Grouping Entities with Shared Properties using Multi-Facet Prompting and Property Embeddings

Amit Gajbhiye[1], **Thomas Bailleux**[2], **Zied Bouraoui**[2], **Luis Espinosa Anke**[1,3], **Steven Schockaert**[1]

[1] Cardiff University, UK [2] CRIL CNRS & University of Artois, France [3] AMPLYFI, UK

{bouraoui,bailleux}@cril.fr {espinosa-ankel,schockaerts1}@cardiff.ac.uk

Abstract

Methods for learning taxonomies from data have been widely studied. We study a specific version of this task, called *commonality identification*, where only the set of entities is given and we need to find meaningful ways to group those entities. While LLMs should intuitively excel at this task, it is difficult to directly use such models in large domains. In this paper, we instead use LLMs to describe the different properties that are satisfied by each of the entities individually. We then use pre-trained embeddings to cluster these properties, and finally group entities that have properties which belong to the same cluster. To achieve good results, it is paramount that the properties predicted by the LLM are sufficiently diverse. We find that this diversity can be improved by prompting the LLM to structure the predicted properties into different facets of knowledge.[1]

1 Introduction

Taxonomies specify how the entities from a given domain can be organized into meaningful categories. Such resources provide the backbone of knowledge graphs (Hogan et al., 2022) and ontologies (Chandrasekaran et al., 1999), and they play a prominent role in many natural language processing and machine learning tasks (Ma et al., 2018; Huang et al., 2019; Karamanolakis et al., 2020; Zheng et al., 2021). Accordingly, the problem of constructing and enriching these resources has received considerable attention. For instance, a popular variation of this task is *taxonomy learning*, which involves organizing a given set of terms into a tree or directed acyclic graph (DAG), i.e. to uncover the hypernym relationships that exist among them (Bordea et al., 2015). Another variation focuses on *taxonomy enrichment*: given a new term and its definition, find its most suitable position in an existing taxonomy (Jurgens and Pilehvar, 2016).

In this paper, we consider yet another variant of this task, called *commonality identification*: given a set of entities, find groupings of these entities which have some salient property in common. This task was recently studied by Gajbhiye et al. (2023). It is closely related to *hypernym discovery* (Camacho-Collados et al., 2018), which aims to provide a set of hypernyms for a given entity. In our cases, however, we are specifically looking for properties that are shared by different entities, which in practice often requires going beyond the kind of hypernyms that are typically considered in taxonomy learning and hypernym discovery benchmarks (e.g. "objects which are found at the beach").

To find commonalities, Gajbhiye et al. (2023) fine-tuned a BERT bi-encoder to identify shared commonsense properties among a given set of entities. Intuitively, we might expect that recent LLMs can straightforwardly outperform such a strategy. However, directly using LLMs to solve this task is not feasible in large domains, where we may need to find commonalities among tens of thousands of entities. Therefore, in this paper, we exploit LLMs indirectly, namely for generating properties of the entities. Commonalities then correspond to sets of entities for which the same property was predicted. There are two key challenges with this approach. First, we need to ensure that the properties which are generated are sufficiently diverse, since we may not know in advance what facets of knowledge the given dataset focuses on. For instance, to uncover commonalities involving the concept *banana*, we may need to predict hypernyms such as *fruit* and *healthy snack*, or commonsense properties such as *yellow* and *curved*, or more specialized properties such as *potassium-rich* and *contains isoamyl acetate*. We address this by prompting LLMs to generate properties belonging to different facets. The second challenge comes from the fact that the same

[1] Our code and datasets are available at https://github.com/thomas-bllx/grouping-entities-mfp.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 15612–15627
November 4-9, 2025

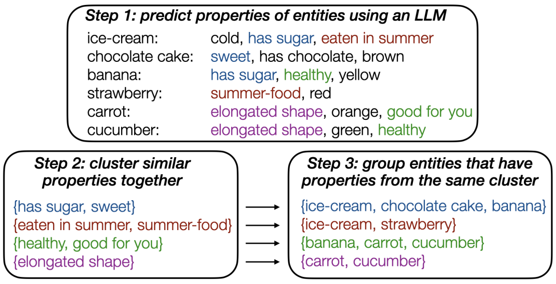

Figure 1: Schematic overview of our approach.

property may be described in different ways. For instance, the LLM may generate *potassium-rich* for one entity and *high in potassium* for another. To address this, we cluster properties using pre-trained embeddings and then group entities for which properties from the same cluster were predicted. This process is illustrated for a toy example in Figure 1.

2 Related Work

Taxonomy learning (Bordea et al., 2015, 2016) typically refers to the following task: given a set of terms from a particular domain, organize them into a directed acyclic graph (DAG). Most approaches first collect a set of candidate hypernym relations and then use some strategy to reduce noise and construct a coherent DAG. Our task in this paper differs from taxonomy learning in that we do not assume that any hypernym terms are given. *Taxonomy enrichment* (Jurgens and Pilehvar, 2016) refers to the task of adding a new term to an existing taxonomy. As a definition for the new term is typically provided, this task is closer to definition modelling (Noraset et al., 2017) than commonality identification. Thirdly, the task of *hypernym discovery* (Camacho-Collados et al., 2018) involves generating a comprehensive set of hypernyms for a given term. TaxoLlama (Moskvoretskii et al., 2024) is a prominent example of an LLM-based method that can be used for hypernym discovery, being a Llama model that was fine-tuned on hypernym relations from WordNet. While we similarly use LLMs to generate properties, we rely on pre-trained LLMs, without fine-tuning. This is due to the fact that comprehensive datasets of (sufficiently diverse) properties are not readily available. Finally, the task of *commonality identification* was introduced by Gajbhiye et al. (2023), who relied on a BERT bi-encoder that was trained to predict commonsense properties. For each property in a predefined vocabulary, they used this bi-encoder to find a corresponding set of entities. Subsequently, they refined this set using a Natural Language Inference model.

3 Methodology

Let a (potentially large) set of entities $\mathcal{E}$ be given. The task we consider is to group these entities in meaningful categories. Note that these categories may overlap, and some entities may not belong to any of the categories, which makes this task different from clustering. For instance, given a set of people, we may want to induce categories of people born in the same city, people with the same profession, or employees of the same company.

Our approach uses an LLM to generate a set of properties for each entity. To encourage the model to generate diverse properties, we ask it to list the facets of interest and to specify relevant properties for each of these facets. To clarify what we mean by a *facet*, we present the model with one in-context demonstration. As we assume no prior knowledge about the domain of interest, we always use the same example, listing commonsense properties of bananas[2]. After prompting the model, we end up with a set of (facet, property) pairs for each entity in $\mathcal{E}$. Since the generation process is non-deterministic, we prompt the model 10 times for each entity, as this increases the likelihood that shared properties among the entities in $\mathcal{E}$ will be uncovered. Next, we organize the (facet, property) pairs that were obtained from the LLM into clusters. To this end, we first represent each pair (f, p) as a text string $\gamma(f, p)$, using a template such as "f: p" or "p in terms of f". We then obtain an embedding $e(f, p)$ of this string using an off-the-shelf LLM2Vec text encoder (BehnamGhader et al., 2024). These embeddings are subsequently clustered using affinity propagation. Let us write $C_1, ..., C_\ell$ for the corresponding sets of (property, facet) pairs. Finally, with each cluster C_i, we associate a category of entities E_i, namely the set of entities associated with at least one of the property-facet pairs that belong to C_i:

$$E_i = \{e \,|\, C_i \cap P_e \neq \emptyset\}$$

where P_e is the set of (facet, property) pairs that were predicted by the LLM for entity e.

4 Evaluation

Experimental set-up We refer to our method as Multi-Facet Prompting (MFP). Unless stated oth-

[2]The full prompt can be found in Appendix D.

	CS	SemEval				Wikidata					
		Food	**Scie**	**Equi.**	**Envi.**	**Huma.**	**Poli.**	**Cath.**	**Moun.**	**Song**	**Game**
LLM2Vec clusters	24.0	22.7	30.6	22.8	31.2	18.5	15.5	13.5	12.4	18.5	17.6
MClu	49.3	23.5	38.2	37.2	34.3	58.4	56.1	23.5	38.9	59.1	46.7
TaxoLlama	21.1	30.0	32.3	29.0	41.5	14.7	19.1	4.1	16.0	27.2	22.0
MFP (literal match)	6.1	14.6	15.5	14.9	6.4	6.5	6.6	9.6	15.7	6.3	8.0
MFP (no facets)	48.3	22.1	38.5	39.8	36.7	61.2	58.4	27.1	46.1	63.4	52.1
MFP (LLM2Vec-Mistral)	**60.3**	31.2	**46.0**	52.3	49.2	77.3	**70.6**	31.1	**53.9**	72.4	61.0
MFP 5-shot	52.7	**31.3**	44.5	**55.0**	**54.0**	**79.1**	70.2	**32.1**	52.3	**76.2**	**65.8**

Table 1: Intrinsic evaluation of category induction methods, in terms of category coverage score %.

erwise, we verbalize facet-property pairs as $\gamma(f, p)$ = "f: p" and encode these descriptions with a pre-trained LLM2Vec model based on Mistral-7B. To generate the properties, we use Llama3-8B with a temperature of 0.6. When prompting the LLM to generate facet-property pairs, we normally provide a single in-context demonstration. However, for comparison, we will also consider a 5-shot variant.[3]

Baselines We consider three main baselines. The first baseline uses affinity propagation to cluster LLM2Vec embeddings of the entities (using the same Mistral-7B based model as for MFP). Each cluster is treated as an entity category E_i. Second, we use the multi-facet embeddings from Kteich et al. (2024) (*MClu*) to obtain 10 different embedding spaces (each focusing on different facets). We apply affinity propagation to each of the embedding spaces and treat all of the corresponding clusters as entity categories. Third, we use TaxoLlama (Moskvoretskii et al., 2024) to generate 10 hypernyms for each entity. Let P_e be the set of hypernyms predicted for entity e. Each hypernym h that was predicted for at least two entities induces an entity category $E_h = \{e \mid h \in P_e\}$. We also consider a variant of MFP in which the clustering step is omitted (denoted as *literal match*). In this case, each (property, facet) pair that was predicted for at least two entities induces a category, similar as for the TaxoLlama baseline. Finally, we consider a variant of MFP in which we omit the facets when prompting the LLM (shown as *no facets*).

Intrinsic Evaluation We evaluate our model on a number of datasets where a set of entities is given, and where ground truth categories are available. First, we re-purpose an outlier detection dataset (*CS*) based on commonsense categories from Kteich et al. (2024). The dataset consists of everyday concepts (e.g. airplane, banana, shoes), which have been assigned to 10 categories, each corresponding to a commonsense property (e.g. dangerous). Second, we re-purpose four standard taxonomy construction datasets (Bordea et al., 2015, 2016): *Food*, *Science*, *Equipment* and *Environment*. In our setting, we only provide the model with the leaf nodes from the gold taxonomy. Every other term is then viewed as a category (containing all leaf terms that are descendants). Finally, we constructed a new dataset based on Wikidata. For each problem instance, we selected an entity type (e.g. humans), a Wikidata property p (e.g. cause of death) and a corresponding value v (e.g. gunshot wound). We then selected positive and negative examples. The positive examples are entities e of the considered type such that Wikidata contains the triple (e, p, v). The negative examples are entities e for which this is not the case, but for which Wikidata contains a triple (e, p, v') with $v' \neq v$. In total we constructed 42 such datasets, covering six entity types: *humans*, *politicians*, *cathedrals*, *mountains*, *songs* and *games*. We report the average performance across all problem instances of a given entity type.[4]

Let $G_1, \ldots, G_k$ be the ground truth categories and $E_1, \ldots, E_q$ the predicted categories. For evaluation, we use the following category coverage score:

$$score = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1}^{q} \frac{|G_i \cap E_j|}{|G_i \cup E_j|}$$

In other words, for each ground truth category, we compute the Jaccard similarity to the closest predicted category, and then average the resulting similarities. Note that this is a recall-oriented metric, and methods are not penalized for predicting additional categories. This is because the set of ground

[3] Further details about our experimental set-up can be found in Appendix A.

[4] Appendix A.2 provides further details about the intrinsic evaluation benchmarks. Appendix A.4 presents an example to illustrate the intrinsic evaluation process.

	Wine	Econ	Olym	Tran	SUMO
GloVe*	14.2	14.1	9.9	8.3	34.9
ConCN*	31.3	32.4	29.7	20.9	52.6
ConEmb-F†	31.2	31.8	30.4	20.9	51.7
MClu (ConEmb-F)†	39.9	36.3	32.9	23.1	55.4
LLM2Vec clusters	40.2	37.5	33.5	24.8	56.9
TaxoLlama	40.6	38.1	34.6	26.9	57.4
MFP (LLM2Vec)	**41.4**	**38.4**	**35.3**	**27.9**	**58.2**

Table 2: Results for ontology completion in terms of F1 (%). * Results taken from Li et al. (2023b); † results from Kteich et al. (2024).

	F1
Base model†	49.2
BiEnc properties*	50.9
MClu◇	51.3
LLM2Vec clusters	51.6
TaxoLlama	51.5
MFP (LLM2Vec)	**52.4**

Table 3: Results for ultra-fine entity typing. * Results taken from Gajbhiye et al. (2023); † results from Li et al. (2023a); ◇ results from Kteich et al. (2024).

truth categories cannot be assumed to be exhaustive. We mainly use this score to compare variants of our model, which generate similar numbers of categories. However, care is needed when comparing methods that generate different numbers of categories with the proposed score.

Table 1 summarizes the results. Our method consistently outperforms the baselines. *TaxoLlama* performs particularly poorly on CS, which supports our hypothesis that standard hypernym discovery methods struggle with commonsense properties. The MFP (literal match) variant overall performs very poorly, which demonstrates the importance of the clustering step. The results also clearly show the importance of facet-based prompting. Finally, using 5 in-context demonstrations boosts the results on some, but not all datasets.

Further experiments can be found in Appendix B, where we compare LLM2Vec with alternative embedding methods, compare affinity propagation with alternative clustering algorithms, study the sensitivity of the results to the verbalization of the property-facet pairs, analyze the importance of prompting the model 10 times, and analyze the role of the LLM temperature. In Appendix C we present a qualitative analysis.

Ontology Completion In Table 2 we evaluate the default configuration of our method on the downstream task of ontology completion. This task involves predicting plausible rules which are missing from a given ontology. Following Kteich et al. (2024), we use this task for evaluating the predicted categories. In this case, the entities are the concept names that appear in the ontology. For each category that is predicted, we add a new concept to the ontology, and include rules to encode that this new concept subsumes each of the members of that category. We then apply the ontology completion framework from Li et al. (2019) to the resulting enriched input, and assess whether it allows us to predict the missing rules more accurately. This framework also requires pre-trained concept embeddings. We use the ConCN embeddings from Li et al. (2023b) for this purpose. Further details on this task can be found in Appendix A.3.1.

The results clearly show the effectiveness of the proposed prompting strategy, with substantial and consistent gains over the current state-of-the-art. Interestingly, the LLM2Vec and TaxoLlama baselines also outperform the previous methods, although they consistently underperform our method.

Ultra-fine Entity Typing Ultra-fine entity typing (UFET) is a multi-label classification problem, which involves assigning entity types from a large set of around 10K candidate labels to mentions of entities (Choi et al., 2018). We use our method to find categories among these 10K labels. Following Gajbhiye et al. (2023), we then add these categories as additional synthetic labels to the training data. In particular, if label l is predicted to belong to category C, then for each training example that has the label l, we add C as an additional label. We then train a standard UFET model on the augmented label set. During evaluation, we omit the newly added synthetic labels from any predictions. Essentially, this strategy uses the predicted categories as a form of semantic regularization during training. For these experiments, we use BERT-base as the entity mention encoder and use the UFET model from Pan et al. (2022). Further details on this task can be found in Appendix A.3.2.

The results in Table 3 confirm the effectiveness of the proposed prompting strategy. We again find that even our two considered baselines outperform the previous methods, while our method achieves the best results.

5 Conclusions

We have proposed a simple but effective strategy for identifying commonalities among large sets of entities, based on prompting an LLM to describe properties that belong to different facets. We found the use of facets to be important for achieving the best results. The generated properties are then clustered to determine entity categories of interest. We found this approach to outperform the state-of-the-art in two downstream applications: ontology completion and ultra-fine entity typing.

Acknowledgments

This work was supported by EPSRC grants EP/V025961/, EP/W003309/1 and by ANR-22-CE23-0002 ERIANA.

Limitations

Our focus in this paper has been on showing the potential of LLMs for finding commonalities among large sets of entities (i.e. settings where the task itself cannot be solved using LLMs directly). However, we have not attempted to compare the performance of different LLMs for this purpose, and have only evaluated a small number of different prompts. It is thus likely that further performance increases can be obtained by varying some of the default choices that we have made.

In our evaluation, we have focused on assessing the ability of our method to find meaningful categories, but have not attempted to construct full taxonomies, e.g. as trees or directed acyclic graphs. It seems plausible that using hierarchical clustering (rather than affinity propagation) would allow us to construct meaningful taxonomies, but evaluating this has been left as a topic for future work.

Our strategy to obtain commonalities by clustering property embeddings also has some limitations, which are discussed in detail in the qualitative analysis in Appendix C. A possible solution which could be considered in future work is to fine-tune the LLM on the initial set of commonalities, and repeating the proposed strategy with the resulting model, or to provide the initial set of commonalities as part of the prompt. However, such approaches would significantly increase the computational overhead of the method. Another possibility would be to apply some kind of post-processing on the entity clusters, for instance by using an NLI model to filter spurious entities, as was done by Gajbhiye et al. (2023).

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *CoRR*, abs/2404.05961.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. SemEval-2015 task 17: Taxonomy extraction evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, Colorado. Association for Computational Linguistics.

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.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 712–724, New Orleans, Louisiana. Association for Computational Linguistics.

B. Chandrasekaran, John R. Josephson, and V. R. Benjamins. 1999. What are ontologies, and why do we need them? *IEEE Intell. Syst.*, 14(1):20–26.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Amit Gajbhiye, Zied Bouraoui, Na Li, Usashi Chatterjee, Luis Espinosa-Anke, and Steven Schockaert. 2023. What do deck chairs and sun hats have in common? uncovering shared properties in large concept vocabularies. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10587–10596, Singapore. Association for Computational Linguistics.

Amit Gajbhiye, Luis Espinosa-Anke, and Steven Schockaert. 2022. Modelling commonsense properties using pre-trained bi-encoders. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3971–3983, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2022. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37.

Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 573–581. ACM.

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.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8489–8502, Online. Association for Computational Linguistics.

Hanane Kteich, Na Li, Usashi Chatterjee, Zied Bouraoui, and Steven Schockaert. 2024. Modelling commonsense commonalities with multi-facet concept embeddings. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1467–1480, Bangkok, Thailand. Association for Computational Linguistics.

Na Li, Zied Bouraoui, and Steven Schockaert. 2019. Ontology completion using graph convolutional networks. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, volume 11778 of *Lecture Notes in Computer Science*, pages 435–452. Springer.

Na Li, Zied Bouraoui, and Steven Schockaert. 2023a. Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11744–11756, Singapore. Association for Computational Linguistics.

Na Li, Hanane Kteich, Zied Bouraoui, and Steven Schockaert. 2023b. Distilling semantic concept embeddings from contrastively fine-tuned language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 216–226. ACM.

Jianxin Ma, Peng Cui, Xiao Wang, and Wenwu Zhu. 2018. Hierarchical taxonomy aware network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1920–1929. ACM.

Viktor Moskvoretskii, Ekaterina Neminova, Alina Lobanova, Alexander Panchenko, and Irina Nikishina. 2024. TaxoLLaMA: WordNet-based model for solving multiple lexical semantic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2331–2350, Bangkok, Thailand. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3259–3266. AAAI Press.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4317–4323. ijcai.org.

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.

Jianming Zheng, Fei Cai, Wanyu Chen, Wengqiang Lei, and Honghui Chen. 2021. Taxonomy-aware learning for few-shot event detection. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3546–3557. ACM / IW3C2.

A Additional Experimental Details

A.1 Implementation Details

For our MFP model, we generate facet-property pairs using a pre-trained Llama3-8B[5] model with a temperature of 0.6. Descriptions are encoded with a pre-trained LLM2Vec model based on Mistral-7B[6]. For the *LLM2Vec clusters* baseline, we use the same pre-trained LLM2Vec model based on Mistral. For the *TaxoLlama* baseline, we rely on a pre-trained model based on Llama 3[7]. We prompt the model to generate up to 10 hypernyms for each entity. In practice, the number of hynpernyms provided varies from 3 to 10. For *MClu*, we used the bi-encoder model from Kteich et al. (2024) based on BERT-large.[8] We first obtain the concept and facet embeddings for all entities from the considered dataset. As proposed by Kteich et al. (2024), we then cluster the facet embeddings in 10 clusters,

[5]`https://huggingface.co/meta-llama/Meta-Llama-3-8B`

[6]`https://huggingface.co/McGill-NLP/LLM2Vec-Mistral-7B-Instruct-v2-mntp`

[7]`https://huggingface.co/VityaVitalich/TaxoLlama3.1-8b-instruct`

[8]Available from `https://github.com/hananekth/facets_concepts_embeddings`

and use these clusters to obtain 10 facet-specific embeddings for each entity. Subsequently, we use affinity propagation in each of the 10 resulting embedding spaces, and use the clusters from all embedding spaces as predictions of entity categories.

In our experiments, we use affinity propagation to cluster with the default hyper-parameters of scikit-learn. The damping factor was set to 0.5, with a maximum of 200 iterations and a stopping criterion after 15 iterations without change. The affinity used was the Euclidean distance and the preference value was set to the median of the data similarities.

A.2 Intrinsic Evaluation Tasks

Table 4 provides an overview of the considered evaluation datasets.

A.2.1 SemEval Taxonomies

The taxonomies for *Food*, *Science* and *Equipment* come from TexEval-1 (Bordea et al., 2015)[9], whereas the *Environment* taxonomy comes from the second iteration of TexEval (Bordea et al., 2016)[10].

A.2.2 Wikidata Benchmark

Table 5 provides an overview of the 42 properties that were used for creating the WikiData benchmark. This dataset was constructed by, first, identifying frequent properties among popular Wikidata entities (popularity measured by QRank, a popularity metric based on pageviews[11]), then manually selecting suitable categories and `<predicate, relation>` instances that were above a minimum frequency threshold of 5. For each positive example, we sampled at most 2x negatives which, as discussed in Section 4, have the constraint that they must exist in the same relation but with a different predicate. For example, for the category *videogames* and property *single-player*, a positive example would be *Super Mario Bros* (the 1985 version game), whereas a negative example would be *League of Legends*. In terms of dataset statistics, we can see from Figure 2 a mostly balanced distribution between positive and negative examples, and imbalanced going both ways for mountains and humans, on one side, and politicians, on the other.

[9] https://alt.qcri.org/semeval2015/task17/index.php?id=data-and-tools
[10] https://alt.qcri.org/semeval2016/task13/
[11] https://github.com/brawer/wikidata-qrank.

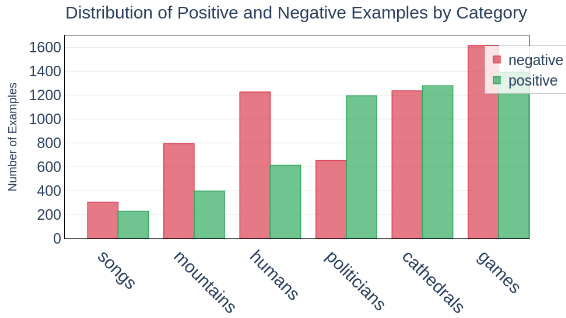

Figure 2: WikiData positive and negative examples.

A.3 Extrinsic Evaluation Tasks

We now provide some additional details about the two extrinsic evaluation tasks.

A.3.1 Details of Ontology Completion Task

Ontologies are essentially sets of rules, encoding how the different concepts from a given domain are related. By relying on logical connectives and quantifiers, they can encode knowledge in a more fine-grained way than what is possible in standard taxonomies. For instance, we can consider the following rule:

$$\mathit{Female}(x) \wedge \mathit{Child}(y, x) \rightarrow \mathit{Mother}(x)$$

This rule expresses the knowledge that a female person who has a child is a mother. The task of ontology completion consists in predicting plausible rules which are missing from a given ontology. Li et al. (2019) treat this problem as a binary classification problem. Specifically, starting from a held-out rule, they replace one of the predicates by a placeholder $\star$, leading to a so-called rule template. For instance, for the aforementioned rule, we end up with the following template:

$$\tau(\star) = \mathit{Female}(x) \wedge \mathit{Child}(y, x) \rightarrow \star(x)$$

For any given concept C, we can consider the rule $\tau(C)$. The task, considered by Li et al. (2019) is to predict, for a given concept C, whether $\tau(C)$ is a valid rule. In the previous example, $C = \mathit{Mother}$ is a positive example, whereas $C = \mathit{Father}$ or $C = \mathit{Bicycle}$ would be negative examples. To solve this task, they proposed a GNN-based formulation, using a graph that captures the given set of rules. The nodes of this graph correspond to the concepts from the given ontology. The performance of this method is highly sensitive to the input features which are used for these concept

Dataset	Entities	Ground truth categories
CS	Everyday concepts	Commonsense properties
Food	Leaf nodes from the Food taxonomy	Non-leaf nodes from the Food taxonomy
Science	Leaf nodes from the Science taxonomy	Non-leaf nodes from the Science taxonomy
Equipment	Leaf nodes from the Equipment taxonomy	Non-leaf nodes from the Equipment taxonomy
Environment	Leaf nodes from the Environment taxonomy	Non-leaf nodes from the Environment taxonomy
Humans	Wikidata entities of type human	Property-value pairs
Politicians	Wikidata entities of type politician	Property-value pairs
Cathedrals	Wikidata entities of type cathedral	Property-value pairs
Mountains	Wikidata entities of type mountain	Property-value pairs
Songs	Wikidata entities of type song	Property-value pairs
Games	Wikidata entities of type game	Property-value pairs

Table 4: Overview of the intrinsic evaluation dataset. Each dataset consists of a group of entities and some ground truth categories (which are interpreted as sets of entities that have some property or hypernym).

Entity type	Property	Value
humans	cause of death	gunshot wound
	given name	David
	occupation	politician
	award received	Star on Hollywood Walk of Fame
	educated at	Lee Strasberg Theatre and Film Institute
	record label	Columbia Records
	instrument	piano
	has works in the collection	Victoria and Albert Museum
	languages spoken, written or signed	Spanish
politicians	award received	Grand Cross of the Legion of Honour
	manner of death	homicide
	member of political party	Republican Party
	sex or gender	male
	occupation	poet
	position held	Roman emperor
	given name	John
cathedrals	architectural style	Gothic architecture
	religion or worldview	Catholicism
	heritage designation	part of UNESCO World Heritage Site
	country	Italy
	dedicated to	Virgin Mary
mountains	instance of	volcano
	continent	Asia
	mountain range	Andes
	country	Switzerland
	parent peak	Finsteraarhorn
songs	has characteristic	debut single
	genre	hip hop music
	nominated for	Grammy Award for Song of the Year
	performer	Jennifer Lopez
	composer	Karl Martin
	record label	EMI
	genre	J-pop
	lyrics by	Brian May
games	subclass of	Star Wars video game
	business model	free-to-play
	platform	Xbox 360
	distributed by	Steam
	software engine	Unreal Engine 3
	genre	massively multiplayer online game
	uses	isometric view
	game mode	single-player video game

Table 5: Overview of the properties that were included in the Wikidata benchmark.

nodes, which is why this task has been chosen for evaluating concept embeddings in previous work (Li et al., 2023b). Kteich et al. (2024) showed that encoding commonalities can also help to improve results. In this paper, we follow their methodology.

In particular, we start by setting the set of entities $\mathcal{E}$ as the set of all concept names that appear in the given ontology. We then apply our method for identifying commonalities. Let $E_i = \{e_1, ..., e_m\}$ be one of the predicted entity categories. We then add the following rules to the ontology:

$$e_1(x) \rightarrow Z_i(x)$$
$$...$$
$$e_m(x) \rightarrow Z_i(x)$$

where Z_i is a fresh concept name. We repeat this for each of the identified commonalities. In this way, the identified commonality is encoded in the ontology, and thus taken into account when constructing the graph. As the results in Table 2 show, this can significantly improve the results.

A.3.2 Details of UFET Task

Ultra-fine entity typing (UFET) is a multi-label classification task, which consists in assigning semantic types to an entity which is mentioned in a given sentence (Choi et al., 2018). In contrast to the standard entity typing task, where broad types such as *person* or *place* are used, in the case of UFET a large set of around 10K candidate labels are considered. Consider the following example:

> In contrast to the male way of thinking, in which priority has always been given to considerations of political and economic power, Annette Lu has emphasized "soft national power."

The task is to assign all labels that apply to the highlighted entity span ("Annette Lu"), which for this example are: *person*, *officeholder*, *president*, *official*, *leader*, *incumbent*. We start from the approach from Pan et al. (2022), which uses a fine-tuned BERT encoder with a soft prompt of the form "x[P1]m[P2][P3][MASK]", where x is the given sentence, m is the highlighted entity span, and [P1], [P2] and [P3] are special tokens with learnable embeddings. The labels are predicted by a linear multi-label classification head on top of the contextualized representation of the [MASK] token.

Gajbhiye et al. (2023) used the same approach as Pan et al. (2022), but augmented the training data with the identified commonalities. In particular, they consider the set of all entity types as the set of entities $\mathcal{E}$. For each entity category $E_i = \{e_1, ..., e_m\}$ that was discovered, they introduce a new (artificial label) l_i. They then add this label to each training example that was labeled with at least one of the types in E_i. For test examples, any predictions of these artificial labels l_i are simply discarded. Essentially, this approach regularizes the embeddings that are learned by the model, by ensuring that types which have something meaningful in common can be linearly separated from the others. In this paper, we follow the same approach, but use our proposed commonality identification method, rather than the bi-encoder based strategy from Gajbhiye et al. (2023).

The approach from Gajbhiye et al. (2023) follows a strategy proposed by Li et al. (2023a), who instead used clusters of pre-trained concept embeddings to identify commonalities. The main limitation of this clustering strategy, however, is that the commonalities that can be identified are not sufficiently diverse. To address this, Kteich et al. (2024) proposed a multi-facet concept embedding strategy, which led to improved UFET results. Essentially, they learn 10 different concept embeddings, each focusing on different kinds of properties. By computing clusters from each of the 10 embedding spaces, a more diverse set of categories can be obtained, compared to standard embeddings. We report for this variant as MClu.

A.4 Illustration of the Intrinsic Evaluation Process

We illustrate the intrinsic evaluation process, using a toy example. For this toy example (which is also illustrated in Figure 1), we start from the following taxonomy:

- *ice-cream* is-a *dessert*
- *chocolate cake* is-a *dessert*
- *banana* is-a *fruit*
- *strawberry* is-a *fruit*
- *banana* is-a *sweet thing*
- *dessert* is-a *sweet thing*
- *carrot* is-a *vegetable*

- *cucumber* is-a *vegetable*
- *fruit* is-a *healthy thing*
- *vegetable* is-a *healthy thing*

The set of entities $\mathcal{E}$ that we consider for the evaluation consists of the leaf nodes from this taxonomy: *ice-cream*, *chocolate cake*, *banana*, *strawberry*, *carrot*, *cucumber*. The non-leaf concepts are: *dessert*, *fruit*, *sweet thing*, *vegetable*, *healthy thing*. Each non-leaf concept is used as a ground truth category. These categories formally correspond to the set of "leaf descendants" of the corresponding terms. In particular, the ground truth categories are:

$$
\begin{aligned}
G_1 &= \{\textit{ice-cream}, \textit{chocolate cake}\} \\
G_2 &= \{\textit{banana}, \textit{strawberry}\} \\
G_3 &= \{\textit{ice-cream}, \textit{chocolate cake}, \textit{banana}, \\
&\qquad \textit{strawberry}\} \\
G_4 &= \{\textit{carrot}, \textit{cucumber}\} \\
G_5 &= \{\textit{banana}, \textit{strawberry}, \textit{carrot}, \textit{cucumber}\}
\end{aligned}
$$

Our method first predicts properties for the entities. For simplicity, we do not consider facet-based prompting for this example. Let us now assume that the following properties are predicted:

- *ice-cream*: *cold*, *has sugar*, *eaten in summer*
- *chocolate cake*: *sweet*, *has chocolate*, *brown*
- *banana*: *has sugar*, *healthy*, *yellow*
- *strawberry*: *summer-food*, *red*
- *carrot*: *elongated shape*, *orange*, *good for you*
- *cucumber*: *elongated shape*, *green*, *healthy*

Next, the properties are clustered based on their pre-trained embedding. Suppose this gives us the following clusters:

$$
\begin{aligned}
C_1 &= \{\textit{has sugar}, \textit{sweet}\} \\
C_2 &= \{\textit{eaten in summer}, \textit{summer-food}\} \\
C_3 &= \{\textit{healthy}, \textit{good for you}\} \\
C_4 &= \{\textit{elongated shape}\}
\end{aligned}
$$

Note that properties such as *yellow* are ignored at this point, as they were only predicted for a single entity and were not clustered together with other properties. The four property clusters now give rise to four corresponding entity categories:

$$
\begin{aligned}
E_1 &= \{\textit{ice-cream}, \textit{chocolate cake}, \textit{banana}\} \\
E_2 &= \{\textit{ice-cream}, \textit{strawberry}\} \\
E_3 &= \{\textit{banana}, \textit{carrot}, \textit{cucumber}\} \\
E_4 &= \{\textit{carrot}, \textit{cucumber}\}
\end{aligned}
$$

For evaluation, we check for each ground truth category whether we have a corresponding predicted category:

- For G_1, the most similar predicted category is E_1 with a Jaccard similarity of 2/3.
- For G_2, the most similar predicted category is E_2 with a Jaccard similarity of 1/3.
- For G_3, the most similar predicted category is E_1 with a Jaccard similarity of 3/4.
- For G_4, the most similar predicted category is E_4 with a Jaccard similarity of 1.
- For G_5, the most similar predicted category is E_3 with a Jaccard similarity of 3/4.

The overall evaluation is the average of these 5 scores.

		SemEval			
	CS	**Food**	**Scie**	**Equi.**	**Envi.**
MFP (SBERT)	58.3	30.0	45.9	46.7	42.8
MFP (BiEnc pre-trained)	**62.3**	26.3	44.6	42.6	39.2
MFP (BiEnc generated)	58.8	31.0	**46.0**	42.9	45.7
MFP (LLM2Vec-Llama)	58.8	28.6	44.8	49.7	**49.3**
MFP (LLM2Vec-Mistral)	60.3	**31.2**	**46.0**	**52.3**	49.2

Table 6: Analysis of embedding methods on some of the intrinsic evaluation datasets, in terms of category coverage score %.

B Additional Experiments

Comparison of Embedding Methods Table 6 shows the results we obtained with a number of variants of the property embedding method. For these experiments, we focus on the Commonsense and SemEval datasets. First, the table includes a variant where SBERT is used (Reimers and Gurevych, 2019). Next, we included a variant where a LLM2Vec model based on Llama 3.0 is used[12], instead of the Mistral-based version that

[12] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-supervised

	CS	SemEval Food	Scie	Equi.	Envi.
AFFINITY PROPAGATION					
$\lambda = 0.5$	**60.3**	31.2	46.0	**52.3**	49.2
$\lambda = 0.7$	55.3	30.7	**47.0**	46.8	50.1
$\lambda = 0.9$	55.3	30.4	46.6	46.8	49.6
HDBSCAN					
$\alpha = 0.5$	53.4	28.8	44.1	42.2	42.2
$\alpha = 1.0$	52.1	22.4	36.4	39.4	38.9
$\alpha = 1.5$	34.9	2.2	2.0	1.4	15.2
K-MEANS					
K=1000	56.0	18.2	34.4	24.6	32.9
K=2500	57.2	26.3	41.2	45.2	44.2
K=5000	54.0	**32.4**	44.7	49.4	**58.0**

Table 7: Analysis of clustering variants on some of the intrinsic evaluation datasets, in terms of category coverage score %.

was used in the main paper. We also consider a variant in which the property encoder of a BERT bi-encoder is used for encoding the descriptions. We consider two versions: using the pre-trained encoder from Gajbhiye et al. (2022) (denoted as *BiEnc pre-trained*) and training a bi-encoder from scratch on the set of (property, facet) pairs that were obtained from the LLM (denoted as *BiEnc generated*). For the latter case, we follow the same training methodology as Gajbhiye et al. (2022), where positive examples are (entity, verbalization) pairs, with each verbalization a phrase of the form "facet: property". For each positive example, we include 5 negative examples, which we obtain by corrupting positive examples (by swapping property-facet verbalizations with those from other concepts).

The results in Table 6 show that SBERT and the bi-encoder variants can achieve competitive results. The variant with the pre-trained bi-encoder achieves the best results on the Commonsense benchmark, which is perhaps unsurprising since this model was pre-trained on commonsense properties. The *BiEnc generated* variant is also competitive, achieving results which sometimes match the LLM2Vec based approach.

Comparison of Clustering Methods We analyze the performance of MFP for a number of different clustering methods. For these experiments, we use the default configuration of our model, based on LLM2Vec-Mistral, changing only the clustering method. For all methods, we use the scikit-learn implementations. Recall that for the main experiments, we used affinity propagation, with the damping factor λ set to its default value of 0.5. In Table 7, we compare this with two other choices: $\lambda = 0.7$ and $\lambda = 0.9$. We also show results for HDBScan[13], where we vary the distance scaling parameter α from 0.5 to 1.5, with 1.0 being the default value. Finally, we evaluate K-means[14], where the number of clusters K is varied from 1000 to 5000.

The results in Table 7 show that affinity propagation is a good choice overall, achieving either the best results or close to the best results in most cases. The main exception is the Environment taxonomy, where K-means with $K = 5000$ performs best. HDBScan consistently underperforms affinity propagation. The performance of K-means is sensitive to the number of clusters, where higher values of K are generally better.

	CS	SemEval Food	Scie	Equi.	Envi.
f: p	**60.3**	31.2	**46.0**	**52.3**	49.2
p in terms of f	54.7	30.8	45.7	50.4	**50.7**
p as a feature of f	59.1	28.5	44.4	51.7	49.2
p as it pertains to f	56.3	**31.8**	44.5	52.1	48.9
p	51.2	24.5	40.1	41.1	39.4

Table 8: Comparison of different verbalizations of property-facet pairs, in terms of category coverage score %.

Nr. iterations	CS	SemEval Food	Scie	Equi.	Envi.
1	44.6	20.1	32.7	30.8	34.4
3	48.8	26.0	42.8	42.9	40.7
5	49.2	28.1	41.9	44.7	46.8
10	**60.3**	**31.2**	**46.0**	**52.3**	**49.2**

Table 9: Analysis of the impact of the number of prompt iterations (for the standard MFP model).

Verbalization of Property-Facet Pairs Table 8 compares the effect of changing how property-facet pairs are verbalized. We compare the following prompts:

1. f: p (e.g. "colour: yellow")
2. p in terms of f (e.g. "yellow in terms of colour")
3. p as a feature of f (e.g. "yellow as a feature of colour")

[13] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.HDBSCAN.html
[14] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

		SemEval			
Temperature	**CS**	**Food**	**Scie**	**Equi.**	**Envi.**
0.5	59.6	30.4	45.9	50.1	46.0
0.6	60.3	31.2	46.0	**52.3**	**49.2**
0.7	60.1	30.9	**46.3**	52.0	49.1
0.8	**60.4**	**31.9**	46.0	50.9	48.1
0.9	58.3	30.3	45.7	48.5	48.3
1.0	56.9	30.1	40.3	46.9	45.4

Table 10: Analysis of the impact of different temperatures (for the standard MFP model).

4. p as it pertains to f (e.g. "yellow as it pertains to colour")

5. p (e.g. "yellow")

For this experiment, we use our default model, with LLM2Vec as the encoder. We can see that the default prompt ("f: p") consistently performs well. While there is some variation across the different prompts, the first four prompts perform broadly comparably. However, the last prompt, which omits the name of the facet f underperforms substantially. This clearly shows the importance of including the facet name, which is often needed to disambiguate the meaning of a property.

Analysis of Prompt Iterations In our experiments, when generating property-facet pairs with Llama 3, we repeat the generation process 10 times, to obtain a larger and more diverse set of properties than would be possible with a single prompt. Table 9 analyzes the impact of repeating the prompt, showing the results that are obtained when using different numbers of iterations. For these results, we have used our default MFP model (with LLM2Vec-Mistral embeddings and the "f: p" verbalization strategy). As we can see, the number of prompt iterations has a strong effect on the results, where even with 5 iterations the results are clearly below those of the full model.

Analysis of LLM Temperature Table 10 shows the results for different choices of the temperature, when generating property-facet pairs with Llama 3. For these results, we have used our default MFP model (with LLM2Vec-Mistral embeddings and the "f: p" verbalization strategy). As for the main experiment, in all cases, we prompt Llama 3 a total of 10 times. The results show that temperatures between 0.6 and 0.8 generally lead to the best results.

C Qualitative Analysis

LLM Outputs In Table 11, we list a few selected example concepts from the SemEval food taxonomy, alongside properties generated with the base prompt and the faceted variant. These examples illustrate the broader diversity of the generated properties, when facets are used, as well as their granularity and lower overlap. For instance, for the concept *506 Chinese noodles*, we find more ingredients (egg, rice), mentions of their shape, and more granularity in general (from "used in many dishes" to naming them, e.g., *lo mein* or *chow mein*). Similarly with *Ritz crackers*, the faceted prompt obtains more specificity related to their taste, for *breakfast sandwich* a similar pattern with the type of bread, and finally, for *ponzu* we obtain a geographical property in "origin".

Property Clusters Table 12 shows some of the property clusters that were found for the Wikidata benchmark. For this analysis, we used the MFP model with LLM2Vec-Mistral embeddings, but rather than using our default verbalization strategy, we only used the property p for verbalizing property-facet pairs (i.e. the variant corresponding to the bottom row of Table 8). The examples in the table illustrate how the model is able to identify a diverse range of commonalities. For instance, for the dataset about the architectural style of cathedrals (top part of Table 12), we can see categories corresponding to different aspects of cathedrals, including their associated religion (3rd example), the architectural style (5th example) and building materials (6th example). For the dataset about the country of mountains (middle part of Table 12), we can see examples related to the presence of mountain huts (2nd example), the nature of the hiking trails or mountaineering routes (3rd and 4th example), and the volcanic nature of the mountain (6th example), among others. For the dataset about single-player video games (bottom part of Table 12), we can see examples related to the genre of the game (first four examples), the geographic location where the game takes place (6th example) and the platform where the game was released (last example).

Table 12 also illustrates some of the problems with the MFP strategy. First, because the prompt only relies on the name of the entity, problems may arise when that name is ambiguous. This is clearly illustrated by the first example in the middle part of the table, which refers to hotels and restau-

Entity	Method	Properties
Chinese noodles	baseline	["made from wheat flour", "long and thin shape", "often served in soups", "popular in China", "used in stir-fries", "cooked in boiling water", "dried or fresh forms", "used in many dishes", "has high carbohydrate content", "often paired with vegetables"]
	faceted	{"type": ["dried", "fresh"], "material": ["wheat flour", "rice flour", "egg"], "shape": ["long", "short", "curved"], "cooking method": ["boiling", "steaming", "stir-frying"], "origin": ["China", "Asia"], "texture": ["soft", "firm", "chewy"], "popular dishes": ["lo mein", "chow mein", "pad thai"]}}
Ritz crackers	baseline	["type of snack", "made by Nabisco", "used for fruit and nut", "has a crunchy texture", "often eaten with peanut butter", "available in stores", "popular snack", "has a distinctive taste", "comes in various flavors", "is a packaged food"]
	faceted	{"category": ["snack", "cracker", "food"], "brand": ["Ritz"], "taste": ["salted", "buttery"], "texture": ["crunchy"], "used for": ["snacking", "making sandwiches", "making croutons"], "origin": ["United States", " Nabisco"], "shelf life": ["long-lasting"], "pairing": ["cheese", "meat", "jam"]}}
breakfast sandwich	baseline	["typically eaten in morning", "contains egg and meat", "served on bread", "popular in fast food", "often with cheese", "can be customized", "convenient meal", "appeals to many", "may include hash browns", "popular in diners"]
	faceted	{"components": ["eggs", "bacon", "cheese", "bread"], "type of bread": ["English muffin", "bagel", "toasted bun"], "fillings": ["scrambled eggs", "fried eggs", "avocado", "ham"], "size": ["small", "medium", "large"], "occasion": ["breakfast", "brunch", "lunch"], "cuisine": ["American", "fast food"], "price range": ["affordable", "moderate"]}}
ponzu	baseline	["Japanese citrus sauce", "made from soy sauce and citrus juice", "used as marinade or dressing", "commonly served with sushi"]
	faceted	{"concept": "ponzu", "facet_properties_dict": {"category": ["Japanese condiment", "sauce", "dressing"], "ingredients": ["soy sauce", "citrus juice", "vinegar", "sugar"], "taste": ["sour", "sweet", "savory"], "used for": ["dressing salads", "marinating meat", "making sauces", "adding flavor"], "origin": ["Japan", "Chinese influence"], "common pairing": ["sashimi", "tempura", "noodles", "rice"]}}

Table 11: Examples of model responses for entities from the SemEval food taxonomy, for both the baseline (*MFP no facets*) and faceted prompt (i.e. the default *MFP* configuration).

rants that have been named after a mountain. We can also see this issue in the third example in the bottom part, where most of the properties refer to computer games which are set in space. However, we can also see properties which do not refer to games as at all, such as *recipient: astronauts* and *recipient: space agency*. These properties were predicted for the game *Martian Memorandum*, which the model appears to have confused with an actual memorandum of understanding between space agencies. Some of the categories involve properties that would apply to all entities in the considered datasets, such as the second example in the top part (referring to cathedrals as places of worships) and the fifth example in the bottom part (referring to games as being entertaining).

Further issues arise because of how the properties are clustered. For instance, in the fourth example in the top part of the table, the model has clustered all properties that refer to the construction date, which results in an uninformative cluster that would ideally cover all cathedrals. Similarly, in the seventh example in the middle part, mountains of all different heights are grouped together. There are also examples of semantically meaningful clusters where one or two properties have been included which do not belong. For instance, the seventh example in the bottom part describes games that have been released on Apple devices, except that the property *platforms:linux* was also included. Furthermore, some issues arise due to the fact that the facet name was not included in the verbalization, for the variant that was used here. This is most notable in the sixth example in the bottom part, which mixes different types of references to geographic locations. For instance, *origin: japan* was predicted for *pac-man*, referring to where the game was originally developed and released. On the other hand, *location: scotland* was predicted for *true golf classics: wicked 18*, capturing the fact that parts of the game are set in Scotland. A final issue (not shown in the table) is that there are sometimes multiple clusters that refer to the same (or a very similar) property. For instance, in the

Dataset	Property clusters
Gothic architecture (cathedrals)	1. used for: tours, used for: concerts, activities: concerts, notable events: frequent concerts, activities: tours, current use: tours, features: organ concerts, ...
	2. used for: worship, purpose: worship, function: prayer, rituals and practices: prayer, purpose: prayer, ...
	3. cultural significance: korean catholicism, type: catholic, celebration: roman catholic tradition, connected to: catholic church, religion: catholic, type: roman catholic, significance: roman catholic cathedral, ...
	4. construction date: 15th century, construction: 16th century, age: 16th century, construction period: 19th century, construction: 17th century, era: 14th century, construction date: 13th century, construction: 20th century, construction period: 11th century, ...
	5. style: baroque, architectural style: art deco, architecture: rococo, architecture style: baroque, architectural influences: italian baroque architecture, features: baroque interior, notable for: baroque tower, notable for: baroque facade, ...
	6. architecture: white stone, building materials: marble, construction material: marble, ...
	7. artworks: stained glass, artistic significance: famous for stained glass, features: stained glass, ...
	8. features: dome, iconic features: dome, notable features: dome, ...
Switzerland (mountains)	1. facilities: hotels, features: restaurant, infrastructure: restaurant, type: hotel, category: restaurant, accommodation: hotel, function: hotel, ...
	2. infrastructure: mountain huts, access: mountain refuge, hiking trails: mountain huts, features: mountain hut, hiking: mountain hut nearby, tourism: mountain lodge, ...
	3. route: rock climbing, route types: rock climbing, required skills: rock climbing, recreational use: rock climbing, used for: rock climbing, access: mountain climbing, mountaineering: ice climbing, difficulty: mountaineering, ...
	4. recreational activities: hiking, recreational use: hiking, popular for: hiking, notable for: hiking trail, route: hiking trail, recommended equipment: hiking boots, route: hiking path, difficulty level: challenging hike, ...
	5. importance: tourist attraction, popularity: tourist spot, type: outback attraction, use: tourist attraction, accessibility: popular tourist destination, landmark: popular tourist spot, ...
	6. geography: volcanic, activity: volcanic activity, formation: volcanic activity, geological significance: volcanic activity, notable for: ongoing volcanic activity, mountain type: volcanic, popularity: popular among volcanologists, scientific interest: study of volcanic activity, ...
	7. height: over 2,000 meters, elevation: over 2,000 meters, height: over 3,000 meters, height: over 4,000 meters, height: over 1,000 meters, ...
	8. notable for: unique shape, appearance: distinctive shape, known for: unique appearance, ...
	9. routes: west ridge, climbing route: north-west ridge, features: ridge, natural feature: ridge, geography: ridge, most popular route: south ridge, physical features: ridge, ...
Single-player video game (games)	1. genre: stealth, gameplay mechanics: stealth, game genre: stealth, features: silent running, activities: spying, gameplay style: infiltration, ...
	2. genre: shooting, gameplay mechanics: shooting, abilities: shooting, ...
	3. recipient: astronauts, usage: spacecraft, game setting: galaxy, storyline: space station, aircraft: spacecraft, features: space stations, hazard: space debris, recipient: space agency, ...
	4. gameplay: solving puzzles, objective: solving puzzles, gameplay: solve puzzles, storyline: solve puzzles, ...
	5. purpose: entertainment, relevance: entertainment, use: entertainment, functions: entertaining , ...
	6. origin: japan, location: scotland, locations: paris, location: egypt, geographical region: europe, setting: mexico, location: italy, origin: germany, ...
	7. platforms: macos, platform: ios, platform: macintosh, platform: mac os x, platforms: linux, platform: mac
	8. used for: video games, used in: video games, influence on popular culture: video games, type: video game, platforms: video games, game type: computer game, medium: electronic game, genre: electronic game, game type: video game, game series: video game, format: video game, ...

Table 12: Examples of property clusters that were found for the Wikidata benchmark. Results were obtained using the MFP strategy with LLM2Vec-Mistral embeddings, when only the property p is used for verbalizing the property-facet pairs.

Property clusters
1. taste: sweet, taste: tangy, taste: cooling, taste: savory, taste: spicy, taste: nutty, taste: salty, taste: sour, taste: earthy, ...
2. preparation: steamed, cooking method: boiled, cooking method: grilling, cooking method: toasting , ...
3. diet: insects, diet: worm, diet: pollen, diet: fly, diet: grass, diet: plant sap, diet: flakes, diet: hay, diet: carrion, ...
4. diet: sugar water, diet: sweet liquids, diet: sugary drinks
5. crust: flaky, crust: crispy, crust: thin, crust: baked, crust: thick
1. purpose: deliver payload, purpose: logging, purpose: drawing, function: reduce noise, purpose: sifting, ...
2. shape: round, shape: rectangular, shape: oval, shape: triangular, shape: flat, shape: diamond, shape: thick, shape: ring, ...
3. size: small, size: medium, size: large, size: tiny
4. texture: crumbly, texture: fibrous, texture: flaky, texture: crusty, texture: chewy, texture: fluffy, texture: soft, ...
5. features: clothing, features: microwave, features: engine, features: seat, features: faucet, features: turntable, ...

Table 13: Examples of property clusters that were found for the Commonsense benchmark. Results were obtained using the MFP strategy with LLM2Vec-Mistral embeddings, when property-facet pairs are verbalized as $f : p$.

mountains dataset, there are several similar clusters referring to the difficulty of hiking trails.

The examples in Table 12 highlighted a number of issues arising from the fact that facet names were not included when verbalizing the properties. Table 13 illustrates the advantages and risks of including the facet name. For this table, we have used the Commonsense dataset, and clustered properties using our default strategy, verbalizing these properties as $f : p$. The clusters in the top half of the table illustrate cases where the inclusion of the facet name can be regarded as beneficial. For instance, the first cluster groups entities that have a taste, i.e. edible things. While this cluster is rather general (i.e. not focusing on specific tastes), modelling the property of being edible may be useful. Along similar lines, the second cluster groups things which needs to be cooked, i.e. edible things which are not eaten raw. The third cluster essentially groups non-predatory animals. The fourth cluster focuses on a highly specific property (grouping the entities *butterfly*, *hornet*, *wasp* and *housefly*). Similarly, the fifth cluster captures the property of having a crust (grouping the entities *pizza* and *pie*).

As illustrated in the bottom half of Table 13, including the facet name in the verbalization also leads to issues. The first example groups things which have a purpose. While this set of entities is not entirely trivial (e.g. it would not include living things), it is probably too general and abstract to be useful for most downstream applications. Similarly, the second cluster groups things which have a shape, the third cluster groups things which have a size, and the fourth cluster groups things which have a texture. Noting that the commonsense dataset only contains concrete nouns, these clusters are non-informative. The last example groups a rather incoherent set of entities, based on the fact that they have properties involving the facet "features".

D LLM Prompts

We use the following prompt for predicting properties of entities using an LLM.

```
<|begin_of_text|><|start_header_id|>system<|
    end_header_id|>

You are a contestant in the general knowledge
    quiz contest and always answer all kinds of
    common sense questions accurately.
All output must include only valid JSON like the
     following example {"concept": concept, "
    facet_properties_dict": {facet: [list of
    properties with each property less than ten
    words long]}}.
Don't add any explanations before and after the
    JSON.
If you don't know the answer, please don't share
     false information.<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of the banana, write its
    different facets and most salient properties
     under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "banana", "facet_properties_dict":
    {"category": ["fruit", "healthy snack", "
    food"], "color": ["yellow", "green"], "shape
    ": ["curved"], "taste": ["sweet"], "
    nutritional content": ["rich in potassium",
    "high in sugar"], "used for": ["healthy
    snack", "making cakes", "making smoothie"],
    "located at": ["banana tree", "supermarket",
     "fridge", "fruit bowl"]}}<|eot_id|>

For the concept of the <CONCEPT>, write its
    different facets and most salient properties
     under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

The prompt with 5 in-context demonstrations is given as follows:

```
<|begin_of_text|><|start_header_id|>system<|
    end_header_id|>
```

```
You are a contestant in the general knowledge quiz contest and always answer all kinds of common sense questions accurately.
All output must include only valid JSON like the following example {"concept": concept, "facet_properties_dict": {facet: [list of properties with each property less than ten words long]}}.
Don't add any explanations before and after the JSON.
If you don't know the answer, please don't share false information.<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of teacher, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "teacher", "facet_properties_dict": {"role": ["teaching", "guiding students", "assessing", "classroom management"], "workplace": ["school", "college", "university", "training centre"], "skill": ["creativity", "communication", "patience", "subject expertise", "organisation"], "tool": ["textbook", "whiteboard", "computer", "educational materials"], "impact": ["knowledge transfer", "skill development", "inspiration", "career guidance"], "types": ["primary school teacher", "high school teacher", "college professor", "special education teacher"]}}<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of bamboo, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "bamboo", "facet_properties_dict": {"category": ["plant", "grass family", "natural resource"], "growth": ["fast-growing", "tall", "woody stems"], "used for": ["construction", "furniture", "paper", "textiles", "food for pandas"], "location": ["Asia", "tropical regions", "forests", "gardens"], "appearance": ["green stem", "hollow", "jointed nodes"], "environmental impact": ["afforestation", "carbon sequestration", "erosion control", "renewable resource"], "types": ["clumping bamboo", "running bamboo"]}}<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of Bill Gates, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "Bill Gates", "facet_properties_dict": {"profession": ["entrepreneur", "philanthropist", "author"], "founded": ["Microsoft", "Bill & Melinda Gates Foundation"], "known for": ["co-founding Microsoft", "Windows software", "philanthropy"], "born": ["October 28, 1955"], "philanthropy": ["global health", "education", "poverty alleviation", "climate change"], "publications": ["The Road Ahead", "Business @ the Speed of Thought", "How to Avoid a Climate Disaster", "How to Prevent the Next Pandemic"], "awards": ["Presidential Medal of Freedom", "Padma Bhushan", "Knight Commander of the Order of the British Empire"]}}<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of human, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "human", "facet_properties_dict": {"anatomy": ["brain", "heart", "lungs", "skin", "muscles", "bones"], "characteristics": ["bipedal", "intelligent", "social", "emotional", "tool users"], "life stages": ["infant", "child", "adolescent", "adult", "elderly"], "used for": ["social interaction", "cultural development", "technology creation", "reproduction"], "location": ["earth", "cities", "villages", "homes"], "needs": ["food", "water", "shelter", "community", "education"], "communication": ["language", "gestures", "writing", "art"]}}<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of book, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

{"concept": "book", "facet_properties_dict": {"types": ["fiction", "non-fiction", "textbook", "manual", "graphic novel"], "parts": ["cover", "pages", "spine", "chapters", "table of contents"], "used for": ["reading", "education", "entertainment", "reference"], "formats": ["physical", "ebook", "audiobook"], "materials": ["paper", "ink", "binding materials"], "location": ["library", "bookstore", "home", "school", "college"]}}<|eot_id|>
<|start_header_id|>user<|end_header_id|>

For the concept of <CONCEPT>, write its different facets and most salient properties under each facet.<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```