generated without a seed from a set of known concepts, in an iterative fashion using recent prompting techniques. Experiments on five taxonomies using an open-source LLM (Llama-3), while promising, point to the difficulty of correctly placing inner nodes.

1 Introduction

In the food technology industry, taxonomies play a crucial role in business processes related to generation of new consumer and industrial recipes and the adaption thereof to new culinary trends, diets, and sustainability goals. By replacing ingredients in recipes, one can accommodate allergies and dietary restrictions, lower the carbon footprint, react to supply-chain issues, respect seasonality and avoid food waste. The replacement process can, however, be very complex. Veganizing a dessert or cake recipe by replacing eggs influences the entire cooking process. Likewise, changing the type of nuts in a convenience food recipe can have far-reaching consequences for the whole production line, e.g. due to a different fat percentage.

To address these challenges, we investigate the automated generation and completion of taxonomies, i.e. learning taxonomies from data, adding new concepts to existing taxonomies with no human involvement, thereby scaling taxonomies beyond what can be managed by human experts.

Classical taxonomy completion typically involves extracting concepts from a corpus. However, we suggest that it is often more practical to start with a set of known concepts and extend the set while establishing taxonomic relationships. We hypothesize that taxonomies can be iteratively generated using LLMs, without the need for traditional concept extraction (see Fig. 1). This is supported by the state-of-the-art performance of in-context learning with LLMs across a range of natural language processing (NLP) tasks, even without the need for fine-tuning, e.g., (Zhang et al., 2023; Milios et al., 2023). Such an approach is particularly advantageous in situations where it is challenging to provide a suitable corpus for concept extraction.

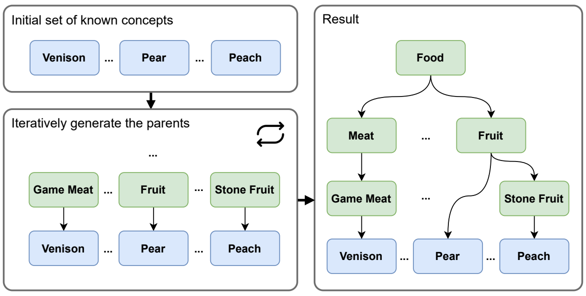

Figure 1: The basic intuition behind the generation process. We start with a set of known concepts and iteratively construct a taxonomy in a bottom-up procedure by prompting large language models (LLMs).

We evaluate our proposed method first on the task of taxonomy completion, before later using it to generate taxonomies without seed relations. In addition to gold-standard comparisons, we rely on recently introduced reference-free metrics which evaluate the robustness and logical adequacy of generated taxonomies (Wullschleger et al., 2025).

In summary, the contributions of this study are novel LLM-based algorithms for 1) taxonomy completion and 2) taxonomy generation given a set of potentially incomplete known concepts. In a comparison to state-of-the-art methods on five taxonomies, we demonstrate the potential of these algorithms for food-related and other taxonomies. Our implementations and datasets are publicly available on GitHub to ensure reproducibility[1].

2 Related Work

The task of taxonomy expansion was introduced as adding leaves to an existing taxonomy (Shen et al., 2018; Fauceglia et al., 2019; Shen et al., 2020; Yu et al., 2020; Manzoor et al., 2020; Ma et al., 2021; Margiotta et al., 2023). However, Zhang et al. (2021) later argued that this is problematic, since it assumes that all newly extracted concepts are hyponyms of existing leaves in the taxonomy. To overcome this assumption, they present a triplet-matching approach, where they predict placements of query concepts as triplets of the form (parent, query, child). This new approach, termed taxonomy

[1] https://github.com/wullli/foodtaxo

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 784–803
July 28-30, 2025 ©2025 Association for Computational Linguistics

completion, allows for new concepts to be included as either hyponyms or hypernyms of existing concepts.

Zeng et al. (2021) formulate an extension to the taxonomy completion task whereby hypernym-hyponym pairs are not explicitly estimated, but candidate positions that require the addition of a new concept are identified. They argue that new concepts should not be extracted, but rather generated, since they can be rare and hard to extract in large text corpora. They initially predict the position in the taxonomy where a concept is missing, and subsequently generate the name of the concept given its position.

In contrast to Zeng et al. (2021), our method does not require a seed taxonomy for training, making it applicable to generating taxonomies solely based on a set of known concepts. We make use of LLMs to generate and place concepts, whereas they train a gated recurrent unit (GRU)-based decoder on the seed taxonomy to generate the names of concepts.

Xu et al. (2023) show few-shot prompting for taxonomy completion to be subpar to their prompt learning method (TacoPrompt). However, aside from few-shot examples, and in contrast to our proposed approach, they do not provide the model with relevant parts of the taxonomy as context. We compare to TacoPrompt in Section 4.

Chen et al. (2023) construct a taxonomy by determining hypernym-hyponym relationships among a set of concepts provided to an LLM, demonstrating that prompt-based methods surpass fine-tuning, particularly as the size of the training taxonomy decreases. However, given the different setting, i.e. constructing a taxonomy using a complete concept set, a direct comparison with our approach is challenging.

3 Methodology

3.1 Problem Definition

Following Zeng et al. (2021), a taxonomy $\mathcal{T} = (\mathcal{E}, \mathcal{V})$ is a directed acyclic graph with edges $(c_p, c_s) \in \mathcal{E}$ pointing from a parent vertex $c_p \in \mathcal{V}$ to a child vertex $c_s \in \mathcal{V}$. In the context of taxonomies, vertices are referred to as *concepts*. Edges represent hypernym-hyponym relations, where the child concept is the least detailed but different specialization of the parent concept.

Unlike traditional approaches (Shen et al., 2020; Manzoor et al., 2020; Zhang et al., 2021; Xu et al., 2023) that assume a complete set of new concepts $\mathcal{Q}$ to be added to $\mathcal{T}$ to obtain a new taxonomy $\mathcal{T}' = (\mathcal{E}', \mathcal{V} \cup \mathcal{Q})$, we assume $\mathcal{Q}$ to be incomplete and allow for the generation of new concepts. Instead of starting with a fixed concept extraction process, we initialize $\mathcal{Q}$ with an incomplete set of known concepts (often leaves) that we want to categorize and iteratively insert into the taxonomy with new concepts generated as needed.

Shen et al. (2020), Manzoor et al. (2020), Zhang et al. (2021), and Xu et al. (2023) assume for simplicity that adding a concept is independent of the attachment of other concepts, resulting in the irrelevance of the order of concept insertion. We observe that we can formulate the task of taxonomy generation as a recursive taxonomy completion task, where we remove the above independence assumption. We start from an initial seed taxonomy $\mathcal{T}_0 = (\{\}, \mathcal{V} = \mathcal{Q} \cup \{p_l, p_r\})$ and iteratively predict placements for each $c \in \mathcal{V}$. A placement is a triplet (c_p, c_q, c_s), where c_q is the query concept that is placed as a child of c_p and as a parent of c_s. Following Manzoor et al. (2020), we add a pseudo-leaf p_l and pseudo-root p_r to T to allow insertion of concepts without parents or children. This means that if c_q is inserted as a leaf, c_s will be the pseudo-leaf node, and if c_q is the root, then c_p is the pseudo-root. Note that c_p can be either an existing concept in $\mathcal{Q}$ or a generated concept. If c_p does not exist in $\mathcal{Q}$, we add it and predict its placement as well, thereby constructing the taxonomy in a bottom-up fashion using completions (Fig. 1).

3.2 Completing Taxonomies

When completing a taxonomy, it, by definition, grows. Due to this, one cannot simply encode the whole tree into a string and use it as context in an LLM, since a ceiling for sequence length would eventually be reached. Instead, we make use of well established techniques, such as chain-of-thought prompting (Wei et al., 2022) and retrieval augmented generation (RAG) (Lewis et al., 2020) as an initial retrieval step to provide the model with only the most relevant part of the taxonomy in order to insert the current query concept. For this purpose, we rely on the demonstrate-search-predict (DSP) paradigm (Khattab et al., 2023a).

The algorithm can be summarized as follows: for each concept, $q \in \mathcal{Q}$, to insert:

1. Retrieve the most similar edges (parent, child) to q based on cosine similarity using FastText embeddings[2] (Bojanowski et al., 2017).[3]

2. Using chain-of-thought (CoT) prompting, retrieve potential candidates for parent concepts of q. In the completion case, these concepts are required to be in the set of existing seed (training) concepts. In case they are not, we repeat the prompt with additional information that the proposed concepts are not valid predictions. We call this backtracking. In the generation case, we allow the model to invent non-existent concepts.

3. Subsequently retrieve the existing children of the proposed parents and again apply CoT prompting to decide which of these children should be attached to the inserted concept.

4. Return all predicted placements as triplets of the form (parent, query, child).

For more detail, see Algorithm 1 in the Appendix.

[2] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.bin.gz

[3] For more detail on how the edges and concepts are encoded as strings, refer to the prompts in Appendix 3.4.

3.3 Generating Taxonomies

We generate a taxonomy without a seed by initializing $\mathcal{Q}$ with a set of known concepts. These are the concepts we want to be able to classify using the taxonomy. Imagine a dataset of cooking recipes. We might want to classify all ingredients into a taxonomy to enable us to easily substitute an ingredient with one of its siblings. However, the set of concepts is unlikely to be complete when it is only initialized with ingredients. Broader concepts, such as *dairy* will presumably not appear as an ingredient. Due to this, our model should predict possibly non-existent parents and children for known concepts, which will be added to the set and subsequently sent to the model for insertion into the current taxonomy. Thereby we construct a taxonomy in a bottom-up procedure. The following is a summary of the steps involved in the algorithm. For a more detailed description refer to Algorithm 2 in the Appendix.

1. Initialize $\mathcal{Q}$, the set of concepts to insert, with all currently known concepts.

2. Sample 100 nodes from $\mathcal{Q}$ and let the LLM write a paragraph on what a potential taxonomy could look like (see, for example, the *Taxonomy Description* in App. B.2.7).

3. While $\mathcal{Q}$ is not empty, do the following.
   (a) Perform the steps described in Section 3.2 for the completion case to insert $q \in \mathcal{Q}$ into the current taxonomy.
   (b) If a new concept is generated, add it to $\mathcal{Q}$.
   (c) Update the taxonomy by inserting all predicted placements into the taxonomy graph.
   (d) Remove q from the set of concepts, $\mathcal{Q}$.

We may not end up with a single root node in the taxonomy. In such cases, all concepts without parents are attached to a pseudo-root. Ideally, the model predicts the pseudo-root as the parent of root nodes, providing a natural stopping criterion.

3.4 Prompts

We show handcrafted prompts for generating parent (Listing 1) and child (Listing 2) concepts of a query. We optimized these prompts by manual trial and error on the validation data. For spacing reasons, we display only two lines of context for both prompts and remove double new-lines. Note that the prompts for generation are slightly different. For a complete impression of prompts and model outputs, please consult Appendix B.2.

3.5 Verifying Placements

We notice that sometimes an LLM will mistakenly interpret the meaning of a parent-child relation as a similarity relation and simply generate predictions of related parent or child concepts on the basis of their similarity. This behaviour persisted even with careful instructions. To mitigate this, we require that the description

~~~~
// Prompt
Which are the most specific parent concepts of the
 given child concept in a taxonomy considering the
 context?
In your reasoning, state how the parent concepts are
  a supertype of the child concept.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Child: Child concept (subtype) that you need to
 place in a taxonomy.

Description: Description of the child concept.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the parents}. We ...

Interpretation: Description of the child concept in
 relation to the context taxonomy. Infer what is
 meant by the child concept from the context.

Parents: Comma separated list of one or more parents
  of the child concept. Valid parents are in the
 context. If there are no suitable parents, return
 None.

---

Context:
```liqueur, sambuca
sugar, sugarloaf
sweet, hardbake
food, comestible
wine, riesling
liqueur, galliano
irish, poteen
flavorer, sassafras
dish, bitok
wine, sauterne
dish, kishke
starches, bap
condiment, chowchow
liqueur, pernod
wine, dubonnet
feed, eatage
cider, scrumpy
dish, rijsttaffel
wine, tokay
...
```

Child: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Reasoning: Let's think step by step in order to
~~~~

~~~~
// Output
Reasoning: Let's think step by step in order to find
  the parents of the child concept "sweetening". We
 can infer that sweetening is related to food and
 its taste, so we need to find the concepts in the
 context that are related to food and taste.

Interpretation: The description of the child concept
  "sweetening" implies that it is an additive that
 enhances the sweetness of food, which is a type of
 comestible.

Parents: flavorer, condiment
~~~~

Listing 1: Parent generation CoT prompt for the completion task. In the generation settings, this prompt looks slightly different (see Appendix B.2)

~~~
// Prompt
Which of the candidates are child concepts (subtypes
 ) of the given parent concept (supertype) in a
 taxonomy?
The context shows existing parent and child concepts
  and whether the children are leaves.
In your reasoning, state how the parent concept is a
  supertype of the selected child concepts.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Candidates: Candidate children of the concept
 separated by commas to select from.

Parent: Parent concept that you need to place in a
 taxonomy.

Description: Description of the parent concept.

Interpretation: Description of the child concept in
 relation to the taxonomy.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the children}. We ...

Leaf: Whether the parent concept should be added as
 a leaf (has no children). Answer with Yes or No.

Children: Comma separated list of candidates that
 are children of the parent concept in a taxonomy.A
 child concept must be a type of the parent concept.
 Separate with commas.

---

Context:
```
...
```

Candidates: salsa, cranberry sauce, dip, soy sauce,
 wasabi, vinegar, spread, duck sauce, chutney,
 marinade, mustard, sauce, mint sauce, green olive,
 pickle relish, black olive, chowchow, pickle

Parent: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Interpretation: The description of the child concept
  "sweetening" implies that it is an additive that
 enhances the sweetness of food, which is a type of
 comestible.

Reasoning: Let's think step by step in order to
~~~

```
// Output
Reasoning: Let's think step by step in order to
 identify the children of the concept "sweetening".
 We can see that sweetening is something added to
 foods to make them taste sweeter, which implies
 that it is a type of additive or condiment.
 Therefore, the child concepts should be types of
 sweetening agents.

Leaf: No

Children: sugar, salsa, cranberry sauce, dip, soy
 sauce, wasabi, vinegar, spread, duck sauce, chutney
 , marinade, mustard, sauce, mint sauce, pickle
 relish
```

Listing 2: Child generation CoT prompt. Prompts are identical for the completion and generation cases.

of the concept (premise) entails the relation (hypothesis) when passed through an natural language inference (NLI) model for all predicted children. For the proposed parent concepts, the verification process is more lenient, requiring only the absence of contradictions. We examine the effect of this step in Section 5.3.

3.6 Backtracking

To ensure valid model outputs, we re-prompt the models at most three times in case outputs do not fulfill constraints. For example, in the completion case, we require all predictions to be valid concepts that exist in the taxonomy, while this is disabled in generation mode so that we can generate suitable missing concepts. More specifically, we leverage the backtracking functionality provided with the DSPy library (Singhvi et al., 2024) in case any of the following assertions fail.

1. The model predicts the query to be its own parent or child.
2. The model predicts non-existent parent concepts (completion case only).
3. The model predicts non-existent child concepts.
4. Parents are predicted, but none of them pass the NLI-verification. This does not apply if the model predicts the pseudo-root as a parent.
5. Children are predicted, but none of them pass the NLI-verification. This does not apply if the model predicts the pseudo-leaf as a child.
6. The concept consists of six or more words.
7. The model predicts children for a concept that are not actually present in the list of candidate children.

4 Completion Experiments

4.1 Data

For benchmarking our completion approach, we follow Xu et al. (2023) and Wang et al. (2022) by evaluating on the SemEval-Food, SemEval-Verb and MeSH datasets. SemEval-Food is the largest taxonomy of the SemEval-2016 Task 13, that was used to evaluate taxonomy extraction methods for a given corpus (Bordea et al., 2016). SemEval-Verb is based on WordNet 3.0 (Fellbaum, 2010) and featured in the SemEval-2016 Task 14, which concerned evaluation of taxonomy enrichment approaches (Jurgens and Pilehvar, 2016). MeSH is a hierarchically organized vocabulary of medical terms (Lipscomb, 2000).

Additionally, we extract a taxonomy from Wikidata[4] by selecting the data-item Food (Q2095) as the root node and extracting all children using the relations *subclass of*, *instances of* and *subproperty of* (Wikidata identifiers P279, P31 and P1647). Lastly, we leverage a proprietary taxonomy provided by a large food market chain that is also being used for recipe development by

[4] https://www.wikidata.org/

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	D	$\lvert$L$\rvert$	$\frac{\lvert L\rvert}{\lvert\mathcal{V}\rvert}$	B
SemEval-Food	1486	1576	9	1184	0.80	5.08
SemEval-Verb	13936	13407	13	10360	0.74	4.12
MeSH	9710	10496	11	5502	0.57	3.88
Wikitax	941	973	7	754	0.80	5.20
CookBook	1985	1984	4	1795	0.90	10.44

Table 1: Statistics regarding the benchmark taxonomies. $|\mathcal{V}|$, $|\mathcal{E}|$, D, $|L|$, $\frac{|L|}{|\mathcal{V}|}$, B represent the node number, edge number, depth, the number of leaves, the ratio of leaves and the branching factor of the taxonomy.

Dataset	Train $\lvert\mathcal{V}\rvert$	Val $\lvert\mathcal{V}\rvert$	Test $\lvert\mathcal{V}\rvert$
SemEval-Food	1190 (80.0%)	148 (10.0%)	148 (10.0%)
SemEval-Verb	11996 (86.0%)	1000 (7.0%)	1000 (7.0%)
MeSH	8072 (83.0%)	819 (8.0%)	819 (8.0%)
Wikidata	753 (80.0%)	94 (10.0%)	94 (10.0%)
CookBook	1589 (80.0%)	198 (10.0%)	198 (10.0%)

Table 2: Node counts per split and dataset for the completion evaluation.

Betty Bossi, a subsidiary publishing company specialized in consumer recipes. We call this the *CookBook* taxonomy. Both taxonomies are available together with the source code.[5]

4.2 Evaluation

Due to our generative approach, we do not return a ranked list of candidate positions, making ranking metrics inappropriate for our case. Thus only precision (P), recall (R) and F1-scores (F1) of candidate positions (parent-query-child triplets) that were generated during inference are calculated. Following Liu et al. (2021), we additionally calculate the Wu & Palmer similarity (WPS) (Wu and Palmer, 1994). It measures the similarity between the paths in a taxonomy and is commonly known for its application as a similarity score with WordNet (Fellbaum, 2010). Let $p(c_t) = \langle c_r, ..., c_t \rangle$ be the path from the pseudo-root concept c_r to a target concept c_t. Likewise, let $\texttt{lca}(c_a, c_b)$ denote the depth of the least common ancestor of the nodes c_a and c_b. The WPS (Eq. 1) represents the similarity between concepts c_a and c_b where $p(c_a)$ and $p(c_b)$ are the paths from the root node to c_a and c_b. The score ranges $(0, 1]$, with 1 meaning that they share a parent.

$$WPS_{c_a c_b} = \frac{2 \cdot \texttt{lca}(c_a, c_b)}{|p(c_a)| + |p(c_b)|} \quad (1)$$

We follow Wang et al. (2022) in splitting the benchmark datasets into train (seed), validation and test taxonomies. We randomly exclude nodes (except root) and connect parents of excluded nodes with their children to keep the training (seed) taxonomy intact. An overview of the node counts per split can be found in Table 2.

In order to gain insights into performances across different node types, we provide total scores, as well as leaf and non-leaf scores. The leaf scores are a proxy for the performances on a taxonomy expansion task, where only leaves must be added.

Model selection Since running experiments on LLMs is expensive, and we want to make our approach easily accessible, we restrict our experiments to the open-source model Llama-3 (`Llama-3-70b-Instruct`).[6]

Hypothesis testing Following the recommendations of Dror et al. (2018), we use a two-sided paired randomization test ($\alpha = 0.05$) with 1k resamples to assess significant differences in model performance in the completion experiments. Since listing all p-values would require tables with hundreds of rows, we refrain from adding them here. However, they can be calculated using our published source code.

4.3 Results

Table 3 shows that LLM-based taxonomy completion can be competitive with state-of-the-art methods, even without tuning. The LLM approach is competitive with previous approaches on 3 of the 5 evaluated datasets. It is the best performing method on the CookBook taxonomy. However, it performs rather poorly on SemEval-Verb, the largest of the benchmark taxonomies – it is possible that fine-tuning becomes more advantageous as the size of the taxonomy increases. In all cases, few-shot prompting outperforms zero-shot, although the differences are not always statistically significant.

We further experimented with methods to automatically tune the prompt texts, but observed no significant difference to our manually optimized prompt. For details consult Appendix A.4.

Ablations In order to justify the usage of backtracking and NLI-verification, we evaluated ablated versions of the method on SemEval-Food (Table 4). Improvements are inconsistent overall, except for the non-leaf case, where the unablated model performs best for both zero-shot and few-shot. However, the scores are not significantly different according to randomization tests.

5 Generation Experiments

5.1 Data

To facilitate direct comparisons between true, generated and completed taxonomies, we extract all leaf-concepts from MeSH and SemEval-Food and try to regenerate a taxonomy only based on these known concepts.

5.2 Evaluation

Instead of only comparing our generated taxonomy to a gold standard, we acknowledge that there may be multiple valid taxonomies based on an single initial set of known concepts. Therefore, we additionally assess the taxonomies using reference-free metrics (Wullschleger

[5] `https://github.com/wullli/foodtaxo`

[6] `https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct`

Dataset	Model	Total				Non-Leaf				Leaf			
		WPS	F1	P	R	WPS	F1	P	R	WPS	F1	P	R
SemEval-Food	Arborist	0.7184	0.0828	0.1284	0.0611	0.7794	0.0199	0.0800	0.0114	0.7060	0.1318	0.1382	0.1259
	QEN	0.8900	0.2919	0.4527	0.2154	0.9042	0.0498	0.2000	0.0284	0.8871	0.4806	0.5041	0.4593
	TEMP	0.8945	0.3529	0.5473	0.2605	0.9155	0.0896	0.3600	0.0511	0.8902	0.5581	0.5854	0.5333
	TMN	0.8226	0.1089	0.1689	0.0804	0.8365	0.0299	0.1200	0.0170	0.8198	0.1705	0.1789	0.1630
	TacoPrompt	**0.9054**	**0.4052**	**0.6284**	**0.2990**	**0.9603**	**0.0995**	0.4000	**0.0568**	**0.8942**	**0.6434**	**0.6748**	**0.6148**
	TaxoExpan	0.8021	0.0566	0.0878	0.0418	0.8288	0.0100	0.0400	0.0057	0.7967	0.0930	0.0976	0.0889
	Llama-3 Few-Shot	0.8560	0.3025	0.5076	0.2154	0.8168	0.0914	**0.4286**	0.0511	0.8639	0.4715	0.5225	0.4296
	Llama-3 Zero-Shot	0.8164	0.2192	0.3780	0.1543	0.8005	0.0508	0.2381	0.0284	0.8196	0.3568	0.4057	0.3185
SemEval-Verb	Arborist	0.7430	0.0000	0.0000	0.0000	0.7359	0.0000	0.0000	0.0000	0.7437	0.0000	0.0000	0.0000
	QEN	**0.8321**	0.0967	0.1205	0.0808	**0.8624**	0.0056	0.0127	0.0036	**0.8292**	0.1167	0.1323	0.1044
	TEMP	0.8184	0.1431	0.1782	0.1195	0.8146	0.0224	0.0506	0.0144	0.8187	0.1695	0.1922	0.1516
	TMN	0.8036	0.0081	0.0100	0.0067	0.8276	0.0056	0.0127	0.0036	0.8012	0.0086	0.0097	0.0077
	TacoPrompt	0.8242	**0.1652**	**0.2058**	**0.1380**	0.8607	**0.0392**	**0.0886**	**0.0252**	0.8207	**0.1929**	**0.2187**	**0.1725**
	TaxoExpan	0.7896	0.0161	0.0201	0.0135	0.7756	0.0000	0.0000	0.0000	0.7910	0.0197	0.0223	0.0176
	Llama-3 Few-Shot	0.7879	0.0630	0.0814	0.0513	0.8332	0.0113	0.0263	0.0072	0.7835	0.0745	0.0877	0.0648
	Llama-3 Zero-Shot	0.7792	0.0608	0.0784	0.0497	0.8019	0.0113	0.0267	0.0072	0.7770	0.0718	0.0841	0.0626
MeSH	Arborist	0.5131	0.0000	0.0000	0.0000	0.5394	0.0000	0.0000	0.0000	0.5008	0.0000	0.0000	0.0000
	QEN	0.8609	0.1181	0.1978	0.0842	0.8815	0.0385	0.1077	0.0234	**0.8513**	0.2081	0.2397	0.1838
	TEMP	0.8311	0.1866	0.3126	0.1330	0.8686	0.0742	0.2077	0.0452	0.8137	0.3137	0.3614	0.2771
	TMN	0.5241	0.0000	0.0000	0.0000	0.5515	0.0000	0.0000	0.0000	0.5114	0.0000	0.0000	0.0000
	TacoPrompt	**0.8613**	**0.2201**	0.3687	**0.1569**	**0.9070**	0.0673	0.1885	0.0410	0.8401	**0.3929**	**0.4526**	**0.3471**
	TaxoExpan	0.5194	0.0020	0.0202	0.0010	0.5494	0.0000	0.0000	0.0000	0.5054	0.0051	0.0351	0.0027
	Llama-3 Few-Shot	0.8509	0.2139	**0.3750**	0.1496	0.8616	**0.1126**	**0.3333**	**0.0677**	0.8459	0.3301	0.3943	0.2840
	Llama-3 Zero-Shot	0.8481	0.1662	0.2877	0.1169	0.8563	0.0845	0.2460	0.0510	0.8444	0.2597	0.3071	0.2250
Wikidata	Arborist	0.7865	0.0556	0.0638	0.0492	0.7467	0.0000	0.0000	0.0000	0.7935	0.0741	0.0750	0.0732
	QEN	0.8663	0.1574	0.1809	0.1393	0.8143	0.0370	0.0714	0.0250	0.8754	0.1975	0.2000	0.1951
	TEMP	0.8513	0.2593	0.2979	0.2295	0.8710	0.1111	0.2143	0.0750	0.8479	0.3086	0.3125	0.3049
	TMN	0.7973	0.0926	0.1064	0.0820	0.7650	0.0370	0.0714	0.0250	0.8029	0.1111	0.1125	0.1098
	TacoPrompt	**0.8888**	0.2130	0.2447	0.1885	**0.8882**	0.1111	0.2143	0.0750	0.8889	0.2469	0.2500	0.2439
	TaxoExpan	0.7818	0.0185	0.0213	0.0164	0.8599	0.0000	0.0000	0.0000	0.7682	0.0247	0.0250	0.0244
	Llama-3 Few-Shot	0.8864	**0.2870**	**0.3298**	**0.2541**	0.8465	**0.1481**	**0.2857**	**0.1000**	**0.8934**	**0.3333**	**0.3375**	**0.3293**
	Llama-3 Zero-Shot	0.8744	0.2407	0.2766	0.2131	0.8166	0.1111	0.2143	0.0750	0.8845	0.2840	0.2875	0.2805
CookBook	Arborist	0.8536	0.0156	0.0202	0.0127	0.8743	0.0253	0.1000	0.0145	0.8513	0.0112	0.0112	0.0112
	QEN	0.9099	0.1868	0.2424	0.1519	0.9086	0.0253	0.1000	0.0145	0.9101	0.2584	0.2584	0.2584
	TEMP	0.9206	0.2529	0.3283	0.2057	0.9452	0.0506	0.2000	0.0290	0.9179	0.3427	0.3427	0.3427
	TMN	0.8495	0.0623	0.0808	0.0506	0.8990	0.0253	0.1000	0.0145	0.8439	0.0787	0.0787	0.0787
	TacoPrompt	0.9243	0.2879	0.3737	0.2342	0.9300	0.0506	0.2000	0.0290	0.9236	0.3933	0.3933	0.3933
	TaxoExpan	0.8234	0.0272	0.0354	0.0222	0.7713	0.0127	0.0500	0.0072	0.8293	0.0337	0.0337	0.0337
	Llama-3 Few-Shot	**0.9342**	**0.3327**	**0.4359**	**0.2690**	**0.9629**	**0.0633**	**0.2500**	**0.0362**	**0.9310**	**0.4533**	**0.4571**	**0.4494**
	Llama-3 Zero-Shot	0.9089	0.2383	0.3112	0.1930	0.9343	0.0380	0.1500	0.0217	0.9060	0.3277	0.3295	0.3258

Table 3: Scores of the completion evaluation on all datasets. All scores that are not significantly different to the best model according to a two-sided paired randomization test ($\alpha = 0.05$) with 1k resamples are underlined. Note that due to the rarity of non-leaves, these results rarely show significant differences.

Setting	Model	Total				Non-Leaf				Leaf			
		WPS	F1	P	R	WPS	F1	P	R	WPS	F1	P	R
Zero-Shot	w/o Backtracking	0.7970	**0.2454**	**0.4380**	**0.1704**	0.7654	0.0306	0.1500	0.0170	0.8034	**0.4237**	**0.4950**	**0.3704**
	w/o NLI Validation	**0.8206**	0.2257	0.3788	0.1608	0.7567	0.0406	0.1905	0.0227	**0.8336**	0.3740	0.4144	0.3407
	Complete	0.8175	0.2192	0.3780	0.1543	**0.8027**	**0.0508**	**0.2381**	**0.0284**	0.8205	0.3568	0.4057	0.3185
Few-Shot	w/o Backtracking	0.8052	0.2593	0.4628	0.1801	0.7616	0.0622	0.3529	0.0341	0.8140	0.4184	0.4808	0.3704
	w/o NLI Validation	0.8581	0.2793	0.4662	0.1994	0.8175	0.0711	0.3333	0.0398	**0.8664**	0.4453	0.4911	0.4074
	Complete	**0.8583**	**0.3025**	**0.5076**	**0.2154**	**0.8282**	**0.0914**	**0.4286**	**0.0511**	0.8645	**0.4715**	**0.5225**	**0.4296**

Table 4: Ablation study of NLI-verification and Backtracking on the completion task for SemEval-Food. All scores that are not significantly different to the best model according to a two-sided paired randomization test ($\alpha = 0.05$) with 1k resamples are underlined.

et al., 2025). In particular we evaluate concept similarity correlation (CSC) and NLI-verification (NLIV), and compare scores between the generated and benchmark taxonomies.

CSC measures taxonomy robustness by correlating the taxonomic similarity of concepts (using WPS) with their semantic similarities according to an embedding model. Robustness indicates how well a taxonomy can tell things apart, meaning how clearly the concepts in a taxonomy represent different ideas (orthogonality) and how closely related sibling concepts are (cohesiveness).

NLIV evaluates logical adequacy by checking the validity of relations in a taxonomy. More specifically, if the process of classification is a walk on a taxonomy graph (from root node to classified node), then NLIV estimates classification probabilities with NLI and normalizes them by walk length. For example, in a food taxonomy, given the relation (*antipasto*, *appetizer*), the premise *"antipasto is a course of appetizers in an Italian meal"* and hypothesis *"antipasto is a kind of appetizer"* are passed to an NLI-model. NLIV has two versions: weak (NLIV-W), where the premise must not contradict the hypothesis, and strong (NLIV-S), where the premise must entail it. Note that due to our model-internal NLI-verification (see Section 3.5), results might be biased towards our model. However, we use two unrelated NLI-models for evaluation and completion to improve fairness (see Appendix A.3).

Dataset	Taxonomy	vs. Gold Standard		Reference-free		
		Position-F1	Parent-F1	NLIV-W	NLIV-S	CSC
SemEval-Food	TacoPrompt	0.6432	**0.7249**	0.3479	0.0451	-0.0023
	True	-	-	0.9641	**0.2017**	0.0426
	Completed	**0.6435**	0.7159	0.9525	0.1774	0.0097
	Generated	0.0234	0.0390	**0.9726**	0.1298	**0.0777**
MeSH	TacoPrompt	**0.6584**	**0.7397**	0.5638	0.0510	0.0050
	True	-	-	**0.8502**	**0.1680**	0.0614
	Completed	0.6368	0.7275	0.8412	0.1560	0.0518
	Generated	0.0094	0.0175	0.8167	0.1237	**0.1051**

Table 5: Comparison of metrics for the true taxonomy, completed taxonomy (Ours and TacoPrompt) and a taxonomy constructed by our generation method.

Configuration	CSC	NLIV-S	NLIV-W
w/o NLI-Verification	**0.0785**	0.1126	0.9630
w/o Taxonomy Description	0.0386	0.1140	0.9607
w/o Generation	0.0445	**0.1519**	0.9717
w/o Backtracking	0.0328	0.1091	0.9683
Complete	0.0703	0.1298	**0.9726**

Table 7: Ablation study highlighting the effects of NLI validation and taxonomy description on the generation metrics. The study was done by constructing a taxonomy using all leaf concepts from SemEval-Food.

Gold-Standard Comparison For reference, we also calculate F1-scores on the complete gold standard taxonomy, which indicate how much of the gold standard was recovered during generation. The Position-F1 indicates how many triplets were matched, while the Parent-F1 indicates how often the correct parent, but not child, was predicted.

5.3 Results

Table 5 shows a comparison of our generation method against the gold standard, TacoPrompt and our completion method on SemEval-Food and MeSH. We can see that our reference-free scores are competitive with the gold standard and according to CSC even better on both datasets. However, the CSC score does not respect that there might be invalid relationships in the taxonomy (not of type `is-a`) and we find by qualitative inspection that NLIV better represents the actual quality of the taxonomy. Further, we notice that there are frequent erroneous classifications (example Fig. 2c), which are not well captured by the metrics. Such issues likely stem from poor model performance on non-leaves (Table 3). Table 6 shows statistics regarding the generated taxonomies.

Ablations In order to test the effectiveness of our modeling choices, we conducted an ablation study by removing different mechanisms from our algorithm. In Table 7 we present the results for models without NLI-verification, taxonomy description, backtracking, and generation. Without generation, only existing concepts can be used to build the taxonomy. In the configuration without a taxonomy description, we remove the initial step, where we let an LLM imagine a potential taxonomy.

All of our mechanisms result in an improvement of either CSC or NLIV. We observe the best CSC score for the model without NLI-verification, but when qualitatively exploring the taxonomy generated by this model, we observe frequent cases where an edge does not represent an *is-a* relation, which is better reflected in the NLIV score.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	D	$\lvert$L$\rvert$	$\frac{\lvert L\rvert}{\lvert\mathcal{V}\rvert}$	B
MeSH	6908	6858	10	5712	0.83	5.65
SemEval-Food	1213	1257	11	1130	0.93	15.14
SemEval-Food (w/o NLI)	1203	1216	6	1122	0.93	15.01
SemEval-Food (w/o Backtracking)	1228	1272	7	1108	0.90	10.60
SemEval-Food (w/o Generation)	1233	1251	12	1135	0.92	12.77

Table 6: Statistics regarding generated taxonomies. $|\mathcal{V}|$, $|\mathcal{E}|$, D, $|L|$, $\frac{|L|}{|\mathcal{V}|}$, B represent the node number, edge number, depth, the number of leaves, the ratio of leaves and the branching factor of the taxonomy.

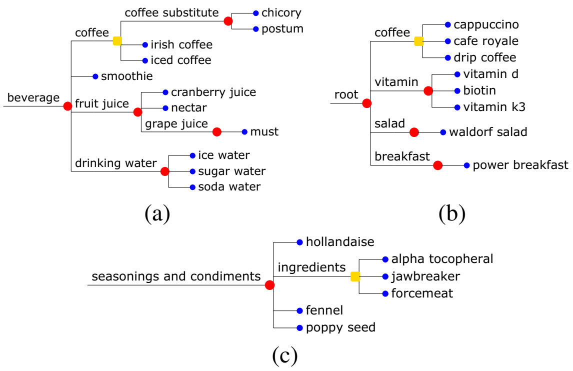

Figure 2: Examples of generated sub-graphs of the taxonomies. Depiction (a) is the gold standard neighborhood of *coffee*, while (b) is an example of the generated taxonomy based on SemEval-Food leaves. An erroneously classified non-leaf is shown in (c).

6 Conclusion

We introduce an algorithm for the generation of taxonomies given a set of known concepts using LLMs, thereby enabling us to scale taxonomies to dataset sizes beyond what can be managed by human curators with sensible efforts. We benchmark our LLM-based approach against state-of-the-art taxonomy completion methods, demonstrating its potential. Despite the fact that our research endeavor stems from the food technology industry, the presented methods for taxonomy generation and completion are general and agnostic to the concrete use-case or industry. Some of our experiments therefore involve linguistic and healthcare taxonomies.

The taxonomies generated by our method achieve promising scores across existing quality metrics. However, qualitative inspection reveals that they still fall short of the nuance seen in human-curated taxonomies. We conclude that for LLM-based taxonomy generation to reach practical utility, significant advances are still needed, particularly in the reliable placement of non-leaf concepts.

7 Limitations

- Due to the computational overhead associated with LLMs, our experiments are only carried out using one open-source LLM. Care should be taken when interpreting results based on one LLM alone.
- Our current approach does not generate taxonomies with respect to a target application, which is important in practical scenarios.
- While reference-free metrics hint at taxonomy quality, they are likely non-exhaustive and always need to be assessed in combination, since they measure different properties of taxonomy quality.

8 Acknowledgements

We would like to express our sincere appreciation to Betty Bossi[7] for their support of this research project and for providing us with their taxonomy used for recipe development. This research is supported through computing resources by the ADAPT Centre for Digital Content Technology, which is funded under the SFI Research Centres Programme (Grant 13/RC/2106_P2) and is co-funded under the European Regional Development. The authors thank the reviewers for their insightful and helpful comments.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, California. Association for Computational Linguistics.

Boqi Chen, Fandi Yi, and Dániel Varró. 2023. Prompting or fine-tuning? a comparative study of large language models for taxonomy construction. In *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 588–596.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker‘s guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Nicolas Rodolfo Fauceglia, Alfio Gliozzo, Sarthak Dash, Md. Faisal Mahbub Chowdhury, and Nandana Mihindukulasooriya. 2019. Automatic taxonomy induction and expansion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 25–30, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 2010. *WordNet*, pages 231–243. Springer Netherlands, Dordrecht.

David Jurgens and Mohammad Taher Pilehvar. 2016. SemEval-2016 task 14: Semantic taxonomy enrichment. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1092–1102, San Diego, California. Association for Computational Linguistics.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023a. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *Preprint*, arXiv:2212.14024.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023b. Dspy: Compiling declarative language model calls into self-improving pipelines. *Preprint*, arXiv:2310.03714.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

C E Lipscomb. 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–266.

Zichen Liu, Hongyuan Xu, Yanlong Wen, Ning Jiang, HaiYing Wu, and Xiaojie Yuan. 2021. TEMP: Taxonomy expansion with dynamic margin loss through taxonomy-paths. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3854–3863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingyu Derek Ma, Muhao Chen, Te-Lin Wu, and Nanyun Peng. 2021. HyperExpan: Taxonomy expansion with hyperbolic representation learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4182–4194, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*, WWW '20, page 2044–2054, New York, NY, USA. Association for Computing Machinery.

[7] https://www.bettybossi.ch/

Daniele Margiotta, Danilo Croce, and Roberto Basili. 2023. Taxosbert: Unsupervised taxonomy expansion through expressive semantic similarity. In *Deep Learning Theory and Applications*, pages 295–307, Cham. Springer Nature Switzerland.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, Singapore. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*, WWW '20, page 486–497, New York, NY, USA. Association for Computing Machinery.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2180–2189, New York, NY, USA. Association for Computing Machinery.

Arnav Singhvi, Manish Shetty, Shangyin Tan, Christopher Potts, Koushik Sen, Matei Zaharia, and Omar Khattab. 2024. Dspy assertions: Computational constraints for self-refining language model pipelines. *Preprint*, arXiv:2312.13382.

Suyuchen Wang, Ruihui Zhao, Yefeng Zheng, and Bang Liu. 2022. Qen: Applicable taxonomy completion via evaluating full taxonomic relations. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1008–1017, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Pascal Wullschleger, Majid Zarharan, Donnacha Daly, Marc Pouly, and Jennifer Foster. 2025. No gold standard, no problem: Reference-free evaluation of taxonomies. *Preprint*, arXiv:2505.11470.

Hongyuan Xu, Ciyi Liu, Yuhang Niu, Yunong Chen, Xiangrui Cai, Yanlong Wen, and Xiaojie Yuan. 2023. TacoPrompt: A collaborative multi-task prompt learning method for self-supervised taxonomy completion. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15804–15817, Singapore. Association for Computational Linguistics.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1026–1035, New York, NY, USA. Association for Computing Machinery.

Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2104–2113, New York, NY, USA. Association for Computing Machinery.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Jieyu Zhang, Xiangchen Song, Ying Zeng, Jiaze Chen, Jiaming Shen, Yuning Mao, and Lei Li. 2021. Taxonomy completion via triplet matching network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4662–4670.

A Implementation Details

A.1 Algorithms

The proposed methods for completion and generation are formulated in more detail than in the main section in algorithms 1 and 2 respectively.

A.2 Embeddings

For the retrieval step in our proposed models, we used FastText (Bojanowski et al., 2017). In order to avoid a biased evaluation, we instead used sentence transformer embeddings[8] (Reimers and Gurevych, 2019) for CSC to measure semantic similarity.

A.3 NLI Verification

To minimise bias between model inference and evaluation, we use two different models. For the verification of generated concepts in the inference, we used *ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli*[9], and for the NLIV metric during the evaluation *facebook/bart-large-mnli*[10].

A.4 DSPy

The DSPy library (Khattab et al., 2023b), enables us to use RAG in an off-the-shelf manner and to tune prompts per model and datasets with a hyper-parameter-tuning like approach.

Due to issues we encountered with Llama3 and the DSPy library, we customized the template DSPy uses to generate prompts, by more clearly separating the input and output fields[11]. The customization is apparent in our examples of prompts and outputs, as shown in Section 3.4.

DSPy provides optimizers which can be used to tune prompts given validation and training data. We evaluated the automated tuning of instruction texts with their COPRO optimizer. This optimizer generates variations of a predefined prompt using a language model and evaluates its effectiveness on validation examples. It keeps the most promising examples and generates further variations them. Results of the comparison of instruction-tuned (COPRO) against our handcrafted prompts can be found in Table 8. We randomly sampled 20 concepts from our validation and training sets respectively and ran the optimizer by generating 5 initial variations of our default prompt and allowing 2 subsequent variations on each.

A.5 Processing LLM Outputs

It is possible, that an LLM predicts a set of parents for a concept, where inside that set one parent is already an ancestor of another in the taxonomy. In such a case, we select the most specific concept (furthest from the root).

A.6 Concept Descriptions

The MeSH, SemEval-Food and SemEval-Verb datasets include descriptions for all concepts. For Wikidata and CookBook we have no concept descriptions and instead generated descriptions using `gpt-4o-mini`.

A.7 Evaluation metrics

We notice that some test concepts in SemEval-Verb do not have gold standard positions. We do not calculate any scores for such concepts but average over the available gold standards. Note that, since we follow Zhang et al. (2021) and assume that the task is N independent attachment problems, it is possible that we create cycles by inserting all predicted placements into an existing taxonomy. The calculation of quality attributes, such as robustness, requires the insertion of concepts to calculate scores. In such cases, we simply drop placements that would lead to cycles and do not consider them during the calculation. The standard metrics used in completion are described below. Note that for a position to be considered correct, both parent and child of the query concept need to be correct. A correctly predicted parent with an incorrectly predicted child will result in a false positive and vice versa.

Recall (R) How many of the true positions were correctly predicted by the model.

$$\frac{TP}{TP + FN} \tag{2}$$

Precision (P) How many of the predicted positions were correct.

$$\frac{TP}{TP + FP} \tag{3}$$

F1-score (F1) The harmonic mean of the precision and recall for the positions.

$$2 \cdot \frac{P \cdot R}{P + R} \tag{4}$$

B Experiment Details

We reused implementations for the baselines from Xu et al. (2023) and adjusted them for our setting by adding the functionality to output the best placements (triplets) for a query instead of a ranked list, so that we could subsequently calculate F1, precision, and recall. We ensured the quality of the implementation of our metrics by validating them against metrics used by Xu et al. (2023).

B.1 Baselines

We utilized the following state-of-the-art taxonomy completion techniques as baselines for comparison with our proposed method.

[8] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

[9] `https://huggingface.co/ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli`

[10] `https://huggingface.co/facebook/bart-large-mnli`

[11] `https://github.com/wullli/foodtaxo`

Algorithm 1 Taxonomy Completion

```
Require: A query concept $q \in \mathcal{Q}$ to insert into taxonomy $\mathcal{T} = (\mathcal{E}, \mathcal{V})$ and a description $d_q$ for the query concept $q \in \mathcal{Q}$
Ensure: A set of predicted placements $\mathcal{Y}_q$ for the query concept $q$
$\mathcal{Y}_q \leftarrow \emptyset$    ▷ Set of predicted placements for the query $q$
$R \leftarrow \text{Retrieve}(q, \mathcal{T}, k)$    ▷ Retrieve $k$ most relevant edges $R$ by cosine similarity to $q$
$\mathcal{P} \leftarrow \text{CoT}_p(q, R, d_q)$    ▷ Generate candidate parent concepts using CoT prompting
$\mathcal{P} \leftarrow \{p \in \mathcal{P} \mid \neg\text{contradicts}(\lceil_q, \text{"lemma}(q) \text{ is a lemma}(p)\text{"})\}$    ▷ Validate parents with NLI
$\mathcal{C} \leftarrow \{c \in \mathcal{V} \mid c \text{ is a child of any } p \in \mathcal{P}\}$    ▷ Get candidate children
$\mathcal{C} \leftarrow \text{CoT}_c(q, \mathcal{C}, R, d_q)$    ▷ Select valid children using CoT prompting
$\mathcal{C} \leftarrow \{c \in \mathcal{C} \mid \text{entails}(\lceil_q, \text{"lemma}(c) \text{ is a lemma}(q)\text{"})\}$    ▷ Validate children with NLI
for each parent-child combination $(p, c) \in \mathcal{P} \times \mathcal{C}$ do
    if $p$ is a parent of $c$ in $\mathcal{T}$ then
        $\mathcal{Y}_q \leftarrow \mathcal{Y}_q \cup \{(p, q, c)\}$    ▷ Add valid placement to $\mathcal{Y}_q$
    end if
end for
```

Algorithm 2 Taxonomy Generation

```
Require: A set of concepts $\mathcal{Q}$ to insert into taxonomy $\mathcal{T} = (\mathcal{E}, \mathcal{V})$ and a description $d_q \in \mathcal{D}$ for each query concept $q \in \mathcal{Q}$
Ensure: A completed taxonomy $\mathcal{T}$
$\mathcal{V} \leftarrow \mathcal{Q}$
$\mathcal{E} \leftarrow \emptyset$
$\mathcal{Q}_n \leftarrow \{q_1, \dots, q_n\},\ q_i \overset{iid}{\sim} \text{Uniform}(\mathcal{Q})$    ▷ Sample $n$ concepts from $\mathcal{Q}$
$d_t \leftarrow \text{CoT}_d(\mathcal{Q}_n)$    ▷ Describe the potential taxonomy using CoT prompting
while $|\mathcal{Q}| > 0$ do
    $q \leftarrow \text{Next}(\mathcal{Q})$    ▷ Get next query $q$ from set of concepts to add
    $R \leftarrow \text{Retrieve}(q, \mathcal{T}, k)$    ▷ Retrieve $k$ most relevant edges $R$ by cosine similarity to $q$
    $\mathcal{P} \leftarrow \text{CoT}_p(q, R, d_q, d_t)$    ▷ Generate candidate parent concepts using CoT prompting
    $\mathcal{P} \leftarrow \{p \in \mathcal{P}_q \mid \neg\text{contradicts}(d_q, \text{"lemma}(q) \text{ is a lemma}(p)\text{"})\}$    ▷ Validate parents with NLI
    $\mathcal{C} \leftarrow \{c \in \mathcal{V} \mid c \text{ is a child of any } p \in \mathcal{P}\}$    ▷ Get candidate children
    $\mathcal{C} \leftarrow \text{CoT}_c(q, \mathcal{C}, R, d_q, d_t)$    ▷ Select valid children using CoT prompting
    $\mathcal{C} \leftarrow \{c \in \mathcal{C} \mid \text{entails}(d_q, \text{"lemma}(c) \text{ is a lemma}(q)\text{"})\}$    ▷ Validate children with NLI
    $\mathcal{N} \leftarrow \mathcal{P} \setminus \mathcal{V}$    ▷ Get newly generated concepts
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{N}$    ▷ Update set of concepts to add
    $\mathcal{T} \leftarrow \text{InsertParents}(q, \mathcal{P}, \mathcal{T})$    ▷ Insert new parent-query edges into taxonomy.
    $\mathcal{T} \leftarrow \text{InsertChildren}(q, \mathcal{C}, \mathcal{T})$    ▷ Insert new query-child edges into taxonomy.
    $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{q\}$    ▷ Remove added concept
end while
```

Dataset	Model	Total				Non-Leaf				Leaf			
		WPS	F1	P	R	WPS	F1	P	R	WPS	F1	P	R
SemEval-Food	Llama-3 Zero-Shot	**0.8177**	0.2192	0.3780	0.1543	**0.8050**	0.0508	0.2381	0.0284	**0.8203**	0.3568	0.4057	0.3185
	Llama-3 Zero-Shot*	0.7723	**0.2367**	**0.4250**	**0.1640**	0.7407	**0.0619**	**0.3333**	**0.0341**	0.7788	**0.3797**	**0.4412**	**0.3333**
MeSH	Llama-3 Zero-Shot	**0.8549**	**0.1662**	0.2877	**0.1169**	**0.8645**	**0.0845**	0.2460	**0.0510**	**0.8504**	**0.2597**	**0.3071**	**0.2250**
	Llama-3 Zero-Shot*	0.8397	0.1610	**0.2882**	0.1117	0.8473	0.0824	**0.2500**	0.0493	0.8361	0.2518	0.3059	0.2140

Table 8: Comparison of instruction tuning using DSPy optimizers. All scores that are not significantly different to the best model according to a two-sided paired randomization test ($\alpha = 0.05$) with 1k resamples are underlined. Models marked with an asterisk (*) were instruction tuned using DSPy.

- **Arborist**: Manzoor et al. (2020) propose Arborist, an approach to expand textual taxonomies by predicting parents of new nodes with unobserved heterogeneous edge semantics. Arborist learns latent edge representations and node embeddings, optimizing a large-margin ranking loss to minimize the shortest-path distance between predicted and actual parents.

- **QEN**: Wang et al. (2022) propose the Quadruple Evaluation Network (QEN), a taxonomy completion framework using term descriptions, pretrained language models, and code attention for accurate inference while reducing computation. QEN evaluates parent-child and sibling relations to enhance accuracy and reduce noise from pseudo-leaves.

- **TEMP**: Liu et al. (2021) present TEMP, a self-supervised taxonomy expansion method that predicts new concept positions by ranking generated paths. TEMP utilizes pre-trained contextual encoders for taxonomy construction and hypernym detection. Liu et al. (2021) show that pretrained contextual embeddings capture hypernym-hyponym relations effectively.

- **TMN**: Zhang et al. (2021) introduce "taxonomy completion" and propose the Triplet Matching Network (TMN) to find hypernym and hyponym concepts for a query. TMN, featuring a primal scorer, auxiliary scorers, and a channel-wise gating mechanism, outperforms existing methods.

- **TacoPrompt**: Xu et al. (2023) introduce TacoPrompt, employing triplet semantic matching via prompt learning to address imbalanced data, a contextual approach to connect subtask results with final predictions. TacoPrompt also leverages a two-stage retrieval and re-ranking method to enhance inference efficiency.

- **TaxoExpan**: Shen et al. (2020) present TaxoExpan, a self-supervised framework for expanding taxonomies by automatically generating ⟨query concept, anchor concept⟩ pairs from existing taxonomies. TaxoExpan uses this data to predict whether a query concept is the direct hyponym of an anchor concept.

B.2 Prompt

We show the default handcrafted prompts for generating parent (Listing 1) and child concepts (Listing 2) of a query. We optimized these prompts by manual trial and error on the validation data. For spacing reasons, we display only two lines of context for both prompts and remove double new-lines. Note that the prompts for generation are slightly different.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	**D**	$\lvert\mathbf{L}\rvert$	$\frac{\lvert\mathbf{L}\rvert}{\lvert\mathcal{V}\rvert}$	**B**
SemEval-Food	1486	1576	9	1184	0.80	5.08
SemEval-Verb	13936	13407	13	10360	0.74	4.12
MeSH	9710	10496	11	5502	0.57	3.88
Wikitax	941	973	7	754	0.80	5.20
CookBook	1985	1984	4	1795	0.90	10.44
Generated Recipe1M	12376	12745	15	10156	0.82	5.74
Generated MeSH	6908	6858	10	5712	0.83	5.65
Generated SemEval-Food	1213	1257	11	1130	0.93	15.14
Generated SemEval-Food (w/o NLI)	1203	1216	6	1122	0.93	15.01
Generated SemEval-Food (w/o Backtracking)	1228	1272	7	1108	0.90	10.60
Generated SemEval-Food (w/o Generation)	1233	1251	12	1135	0.92	12.77

Table 9: Statistics regarding generated and brenchmark taxonomies. $|\mathcal{V}|$, $|\mathcal{E}|$, D, $|L|$, $\frac{|L|}{|\mathcal{V}|}$, B represent the node number, edge number, depth, the number of leaves, the ratio of leaves and the branching factor of the taxonomy.

In this section we provide examples of model prompts for DSPy-tuned and untuned models. The same example is provided for each setting. The context is only displayed for the first Parent and Child generation prompt to limit verbosity.

B.2.1 Completion: Tuned Zero-Shot Parents

```
// Prompt
Which are the most specific parent concepts of the
 given child concept in a taxonomy considering the
 context?
In your reasoning, state how the parent concepts are
  a supertype of the child concept.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Child: Child concept (subtype) that you need to
 place in a taxonomy.

Description: Description of the child concept.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the parents}. We ...

Interpretation: Description of the child concept in
 relation to the context taxonomy. Infer what is
 meant by the child concept from the context.

Parents: Comma separated list of one or more parents
  of the child concept. Valid parents are in the
 context. If there are no suitable parents, return
 None.

---

Context:
```
...
```

Child: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Reasoning: Let's think step by step in order to
```

```
// Output
Reasoning: Let's think step by step in order to find
 the parents of the child concept "sweetening". We
can infer that sweetening is related to making food
 taste sweeter, which is closely related to sugar
and other sweet substances.
```

````
Interpretation: The child concept ”sweetening”
 refers to an additive that enhances the sweetness
 of food , which is a common theme in the context
 taxonomy.

Parents: sugar , syrup
````

B.2.2 Completion: Tuned Zero-Shot Children

~~~
// Prompt
Which of the candidates are child concepts (subtypes
 ) of the given parent concept (supertype) in a
 taxonomy?
The context shows existing parents and children
 concepts and whether the children are leaves.
In your reasoning , state how the parent concept is a
  supertype of the selected child concepts.
Do not add additional comments or information , only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Candidates: Candidate children of the concept
 separated by commas to select from.

Parent: Parent concept that you need to place in a
 taxonomy.

Description: Description of the parent concept.

Interpretation: Description of the child concept in
 relation to the taxonomy.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the children}. We ...

Leaf: Whether the parent concept should be added as
 a leaf (has no children). Answer with Yes or No.

Children: Comma separated list of candidates that
 are children of the parent concept in a taxonomy.A
 child concept must be a type of the parent concept.
 Separate with commas.

---

Context:
```
...
```

Candidates: corn syrup , molasses , maple syrup ,
 grenadine , sorghum , sugar syrup , corn sugar , lump
 sugar , caramel , granulated sugar , sugarloaf , beet
 sugar , brown sugar

Parent: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Interpretation: The child concept ”sweetening”
 refers to an additive that enhances the sweetness
 of food , which is a common theme in the context
 taxonomy.

Reasoning: Let's think step by step in order to
~~~

````
// Output
Reasoning: Let's think step by step in order to
 identify the children of the concept ”sweetening”.
 We can see that the context taxonomy includes
 various types of sweet substances and additives ,
 which are used to make food taste sweeter. Based on
  this , we can identify the child concepts that are
 types of sweetening agents.

Leaf: No

Children: sugar syrup , corn syrup , molasses , maple
 syrup , grenadine , sorghum , caramel , granulated
 sugar , sugarloaf , beet sugar , brown sugar
````

B.2.3 Completion: Few-Shot Parents

In this example, we can see that backtracking (retrying with instructions) was applied, since the initial answer failed to fulfill the required constraints. Remarks prefixed with double slashes (/) are not actually part of the raw prompt, but serve as comments in the illustrations that follow.

~~~
// Prompt
Which are the most specific parent concepts of the
 given child concept in a taxonomy considering the
 context?
In your reasoning , state how the parent concepts are
  a supertype of the child concept.
Do not add additional comments or information , only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Child: Child concept (subtype) that you need to
 place in a taxonomy.

Description: Description of the child concept.

Previous Reasoning: past Reasoning: with errors

Previous Interpretation: past Interpretation: with
 errors

Previous Parents: past Parents: with errors

Instructions: Some instructions you must satisfy

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the parents}. We ...

Interpretation: Description of the child concept in
 relation to the context taxonomy. Infer what is
 meant by the child concept from the context.

Parents: Comma separated list of one or more parents
  of the child concept. Valid parents are in the
 context. If there are no suitable parents , return
 None.

---

Context:
```tetrazzini , chicken tetrazzini
dish , chicken provencale
dish , chicken marengo
casserole , chicken casserole
broth , chicken broth```

Child: chicken paprika

Description: chicken paprika is chicken simmered in
 broth with onions and paprika then mixed with sour
 cream

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of ”chicken paprika”. We can see
 that ”chicken paprika” is a type of dish , as it is
 a preparation of chicken with certain ingredients.
 Therefore , it can be classified under ”dish”.

Interpretation: ”chicken paprika” is a type of dish
 that involves simmering chicken in broth with
 onions and paprika , and then mixing it with sour
 cream.

Parents: dish
~~~

~~~
---

Context:
```powdered sugar, icing sugar
flavorer, sugar
syrup, sugar syrup
sugar, brown sugar
sugar, beet sugar```

Child: granulated sugar

Description: granulated sugar is sugar in the form
 of small grains

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of granulated sugar. We can see
 that granulated sugar is a type of sugar, so sugar
 is a supertype of granulated sugar.

Interpretation: Granulated sugar is a type of sugar
 that is in the form of small grains, which is a
 specific characteristic of this type of sugar.

Parents: sugar

---

Context:
```wine, blush wine
wine, canary wine
beverage, wine
vinegar, wine vinegar
sauce, wine sauce```

Child: red wine

Description: red wine is wine having a red color
 derived from skins of dark-colored grapes

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of "red wine". We know that "red
 wine" is a type of "wine" because it is described
 as "wine having a red color derived from skins of
 dark-colored grapes".

Interpretation: "Red wine" is a specific type of
 wine, which is a beverage.

Parents: wine, beverage

---

Context:
```tea, ice tea
tea, green tea
tea, herb tea
herb, tea
tisane, camomile tea```

Child: sun tea

Description: sun tea is tea made by exposing tea
 leaves steeped in water to the direct rays of the
 sun; usually served with ice

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of sun tea. We can see that sun
 tea is a type of tea, as it is made from tea leaves
  steeped in water.

Interpretation: Sun tea is a type of tea that is
 made using a specific method, which is by exposing
 the tea leaves to the direct rays of the sun.

Parents: tea

---

Context:
~~~
...
```

Child: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Previous Reasoning: Reasoning: Let's think step by
 step in order to find the parents of "sweetening".
 We can see that "sweetening" is a type of "
 foodstuff" or an ingredient added to food, but
 there is no direct match in the context. However,
 we can relate it to "sweet" which is a type of food
  characteristic.

Previous Interpretation: "Sweetening" is an
 ingredient or substance added to food to make it
 sweeter, which is a characteristic of some foods.

Previous Parents: None

Instructions: None is not a valid parent.

Reasoning: Let's think step by step in order to
```

```
// Output
Reasoning: Let's think step by step in order to find
  the parents of "sweetening". We can see that "
 sweetening" is something added to foods to make
 them taste sweeter, which is related to the concept
  of "sweet" as a characteristic of food. Although
 there is no direct match in the context, we can
 relate it to "foodstuff" as it is an ingredient
 added to food.

Interpretation: "Sweetening" is an ingredient or
 substance added to food to make it sweeter, which
 is a characteristic of some foods.

Parents: foodstuff
```

B.2.4 Completion: Few-Shot Children

~~~
// Prompt
Which of the candidates are child concepts (subtypes
 ) of the given parent concept (supertype) in a
 taxonomy?
The context shows existing parents and children
 concepts and whether the children are leaves.
In your reasoning, state how the parent concept is a
 supertype of the selected child concepts.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Candidates: Candidate children of the concept
 separated by commas to select from.

Parent: Parent concept that you need to place in a
 taxonomy.

Description: Description of the parent concept.

Interpretation: Description of the child concept in
 relation to the taxonomy.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the children}. We ...

Leaf: Whether the parent concept should be added as
 a leaf (has no children). Answer with Yes or No.

Children: Comma separated list of candidates that
 are children of the parent concept in a taxonomy.A
 child concept must be a type of the parent concept.
 Separate with commas.

---

Context:
```tetrazzini (Non-Leaf), chicken tetrazzini (Leaf)
dish (Non-Leaf), chicken provencale (Leaf)
~~~

~~~
dish (Non-Leaf), chicken marengo (Leaf)
casserole (Non-Leaf), chicken casserole (Leaf)
broth (Non-Leaf), chicken broth (Leaf)```

Candidates: chicken cordon bleu, croquette, pudding,
 pasta, succotash, chow mein, cottage pie,
 spaghetti and meatballs, poi, jambalaya, roulade,
 swiss steak, tamale pie, bacon and eggs, enchilada,
  barbecue, meat loaf, patty, lobster thermidor,
 potpie, coquilles saint jacques, sauerbraten, coq
 au vin, sauerkraut, tetrazzini, moussaka, refried
 beans, fondue, dolmas, steak au poivre, viand,
 sukiyaki, timbale, porridge, scallopine, seafood
 newburg, lutefisk, frittata, omelet, soup, pepper
 steak, spanish rice, galantine, barbecued wing,
 salisbury steak, sashimi, couscous, fish and chips,
  piece de resistance, veal cordon bleu, hash, chop
 suey, scotch egg, scrambled eggs, poached egg,
 chicken marengo, casserole, terrine, macedoine,
 pizza, meatball, welsh rarebit, osso buco, kishke,
 chicken paprika, carbonnade flamande, shirred egg,
 scampi, mold, taco, pork and beans, bitok, french
 toast, burrito, scrapple, haggis, pheasant under
 glass, maryland chicken, beef bourguignonne, boiled
  dinner, rijsttaffel, chicken and rice, schnitzel,
 kabob, beef wellington, risotto, paella, tempura,
 special, souffle, mousse, fish stick, tostada, frog
  legs, chili, snack food, ramekin, ham and eggs,
 boiled egg, chicken provencale, rissole, pilaf,
 applesauce, moo goo gai pan, kedgeree, stew,
 coleslaw, tossed salad, herring salad, crab louis,
 tuna fish salad, molded salad, pasta salad, potato
 salad, salmagundi, tabbouleh, chicken salad, fruit
 salad, lamb curry, raita, fruit custard, bavarian
 cream, creme caramel, knish, pirogi, apple turnover
 , samosa

Parent: chicken paprika

Description: chicken paprika is chicken simmered in
 broth with onions and paprika then mixed with sour
 cream

Interpretation: "chicken paprika" is a type of dish
 that involves simmering chicken in broth with
 onions and paprika, and then mixing it with sour
 cream.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 identify the child concepts of "chicken paprika".
 We can see that "chicken paprika" is a type of dish
 , and it has a specific preparation method and
 ingredients. Therefore, it is a subtype of "dish"
 and cannot have children from the given candidates.

Leaf: Yes

Children: Reasoning: Let's think step by step in
 order to identify the child concepts of "chicken
 paprika". We can see that "chicken paprika" is a
 type of dish, and it has a specific preparation
 method and ingredients. Therefore, it is a subtype
 of "dish" and cannot have children from the given
 candidates. Leaf: Yes Children:

---

Context:
```powdered sugar (Non-Leaf), icing sugar (Leaf)
flavorer (Non-Leaf), sugar (Non-Leaf)
syrup (Non-Leaf), sugar syrup (Leaf)
sugar (Non-Leaf), brown sugar (Leaf)
sugar (Non-Leaf), beet sugar (Leaf)```

Candidates: corn sugar, lump sugar, caramel,
 granulated sugar, sugarloaf, beet sugar, brown
 sugar

Parent: granulated sugar

Description: granulated sugar is sugar in the form
 of small grains

Interpretation: Granulated sugar is a type of sugar
 that is in the form of small grains, which is a
 specific characteristic of this type of sugar.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 identify the children of granulated sugar. Since
 granulated sugar is a type of sugar, we can look
 for candidates that are more specific types of
 granulated sugar.

Leaf: No

Children: sugarloaf

---

Context:
```wine (Non-Leaf), blush wine (Leaf)
wine (Non-Leaf), canary wine (Leaf)
beverage (Non-Leaf), wine (Non-Leaf)
vinegar (Non-Leaf), wine vinegar (Leaf)
sauce (Non-Leaf), wine sauce (Leaf)```

Candidates: varietal, burgundy, table wine, dubonnet
 , retsina, vintage, cotes de provence, mulled wine,
  blush wine, jug wine, vermouth, fortified wine,
 bordeaux, plonk, altar wine, tokay, red wine,
 riesling, sauterne, liebfraumilch, muscadet, pinot
 blanc, vouvray, verdicchio, sauvignon blanc,
 chardonnay, canary wine, yquem, straw wine, cold
 duck, champagne

Parent: red wine

Description: red wine is wine having a red color
 derived from skins of dark-colored grapes

Interpretation: "Red wine" is a specific type of
 wine, which is a beverage.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 determine the children of "red wine". Since "red
 wine" is a type of wine, we can look for candidates
  that are specific types of red wine.

Leaf: No

Children: burgundy, bordeaux, tokay, riesling, pinot
  blanc, cold duck

---

Context:
```tea (Non-Leaf), ice tea (Leaf)
tea (Non-Leaf), green tea (Non-Leaf)
tea (Non-Leaf), herb tea (Non-Leaf)
herb (Non-Leaf), tea (Non-Leaf)
tisane (Non-Leaf), camomile tea (Leaf)```

Candidates: ice tea, herb tea, cuppa, cambric tea,
 sun tea, oolong, tea bag, black tea, green tea

Parent: sun tea

Description: sun tea is tea made by exposing tea
 leaves steeped in water to the direct rays of the
 sun; usually served with ice

Interpretation: Sun tea is a type of tea that is
 made using a specific method, which is by exposing
 the tea leaves to the direct rays of the sun.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 determine the children of sun tea. Since sun tea is
  a type of tea, we can look for candidates that are
  also types of tea.

Leaf: No

Children: herb tea, green tea, black tea, ice tea

---

Context:
~~~
...
```

Candidates: concoction, frozen food, canned food,
 coloring, dehydrated food, concentrate, meal,
 starches, juice, dairy product, tapioca, flour,
 carob, breadstuff, egg, ingredient, perishable,
 roughage, corn, pearl barley, oat, wild rice, grist
```

```
, malt, buckwheat, cracked wheat, millet, groats,
 criollo

Parent: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Interpretation: "Sweetening" is an ingredient or
 substance added to food to make it sweeter, which
 is a characteristic of some foods.

Reasoning: Let's think step by step in order to
```

```
// Output
Reasoning: Let's think step by step in order to
 determine the children of "sweetening". Since "
 sweetening" is an ingredient or substance added to
 food to make it sweeter, we can look for candidates
  that are specific types of sweetening agents.

Leaf: No

Children: tapioca, carob
```

B.2.5 Completion: Tuned Few-Shot Parents

~~~
// Prompt
Which are the most specific parent concepts of the
 given child concept in a taxonomy considering the
 context?
In your reasoning, state how the parent concepts are
  a supertype of the child concept.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Child: Child concept (subtype) that you need to
 place in a taxonomy.

Description: Description of the child concept.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the parents}. We ...

Interpretation: Description of the child concept in
 relation to the context taxonomy. Infer what is
 meant by the child concept from the context.

Parents: Comma separated list of one or more parents
  of the child concept. Valid parents are in the
 context. If there are no suitable parents, return
 None.

---

Context:
```foodstuff, flour
meal, pea flour
starches, cracked wheat bread
nutriment, wheat germ
starches, gluten bread```

Child: wheat flour

Description: wheat flour is flour prepared from
 wheat

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of wheat flour. We know that wheat
  flour is a type of flour, and flour is a foodstuff
 . Therefore, foodstuff is a parent of wheat flour.
 Additionally, since wheat flour is a specific type
 of flour, flour is also a parent.

Interpretation: Wheat flour is a type of flour
 prepared from wheat, which is a foodstuff.

Parents: foodstuff, flour

---

Context:
```sauce, hard sauce
candy, hard candy
hard candy, barley sugar
cider, sweet cider
hard candy, sourball```

Child: hard cider

Description: hard cider is alcoholic drink from
 fermented cider; `cider' and `cyder' are European (
 especially British) usages for the fermented
 beverage

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of "hard cider". We can see that "
 hard cider" is a type of "cider" because it is a
 specific type of cider that is fermented.

Interpretation: "hard cider" is a specific type of
 cider that is fermented, which makes it an
 alcoholic drink.

Parents: cider

---

Context:
```sauce, plum sauce
pudding, pease pudding
dessert, pudding
dish, pudding
pudding, carrot pudding```

Child: plum pudding

Description: plum pudding is a rich steamed or
 boiled pudding that resembles cake

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of plum pudding. We can see that
 plum pudding is a type of pudding, and pudding is a
  type of dish and dessert. Also, plum sauce is a
 sauce, but it's not directly related to plum
 pudding.

Interpretation: Plum pudding is a specific type of
 pudding that is a dish and a dessert.

Parents: pudding, dish, dessert

---

Context:
```dish, cottage pie
dessert, apple dumpling
quick bread, corn cake
sweet pickle, bread and butter pickle
conserve, apple butter```

Child: tamale pie

Description: tamale pie is a meat mixture covered
 with cornbread topping that resembles a Mexican
 dish

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 find the parents of tamale pie. We can see that
 tamale pie is a type of dish, and it resembles a
 Mexican dish, so it can be classified under dish.

Interpretation: Tamale pie is a type of dish that
 has a cornbread topping, which is a characteristic
 of quick bread.

Parents: dish, quick bread

---

Context:
~~~
...
```
```

~~~
```

Child: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Reasoning: Let's think step by step in order to
~~~

````
Reasoning: Let's think step by step in order to find
  the parents of sweetening. We can see that
 sweetening is related to sugar, as it is something
 added to foods to make them taste sweeter. Sugar is
  a type of nutriment, so nutriment can be a parent
 of sweetening.

Interpretation: Sweetening is something added to
 foods to make them taste sweeter, which is related
 to sugar and nutriment.

Parents: sugar, nutriment
````

B.2.6 Completion: Tuned Few-Shot Children

~~~
// Prompt
Which of the candidates are child concepts (subtypes
 ) of the given parent concept (supertype) in a
 taxonomy?
The context shows existing parents and children
 concepts and whether the children are leaves.
In your reasoning, state how the parent concept is a
  supertype of the selected child concepts.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Candidates: Candidate children of the concept
 separated by commas to select from.

Parent: Parent concept that you need to place in a
 taxonomy.

Description: Description of the parent concept.

Interpretation: Description of the child concept in
 relation to the taxonomy.

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the children}. We ...

Leaf: Whether the parent concept should be added as
 a leaf (has no children). Answer with Yes or No.

Children: Comma separated list of candidates that
 are children of the parent concept in a taxonomy.A
 child concept must be a type of the parent concept.
 Separate with commas.

---

Context:
```foodstuff (Non-Leaf), flour (Non-Leaf)
meal (Non-Leaf), pea flour (Leaf)
starches (Non-Leaf), cracked wheat bread (Leaf)
nutriment (Non-Leaf), wheat germ (Leaf)
starches (Non-Leaf), gluten bread (Leaf)```

Candidates: soybean meal, semolina, wheat flour,
 plain flour

Parent: wheat flour

Description: wheat flour is flour prepared from
 wheat

Interpretation: Wheat flour is a type of flour that
 is prepared from wheat, which is a foodstuff.

Reasoning: Let's think step by step in order to
 Reasoning: We can see that wheat flour is a type of
 flour, and semolina is also a type of flour.
 Therefore, semolina is a type of wheat flour.
 Similarly, plain flour is also a type of wheat
 flour.

Leaf: No

Children: semolina, plain flour

---

Context:
```fricassee (Non-Leaf), chicken stew (Leaf)
goulash (Non-Leaf), pork and veal goulash (Leaf)
mousse (Non-Leaf), chicken mousse (Leaf)
dish (Non-Leaf), chicken and rice (Non-Leaf)
stew (Non-Leaf), beef stew (Leaf)```

Candidates: chicken cordon bleu, croquette, pudding,
  pasta, succotash, chow mein, cottage pie,
 spaghetti and meatballs, poi, jambalaya, roulade,
 swiss steak, tamale pie, bacon and eggs, enchilada,
  barbecue, meat loaf, patty, lobster thermidor,
 potpie, coquilles saint jacques, sauerbraten, coq
 au vin, sauerkraut, tetrazzini, moussaka, refried
 beans, fondue, dolmas, steak au poivre, viand,
 sukiyaki, timbale, porridge, scallopine, seafood
 newburg, lutefisk, frittata, omelet, soup, pepper
 steak, spanish rice, galantine, barbecued wing,
 salisbury steak, sashimi, couscous, fish and chips,
  piece de resistance, veal cordon bleu, hash, chop
 suey, scotch egg, scrambled eggs, poached egg,
 chicken marengo, casserole, terrine, macedoine,
 pizza, meatball, welsh rarebit, osso buco, kishke,
 chicken paprika, carbonnade flamande, shirred egg,
 scampi, mold, taco, pork and beans, bitok, french
 toast, burrito, scrapple, haggis, pheasant under
 glass, maryland chicken, beef bourguignonne, boiled
  dinner, rijsttaffel, chicken and rice, schnitzel,
 kabob, beef wellington, risotto, paella, tempura,
 special, souffle, mousse, fish stick, tostada, frog
  legs, chili, snack food, ramekin, ham and eggs,
 boiled egg, chicken provencale, rissole, pilaf,
 applesauce, moo goo gai pan, kedgeree, stew, tossed
  salad, molded salad, chicken salad, crab louis,
 salmagundi, potato salad, herring salad, fruit
 salad, tuna fish salad, tabbouleh, pasta salad,
 coleslaw, lamb curry, raita, fruit custard,
 bavarian cream, creme caramel, knish, apple
 turnover, pirogi, samosa

Parent: lamb curry

Description: lamb curry is curry made with lamb

Interpretation: Lamb curry is a type of dish that is
  made with lamb and has a curry flavor.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 identify the children of lamb curry. Lamb curry is
 a type of dish that is made with lamb and has a
 curry flavor. Therefore, it is a subtype of curry
 and dish. Since it is a specific type of curry, it
 can be considered a parent concept.

Leaf: No

Children: None

---

Context:
```water soluble vitamin (Non-Leaf), vitamin c (Leaf
 )
b complex vitamin (Non-Leaf), biotin (Leaf)
filling (Non-Leaf), lekvar (Leaf)
beverage (Non-Leaf), rickey (Non-Leaf)
rum (Non-Leaf), grog (Leaf)```

Candidates: vitamin bc, choline, vitamin b6, biotin,
  niacin, vitamin b2, pantothenic acid

Parent: vitamin bc

Description: vitamin bc is a B vitamin that is
 essential for cell growth and reproduction
~~~

~~~
Interpretation: Vitamin bc is a type of B vitamin,
 which is essential for cell growth and reproduction
 .

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 identify the children of vitamin bc. Vitamin bc is
 a type of B vitamin, so its children should be more
  specific types of vitamin bc.

Leaf: No

Children: choline, vitamin b6

---

Context:
```dish (Non-Leaf), cottage pie (Leaf)
dessert (Non-Leaf), apple dumpling (Leaf)
quick bread (Non-Leaf), corn cake (Leaf)
sweet pickle (Non-Leaf), bread and butter pickle (
 Leaf)
conserve (Non-Leaf), apple butter (Leaf)```

Candidates: irish soda bread, oatcake, date bread,
 date nut bread, banana bread, nut bread, corn cake,
  corn dab, spoon bread, skillet corn bread,
 cornpone, johnnycake, ashcake, baking powder
 biscuit, buttermilk biscuit, rolled biscuit, drop
 biscuit, bran muffin, corn muffin, popover, drop
 scone, chicken cordon bleu, croquette, pudding,
 pasta, succotash, chow mein, cottage pie, spaghetti
  and meatballs, poi, jambalaya, roulade, swiss
 steak, tamale pie, bacon and eggs, enchilada,
 barbecue, meat loaf, patty, lobster thermidor,
 potpie, coquilles saint jacques, sauerbraten, coq
 au vin, sauerkraut, tetrazzini, moussaka, refried
 beans, fondue, dolmas, steak au poivre, viand,
 sukiyaki, timbale, porridge, scallopine, seafood
 newburg, lutefisk, frittata, omelet, soup, pepper
 steak, spanish rice, galantine, barbecued wing,
 salisbury steak, sashimi, couscous, fish and chips,
  piece de resistance, veal cordon bleu, hash, chop
 suey, scotch egg, scrambled eggs, poached egg,
 chicken marengo, casserole, terrine, macedoine,
 pizza, meatball, welsh rarebit, osso buco, kishke,
 chicken paprika, carbonnade flamande, shirred egg,
 scampi, mold, taco, pork and beans, bitok, french
 toast, burrito, scrapple, haggis, pheasant under
 glass, maryland chicken, beef bourguignonne, boiled
  dinner, rijsttaffel, chicken and rice, schnitzel,
 kabob, beef wellington, risotto, paella, tempura,
 special, souffle, mousse, fish stick, tostada, frog
  legs, chili, snack food, ramekin, ham and eggs,
 boiled egg, chicken provencale, rissole, pilaf,
 applesauce, moo goo gai pan, kedgeree, stew, tossed
  salad, molded salad, chicken salad, crab louis,
 salmagundi, potato salad, herring salad, fruit
 salad, tuna fish salad, tabbouleh, pasta salad,
 coleslaw, lamb curry, raita, fruit custard,
 bavarian cream, creme caramel, knish, apple
 turnover, pirogi, samosa

Parent: tamale pie

Description: tamale pie is a meat mixture covered
 with cornbread topping that resembles a Mexican
 dish

Interpretation: Tamale pie is a type of dish that
 has a cornbread topping, which is a type of quick
 bread.

Reasoning: Let's think step by step in order to
 Reasoning: Let's think step by step in order to
 identify the children of tamale pie. We can see
 that tamale pie is a type of dish that has a
 cornbread topping, which is a type of quick bread.
 Therefore, tamale pie is related to quick bread.

Leaf: No

Children: corn cake, corn dab, spoon bread, skillet
 corn bread, cornpone, johnnycake, ashcake

---

Context:
```
...
~~~

Candidates: vitamin, kosher, finger food, wheat germ
 , course, stodge, dish, dainty, puree, mince, meal,
  evaporated milk, yak's milk, sour milk, semi
 skimmed milk, colostrum, homogenized milk, coconut
 milk, scalded milk, chocolate milk, certified milk,
  pasteurized milk, raw milk, low fat milk, formula,
  cows' milk, nonfat dry milk, mother's milk,
 acidophilus milk, skim milk, corn sugar, lump sugar
 , caramel, granulated sugar, sugarloaf, beet sugar,
  brown sugar

Parent: sweetening

Description: sweetening is something added to foods
 to make them taste sweeter

Interpretation: Sweetening is something added to
 foods to make them taste sweeter, which is related
 to sugar and nutriment.

Reasoning: Let's think step by step in order to
```

```
// Output
Reasoning: Let's think step by step in order to
 identify the children of sweetening. Sweetening is
 something added to foods to make them taste sweeter
 , which is related to sugar and nutriment.
 Therefore, its children should be types of
 sweetening agents.

Leaf: No

Children: granulated sugar, lump sugar, corn sugar,
 caramel, brown sugar, beet sugar
```

B.2.7 Generation: Parents

```
// Prompt
Which are the most specific parent concepts of the
 given child concept in a taxonomy considering the
 context?
In your reasoning, state how the parent concepts are
  a supertype of the child concept.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Child: Child concept (subtype) that you need to
 place in a taxonomy.

Description: Description of the child concept.

Taxonomy Description: Description of the taxonomy

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the parents}. We ...

Interpretation: Description of the child concept in
 relation to the context taxonomy. Infer what is
 meant by the child concept from the context.

Parents: Comma separated list of one or more parents
  (supertypes) of the child concept. A parent
 concept must be a more general type of the child
 concept. If there are no suitable existing parents,
  invent them.

---

Context:
```baking ingredients, baking powder
baking ingredients, baking soda
baking ingredients, purpose flour
baking ingredients, unsalted butter
baking ingredients, brown sugar
cooking ingredients, spices and seasonings
```

```
cooking ingredients, oils
spices and seasonings, vanilla extract
spices and seasonings, garlic cloves
spices and seasonings, salt pepper
oils, vegetable oil
oils, olive oil```

Child: baking ingredients

Description: baking ingredients

Taxonomy Description: The topic of the taxonomy
 appears to be "Cooking Ingredients" or "Kitchen
 Pantry Items." The rough structure of the taxonomy
 could be organized into categories such as: 1. Oils
  (e.g., olive oil, vegetable oil) 2. Baking
 Ingredients (e.g., flour, sugar, baking powder,
 baking soda) 3. Spices and Seasonings (e.g., garlic
  cloves, salt, pepper, vanilla extract) 4. Dairy
 Products (e.g., unsalted butter) 5. Miscellaneous (
 e.g., brown sugar) This taxonomy could be further
 divided into subcategories or refined to include
 more specific ingredients or cooking-related items.

Reasoning: Let's think step by step in order to
```

```
// Output
Reasoning: Let's think step by step in order to find
  the most specific parent concepts of "baking
 ingredients". We can observe that "baking
 ingredients" is already a category in the given
 taxonomy, and it has several subtypes such as "
 baking powder", "baking soda", "purpose flour", "
 unsalted butter", and "brown sugar". This suggests
 that "baking ingredients" is a subtype of a more
 general category. Since the taxonomy is about "
 Cooking Ingredients" or "Kitchen Pantry Items", a
 suitable parent concept for "baking ingredients"
 could be "cooking ingredients".

Interpretation: The child concept "baking
 ingredients" refers to a category of ingredients
 used in baking, which is a part of the broader
 topic of cooking ingredients.

Parents: cooking ingredients
```

B.2.8 Generation: Children

~~~
// Prompt
Which of the candidates are child concepts (subtypes
 ) of the given parent concept (supertype) in a
 taxonomy?
The context shows existing parents and children
 concepts and whether the children are leaves.
In your reasoning, state how the parent concept is a
  supertype of the selected child concepts.
Do not add additional comments or information, only
 return the output in the described format.

---

Input description.

Context: List of existing parent-child (
 supertype-subtype) relations in the taxonomy.

Candidates: Candidate children of the concept
 separated by commas to select from.

Parent: Parent concept that you need to place in a
 taxonomy.

Description: Description of the parent concept.

Interpretation: Description of the child concept in
 relation to the taxonomy.

Previous Reasoning: past Reasoning: with errors

Previous Leaf: past Leaf: with errors

Previous Children: past Children: with errors

Instructions: Some instructions you must satisfy

---

Follow the following format.

Reasoning: Let's think step by step in order to ${
 produce the children}. We ...

Leaf: Whether the parent concept should be added as
 a leaf (has no children). Answer with Yes or No.

Children: Comma separated list of candidates that
 are children of the parent concept in a taxonomy.A
 child concept must be a type of the parent concept.
 Separate with commas.

---

Context:
```baking ingredients (Non-Leaf), baking powder (
 Leaf)
baking ingredients (Non-Leaf), baking soda (Leaf)
baking ingredients (Non-Leaf), purpose flour (Leaf)
baking ingredients (Non-Leaf), unsalted butter (Leaf
 )
baking ingredients (Non-Leaf), brown sugar (Leaf)
cooking ingredients (Non-Leaf), spices and
 seasonings (Non-Leaf)
cooking ingredients (Non-Leaf), oils (Non-Leaf)
spices and seasonings (Non-Leaf), vanilla extract (
 Leaf)
spices and seasonings (Non-Leaf), garlic cloves (
 Leaf)
spices and seasonings (Non-Leaf), salt pepper (Leaf)
oils (Non-Leaf), vegetable oil (Leaf)
oils (Non-Leaf), olive oil (Leaf)```

Candidates: spices and seasonings, oils

Parent: baking ingredients

Description: baking ingredients

Interpretation: The child concept "baking
 ingredients" refers to a category of ingredients
 used in baking, which is a part of the broader
 topic of cooking ingredients.

Previous Reasoning: Reasoning: Let's think step by
 step in order to determine the children of "baking
 ingredients". We can see that "baking ingredients"
 is a category of ingredients used in baking, which
 is a part of the broader topic of cooking
 ingredients. Since "spices and seasonings" and "
 oils" are both categories of ingredients, but not
 specifically used only in baking, they cannot be
 considered as children of "baking ingredients".

Previous Leaf: No

Previous Children: Reasoning: Let's think step by
 step in order to determine the children of "baking
 ingredients". We can see that "baking ingredients"
 is a category of ingredients used in baking, which
 is a part of the broader topic of cooking
 ingredients. Since "spices and seasonings" and "
 oils" are both categories of ingredients, but not
 specifically used only in baking, they cannot be
 considered as children of "baking ingredients".
 Leaf: No Children:

Instructions: reasoning lets think step by step in
 order to determine the children of baking
 ingredients. we can see that baking ingredients is
 a category of ingredients used in baking, which is
 a part of the broader topic of cooking ingredients.
  since spices and seasonings and oils are both
 categories of ingredients, they cannot be
 considered as children of baking ingredients., but
 not specifically used only in baking are not valid
 children, since they are not in the candidates.

Reasoning: Let's think step by step in order to
~~~

```
// Output
Reasoning: Let's think step by step in order to
 determine the children of "baking ingredients". We
 can see that "baking ingredients" is a category of
 ingredients used in baking, which is a part of the
 broader topic of cooking ingredients. Since "spices
  and seasonings" and "oils" are categories of
 ingredients used in cooking in general, not
 specifically in baking, they cannot be considered
```

```
 as children of "baking ingredients".

Leaf: No

Children:
```