# CONTOR: Benchmarking Strategies for Completing Ontologies with Plausible Missing Rules

**Na Li**[1], **Thomas Bailleux**[2], **Zied Bouraoui**[2], **Steven Schockaert**[3]
[1] University of Shanghai for Science and Technology, China
[2] CRIL CNRS & University of Artois, France [3] CardiffNLP, Cardiff University, UK
{bailleux,bouraoui}@cril.fr schockaerts1@cardiff.ac.uk
li_na@usst.edu.cn

## Abstract

We consider the problem of finding plausible rules that are missing from a given ontology. A number of strategies for this problem have already been considered in the literature. Little is known about the relative performance of these strategies, however, as they have thus far been evaluated on different ontologies. Moreover, existing evaluations have focused on distinguishing held-out ontology rules from randomly corrupted ones, which often makes the task unrealistically easy and leads to the presence of incorrectly labelled negative examples. To address these concerns, we introduce a benchmark with manually annotated hard negatives and use this benchmark to evaluate ontology completion models. In addition to previously proposed models, we test the effectiveness of several approaches that have not yet been considered for this task, including LLMs and simple but effective hybrid strategies.

## 1 Introduction

Ontologies, in the context of artificial intelligence (AI), are essentially sets of rules which describe how the concepts from a given domain are related (Chandrasekaran et al., 1999). They generalise taxonomies by expressing these relationships using logical connectives, which makes it possible to describe conceptual relationships in a more precise way. Throughout this paper, we will use the common description logic syntax for encoding ontology rules (Baader et al., 2004). For instance, an ontology might contain the following rules:

$$\mathsf{Biologist} \sqcap (\exists\, \mathsf{livesIn.UK}) \sqsubseteq \mathsf{UKScientist} \quad (1)$$

$$\mathsf{Geologist} \sqcap (\exists\, \mathsf{livesIn.UK}) \sqsubseteq \mathsf{UKScientist} \quad (2)$$

$$\mathsf{Chemist} \sqcap (\exists\, \mathsf{livesIn.UK}) \sqsubseteq \mathsf{UKScientist} \quad (3)$$

In this syntax, rules are formulated as concept inclusions $X \sqsubseteq Y$, which encode that every instance of the concept $X$ is also an instance of the concept $Y$. The connective $\sqcap$ corresponds to intersection and $\exists r.C$ is the set of concepts that are related through relation $r$ to some concept from $C$. For instance, (1) expresses the knowledge that every biologist who is located somewhere in the UK is called a "UK scientist". Ontologies are most commonly used for defining specialised terminologies, which is crucial for AI systems in many technical domains, including healthcare (Ivanović and Budimac, 2014), building information modelling (Succar, 2009) and geographic information systems (Schuurman, 2006), just to name a few. They can also be used to enrich knowledge graphs (KGs) (Jain et al., 2021; Allemang and Sequeda, 2024) and to improve classifiers by modelling label dependencies (Kulmanov et al., 2018).

Like other types of knowledge bases, ontologies are often incomplete (Ozaki, 2020). With this in mind, we consider the problem of predicting missing rules. We are specifically interested in the problem of predicting rules between known concepts, rather than in enriching ontologies with new concepts. Previous work has shown the feasibility of ontology completion, but the strengths and weaknesses of existing methods are poorly understood, and suitable evaluation frameworks for studying ontology completion are currently lacking. To address this, we make the following contributions:

- We introduce a benchmark for ontology completion. While previous work has focused on distinguishing between held-out rules and randomly corrupted rules, our benchmark includes manually validated hard negatives.

- We empirically compare, for the first time, the two main ontology completion methods from the literature: the approach based on graph neural networks (GNN) and concept embeddings from Li et al. (2019) and the approach based on Natural Language Inference (NLI) from Chen et al. (2023). For the NLI approach,

*Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8316–8334
November 12-16, 2024 ©2024 Association for Computational Linguistics

we carry out the first analysis of Large Language Models (LLMs) for this problem.

- We show that previous ontology completion strategies are highly complementary and we exploit this observation to implement simple but effective hybrid strategies, which significantly outperform the current state-of-the-art.

## 2 Preliminaries

**Ontologies** We focus on description logic ontologies, which express knowledge in terms of *concept inclusions*. Let $\mathcal{C}$ be a set of *atomic concepts* and $\mathcal{R}$ a set of *relations* (often called *roles* in this context). A *concept* is (i) an expression of the form $C$ with $C \in \mathcal{C}$ (ii) an expression of the form $X \sqcap Y$ with $X$ and $Y$ concepts, (iii) an expression of the form $\exists r.X$ with $r \in \mathcal{R}$ and $X$ a concept, or (iv) one of the trivial concepts $\bot$ or $\top$. An ontology is a set of rules of the form $X \sqsubseteq Y$ (also known as concept inclusions), where $X$ and $Y$ are concepts, called the *body* and the *head* of the rule respectively. Equations (1)–(3) provide examples of such rules. Semantically, each concept corresponds to a set of individuals and $X \sqsubseteq Y$ expresses that every individual which belongs to the concept $X$ also belongs to the concept $Y$. The concept $X \sqcap Y$ contains all individuals that belong to both $X$ and $Y$. The concepts $\top$ and $\bot$ correspond to the universe of all individuals and the empty set respectively. Finally, an individual $x$ belongs to $\exists r.X$ if there exists some individual $y$ such that the relation $r(x, y)$ holds and $y$ belongs to $X$. A detailed understanding of description logics is not needed for this paper. For more details, we refer to Baader et al. (2009).

**Task Formulation** Let $\mathcal{K}$ be an ontology and let $X \sqsubseteq Y$ be a rule which is not in $\mathcal{K}$ (nor logically entailed by this ontology). We consider the problem of predicting whether $X \sqsubseteq Y$ is a valid rule, which we treat as a binary classification problem. Note that $X$ and $Y$ may be complex expressions, built from a set of atomic concepts and relations. While $X$ and $Y$ themselves might not appear in $\mathcal{K}$, we assume that these atomic concepts and relations do, i.e. we focus on predicting missing knowledge about a given set of atomic concepts.

Note that in this setting we do not have access to examples of instances of the various concepts. This means that traditional approaches to rule learning cannot be used. Instead, two types of information are available for predicting missing rules. First, we may be able to exploit the structure of the ontology. For instance, from the rules (1)–(3) we may be able to infer that $\mathsf{Biologist}$, $\mathsf{Geologist}$ and $\mathsf{Chemist}$ share something in common, as these rules follow a similar pattern. Whenever we encounter some knowledge that is true for biologists and geologists, we may then assume that this knowledge also plausibly holds for chemists. Second, we can exploit the fact that ontology concepts generally correspond to natural language terms, which means that language models (Chen et al., 2023) or concept embeddings (Li et al., 2019) can be used to inject relevant knowledge.

**Differences with Related Tasks** Various forms of knowledge base completion have been studied in the literature, which differ from ontology completion in fundamental ways. For instance, knowledge graph completion is about predicting missing factual assertions (e.g. Paris is the capital of France) rather than about predicting rules. Rule learning has been studied in many settings, but usually the aim is to learn rules that capture statistical dependencies, based on examples. In the case of ontology completion, the purpose is to learn rules that are universally valid, expressing knowledge that can be used across a wide range of applications (within a given technical domain). The problem of taxonomy expansion is also related, but focuses on relations between individual terms, rather than between more complex expressions. Moreover, the primary focus in that context has been on adding missing terms to existing taxonomies, rather than on predicting missing concept inclusions.

**Benchmarking** Progress on these various knowledge base completion tasks has been crucially enabled by the emergence of standard benchmarks. In the case of ontology completion, the evaluation methodologies that were used in previous work have serious shortcomings. Moreover, the evaluation protocols that were used are not comparable, meaning that little is known about the relative effectiveness of different kinds of strategies. The main aim of this paper is to address this issue, and thus lay the foundations for future work on this topic.

## 3 The CONTOR Benchmark

Existing evaluations of ontology completion have important limitations. For instance, Li et al. (2019) and Chen et al. (2023) use different formulations of the problem, which makes their evaluations not

| | Training | | Dev | | Test | | IAA |
|---|---|---|---|---|---|---|---|
| | pos | neg | pos | neg | pos | neg | $\kappa$ |
| Wine | 69 | 319 | 10 | 31 | 18 | 15 | 80.0 |
| Economy | 384 | 1744 | 44 | 174 | 96 | 81 | 82.0 |
| Olympics | 135 | 621 | 14 | 63 | 34 | 29 | 83.0 |
| Transport | 615 | 2416 | 135 | 142 | 154 | 145 | 81.0 |
| SUMO | 4377 | 21624 | 735 | 749 | 1095 | 998 | 63.0 |
| FoodOn | 45013 | 221429 | 2260 | 2190 | 2370 | 2155 | 62.0 |
| GO | 103184 | 494708 | 5326 | 5012 | 5431 | 5044 | 58.0 |

Table 1: Overview of CONTOR, showing the number of positive and negative examples in the training and test split, as well as the inter-annotator agreement (IAA), in terms of Cohen's $\kappa$, for the negative test examples.

directly comparable. More fundamentally, these methods were evaluated by using randomly corrupted positive examples as negatives. However, such random corruptions are often relatively easy to identify, especially if they were obtained by swapping concepts for semantically distinct alternatives. For instance, it should be easy enough for a baseline system to predict that $\mathsf{Biologist} \sqsubseteq \mathsf{UK}$ is not a plausible rule. Furthermore, randomly corrupted rules might sometimes correspond to semantically valid rules. To address these limitations, we propose CONTOR (Completing Ontology Rules), a benchmark for ontology completion with manually validated hard negative test examples. In Section 5 we will then use this benchmark to compare, for the first time, the different approaches that have been proposed for ontology completion.

**Ontologies** CONTOR is based on seven well-known ontologies: the five ontologies that were used by Li et al. (2019), namely Wine, Economy, Olympics, Transport and SUMO, and two ontologies that were used by Chen et al. (2023), namely FoodOn and the Gene Ontology (GO). Wine, Economy, Olympics and Transport are small domain-specific ontologies. SUMO is used as a representative example of a larger general-domain ontology. Finally, FoodOn and GO are considerably larger than all the others, while being focused on specialised domains. For Wine, Economy, Olympics, Transport and SUMO, we keep 20% of the rules for testing. For FoodOn and GO, we keep 5% for testing. The remaining rules are split into training and development sets. Some basic statistics of the considered ontologies are summarised in Table 1.

**Negative Training Examples** For the training set, using manually annotated negative examples is not feasible, due to the large number of rules which would have to be checked. Therefore, we still use randomly corrupted negative examples for the training set, following the strategies that were proposed in previous work (Li et al., 2019; Chen et al., 2023) (see Appendix B for details). Finally, note that we only add these randomly corrupted rules as negative examples if they cannot be entailed by the positive examples (i.e. the training split of the given ontology).

**Negative Test Examples** To obtain negative examples for the test set, we rely on human annotators to ensure that the corrupted rules are indeed negative examples. Moreover, we aim to select hard negative examples, given that random corruption often leads to nonsensical rules which are too easy to identify. Specifically, for each positive rule $\alpha \sqsubseteq \beta$, we randomly select one of the concepts $C$ appearing in that rule and replace it with another concept $D$. Rather than selecting this concept arbitrarily, we choose $D$ to be among the 5 most similar concepts to $C$, in terms of the cosine similarity between the fastText embeddings of the corresponding names[1]. Note that selecting similar concepts increases the chances that the corrupted rule is actually a valid rule, which means that human annotation is critical in this case. Each corrupted rule was checked by two annotators, who were trained in formal knowledge representation and were fluent in English. The agreement between the annotators is reported in Table 1. For cases where the annotators disagreed, a third annotator was used to decide on the final label.

## 4 Ontology Completion Strategies

Our main aim in this paper is to empirically compare the success of different ontology completion strategies. First, we focus on two strategies from the literature: an NLI based strategy (Chen et al., 2023) and a model based on GNNs and pre-trained concept embeddings (Li et al., 2019). In addition to providing the first comparison of these two strategies, we will also explore a range of variants of the original models. As we will see, both strategies are in fact highly complementary. Inspired by this finding, we also explore some hybrid strategies.

### 4.1 NLI Based Models

To treat ontology completion as an NLI problem, we need to verbalise the body and head of a given

[1] We used the embeddings trained on Common Crawl from https://fasttext.cc/docs/en/english-vectors.html.

candidate rule. For instance, the body of the rule might be translated to the premise "a biologist who lives in the UK" and the head might be translated to the hypothesis "a UK Scientist". While some care is needed about how the premise and hypothesis are formulated (e.g. how concept names referring the multi-word expressions are tokenised), this approach is intuitive and conceptually straightforward. We experiment with three variants: (i) the BERTSubs model (Chen et al., 2023) from the DeepOnto library[2]; (ii) fine-tuned LLMs; and (iii) ChatGPT and GPT-4 in a zero-shot setting.

**DeepOnto** BERTSubs relies on a fine-tuned LM from the BERT family. The model takes a verbalisation of the rule as input and is fine-tuned as a binary classifier.[3] We rely on the implementation of BERTSubs from the DeepOnto library (He et al., 2023b). For a rule of the form $X \sqsubseteq Y$, we use DeepOnto's verbaliser to obtain a textual description of the concepts $X$ and $Y$. This verbaliser tokenises multi-term expressions and describes the logical structure of complex concepts. For instance, $\mathsf{RedWine}$ becomes "red wine" while $\mathsf{Wine} \sqcap \exists \mathsf{hasColor.Red}$ becomes "wine that has color red". To check the validity of $X \sqsubseteq Y$, an input of the form "[CLS] $d_X$ [SEP] $d_Y$ [SEP]" is used, with $d_X$ and $d_Y$ the descriptions of $X$ and $Y$ respectively. Chen et al. (2023) also experimented with variants that explicitly include some of the ontology context of $X$ and $Y$ as part of the input, but since they did not observe a clear benefit from doing this, we do not consider such variants in this paper. Following Chen et al. (2023), we use RoBERTa-base and RoBERTa-large (Liu et al., 2019) as the pre-trained LM.

**LLMs** NLI based methods rely on the LM's internal knowledge to assess the plausibility of a given rule. As such, LLMs might perform better than smaller models. While He et al. (2023a) obtained somewhat disappointing results with LLMs, their analysis was limited to the zero-shot setting. To allow for a more direct comparison, we will use LLMs that are fine-tuned in a similar way to BERTSubs. To complement our experiments with fine-tuned LLMs, we also report results for ChatGPT (`gpt-3.5-turbo`) and GPT-4 (`gpt-4-turbo`).

### 4.2 Concept Embedding Based Models

We now recall the approach from Li et al. (2019), which uses a GNN with pre-computed concept embeddings for predicting plausible rules. These concept embeddings can be obtained from standard word embeddings (Mikolov et al., 2013; Pennington et al., 2014) or distilled from pre-trained language models (Li et al., 2021; Liu et al., 2021a). To see why concept embeddings can be useful for predicting missing rules, note that ontologies often contain large numbers of "parallel rules", expressing essentially the same knowledge for a number of related concepts. Taking the example of (1)–(3), for any concept $X$ whose embedding is similar to that of $\mathsf{Biologist}$, $\mathsf{Geologist}$ and $\mathsf{Chemist}$, we can plausibly infer that there should be a counterpart to these rules for $X$; e.g. we might thus infer:

$$\mathsf{Physicist} \sqcap (\exists\, \mathsf{livesIn.Britain}) \sqsubseteq \mathsf{UKScientist} \quad (4)$$

Note that the justification comes purely from our prior knowledge about the relatedness of the concepts $\mathsf{Biologist}$, $\mathsf{Geologist}$, $\mathsf{Chemist}$ and $\mathsf{Physicist}$. In particular, the concept $\mathsf{UKScientist}$ does not play any role in this process. This strategy can thus also be used if the concept in the head of the rule has a meaning which only makes sense within the context of the given ontology. However, its main drawback is that it can only be applied if suitable parallel rules are present. Li et al. (2019) developed an embedding based method for ontology completion which uses a Graph Neural Network (GNN) to implement the aforementioned intuition, which first abstracts rules as *rule templates* and then uses these templates to define a graph.

**Rule Templates** A *unary rule template* is obtained by replacing one of the concepts appearing in a rule from the ontology by a placeholder. For instance, the rules (1)–(3) are all instances of the following template:

$$\rho(X) = X \sqcap (\exists \mathsf{livesIn.UK}) \sqsubseteq \mathsf{UKScientist}$$

The notion of rule template allows us to treat the problem of predicting missing rules as a binary classification problem: for a given concept $X$, decide whether $\rho(X)$ is a valid rule or not. Similarly, a *binary rule template* is obtained by replacing two concepts by a placeholder. We can for instance consider the following template:

$$\rho(X, Y) = \mathsf{Biologist} \sqcap (\exists \mathsf{livesIn}.X) \sqsubseteq Y$$

[2] https://krr-oxford.github.io/DeepOnto/

[3] Note that NLI systems are typically ternary classifiers, with entailment, contradiction and neutral as the possible option. In description logic, contradiction is expressed using disjointness rules of the form $X \sqcap Y \sqsubseteq \bot$, which we verbalise as "$X$ and $Y$ implies contradiction". In this way, entailment and contradiction is predicted using the same binary classifier.

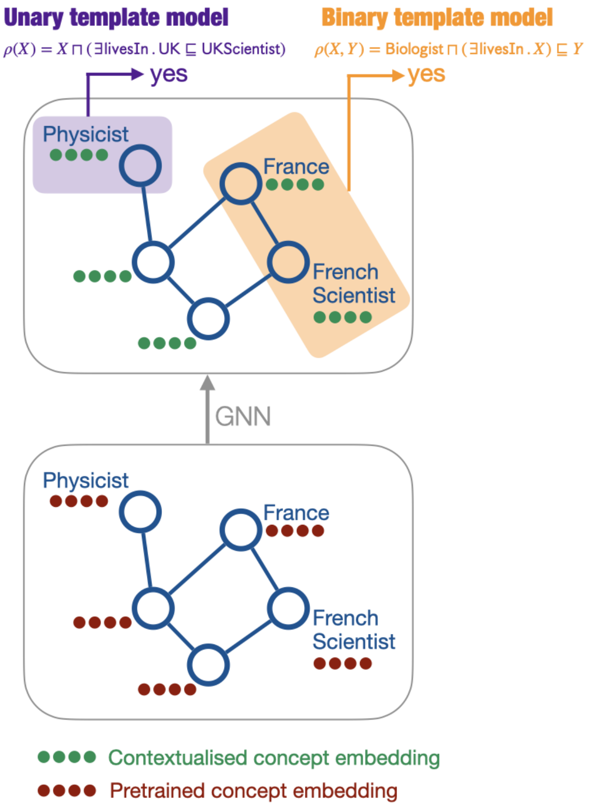
Unary template model
$\rho(X) = X \sqcap (\exists \mathsf{livesIn}\,.\,\mathsf{UK} \sqsubseteq \mathsf{UKScientist})$
yes
Binary template model
$\rho(X, Y) = \mathsf{Biologist} \sqcap (\exists \mathsf{livesIn}\,.\,X) \sqsubseteq Y$
yes
Physicist
France
French Scientist
GNN
Physicist
France
French Scientist
●●●● Contextualised concept embedding
●●●● Pretrained concept embedding

Figure 1: Schematic overview of the considered ontology completion strategies.

Rule (1) is an instance of this template with $X = \mathsf{UK}$ and $Y = \mathsf{UKScientist}$. Each binary template defines a classification problem on concept pairs $(X, Y)$, i.e. decide whether $\rho(X, Y)$ is valid.

**GNN Model** When using rule templates, the problem of ontology completion reduces to that of classifying concepts or concept pairs. While we could directly use pre-trained concept embeddings for this purpose, better results can be achieved by contextualising these embeddings using a GNN. Specifically, we consider a graph with one node for each atomic concept appearing in the training set. Two atomic concepts are connected by an edge if they appear in the same rule. We consider variants of the model with unary and binary templates. For the unary template model, we train a logistic regression classifier for each template. This classifier takes the final layer embedding of a given concept $X$ as input and predicts whether $\rho(X)$ is a valid rule. For the binary template model, we use the scoring function from DistMult (Yang et al., 2015) to predict whether $\rho(X, Y)$ is a valid rule. Li et al. (2019) trained an R-GCN to predict whether a given concept (resp. concept pair) is a valid instance of a given unary (resp. binary) template. We modify this approach in two ways. First, we simplify the model by using standard GNNs, i.e. we only consider whether two concepts appear in the same rule, rather than trying to capture the kinds of rules in which the concepts co-occur. Second, we train the model to predict the validity of a rule, rather than predicting instances of templates, which is essential to allow us to compare this approach with the NLI based strategy. Figure 1 provides a schematic overview of the unary and binary template models. Further details of the different models can be found in Appendix A.

### 4.3 Hybrid Strategies

The concept embedding based model can only predict a given rule if it is an instance of a rule template that is witnessed in the training data. When assessing a candidate rule for which no rule templates are available, rather than reverting to random guessing, it makes sense to revert to a backup ontology completion strategy instead. We consider two simple hybrid strategies based on this intuition.

**Conditional Hybrid Strategy** If the given rule $r$ is an instance of a rule template which is witnessed at least $K$ times in the training data, then we use the strategy based on concept embeddings. Otherwise, we use a fall-back strategy, such as an NLI model.

**Simple Aggregation** Given two models, we predict the rule $r$ as a positive example as soon as either of the two models labels this rule as positive.

## 5 Experiments

We now present our experimental analysis.[4]

**Models** We experiment with the NLI based models from the DeepOnto library, which correspond to fine-tuned **RoBERTa-base** and **RoBERTa-large** encoders. Regarding the fine-tuned LLMs, we experiment with **Mistral-7B**, **Llama3-8B**, **Phi3-medium** and **Gemma-7B**. Where available, we use both the base models and their instruction fine-tuned variants. The latter will be indicated by *IT* in the results tables. To fine-tune the LLMs, we rely on the 4-bit QLoRA implementation from Unsloth AI[5]. For the concept embedding based approach we consider three standard GNN architectures: **GCN** (Kipf and Welling, 2017), **GAT** (Velickovic et al.,

[4] Our dataset and implementation are available at https://github.com/thomas-bllx/CONTOR

[5] https://github.com/unslothai/unsloth

|  | Wine | Economy | Olympics | Transport | SUMO | FoodOn | GO | Average |
|---|---|---|---|---|---|---|---|---|
| NLI Based Models |  |  |  |  |  |  |  |  |
| RoBERTa-base | 57.8 | 78.5 | 76.9 | 65.6 | 76.8 | 76.2 | 72.9 | 72.1 |
| RoBERTa-large | 76.5 | 79.4 | 79.3 | 75.6 | 77.3 | 78.3 | 74.6 | 77.3 |
| Mistral-7B | 67.9 | 80.6 | **87.0** | **84.7** | 79.3 | **82.2** | 80.3 | **80.3** |
| Mistral-7B (IT) | 69.0 | 81.3 | 77.1 | 83.1 | **80.1** | 80.6 | 80.9 | 78.9 |
| Llama3-8B | 57.9 | 57.9 | 51.8 | 62.0 | 75.5 | 72.5 | 77.4 | 65.0 |
| Llama3-8B (IT) | 70.2 | 78.5 | 74.7 | 82.7 | 79.7 | 80.5 | **81.3** | 78.2 |
| Phi3-medium (IT) | 54.4 | **81.4** | 75.2 | 80.0 | 78.4 | 82.0 | 81.0 | 76.1 |
| Gemma-7B | 38.5 | 77.8 | 34.3 | 80.7 | 77.5 | 79.0 | 79.7 | 66.8 |
| Gemma-7B (IT) | 61.6 | 73.4 | 69.0 | 82.8 | 78.5 | 82.1 | 78.6 | 75.1 |
| ChatGPT | 50.8 | 66.5 | 69.5 | 56.1 | 65.6 | 60.3 | 61.7 | 61.5 |
| GPT-4 | 63.7 | 74.8 | 81.0 | 62.4 | 74.2 | 75.7 | 76.7 | 72.6 |
| Concept Embedding Based Models |  |  |  |  |  |  |  |  |
| R-GCN (UT) | 84.8 | 71.6 | 64.7 | 74.9 | 69.4 | 70.8 | 71.3 | 72.5 |
| R-GCN (BT) | 79.9 | 16.5 | 28.5 | 54.0 | 3.7 | 25.7 | 28.3 | 33.8 |
| GCN (UT) | 84.8 | 73.4 | 68.9 | 76.9 | 67.2 | 71.5 | 71.9 | 73.5 |
| GCN (BT) | 78.9 | 16.6 | 28.5 | 52.8 | 3.7 | 26.3 | 28.7 | 33.6 |
| GAT (UT) | 84.8 | 68.0 | 68.6 | 74.3 | 69.3 | 69.4 | 70.6 | 72.1 |
| GAT (BT) | 87.1 | 16.6 | 29.2 | 55.6 | 3.7 | 25.3 | 27.5 | 35.0 |
| GATv2 (UT) | **88.2** | 66.9 | 65.5 | 74.7 | 69.5 | 69.7 | 70.9 | 72.2 |
| GATv2 (BT) | 86.4 | 16.5 | 29.2 | 55.7 | 3.7 | 25.8 | 26.7 | 34.9 |
| FFN (UT) | 70.9 | 70.1 | 59.7 | 65.9 | 61.1 | 65.7 | 67.2 | 64.8 |
| FFN (BT) | 73.1 | 16.4 | 27.2 | 49.5 | 3.7 | 27.6 | 30.9 | 32.6 |

Table 2: Overview of the main results in terms of F1 (%). For the GNN models, ConCN embeddings were used as input features. The NLI based models are trained on each ontology separately.

2018) and **GATv2** (Brody et al., 2022). We furthermore compare with the model from Li et al. (2019), which uses a graph with different edge types, corresponding to the binary templates, and relies on an **R-GCN** (Schlichtkrull et al., 2018) to take these edge types into account. As a baseline, to show the benefit of using GNNs, we also compare with a variant which feeds the concept embeddings to a feedforward network instead of a GNN (shown as FFN). Unless specified otherwise, we use the ConCN concept embeddings (Li et al., 2023) as input features for the GNNs. Full experimental details can be found in Appendix B.

**Results** The main results are summarised in Table 2. For this experiment, the NLI models have been fine-tuned on each ontology separately.[6] Surprisingly, the RoBERTa-large model from DeepOnto performs better than some of the LLMs, and only slightly underperforms the best LLMs. The best NLI models also outperform the concept embedding based models on average. Among the fine-tuned LLMs, Mistral achieves the best results. In the case of Llama3 and Gemma, the instruction fine-tuned variants outperform the base models, but this effect is not observed for Mistral. ChatGPT and GPT-4 (zero-shot) perform worse than the best fine-tuned models. For the concept embedding based models, the unary templates generally perform much better than the binary templates. The different GNN architectures perform similarly, with the best overall results achieved by the GCN. The feedforward model underperforms the GNNs. Significant further analysis about the considered strategies can be found in Appendix C.

Overall, our most surprising finding is the relatively poor performance of the LLMs. While Mistral achieves the best results on average, its improvement over the much smaller RoBERTa-large model is limited. This highlights the fact that ontology completion is not a pure NLI problem. It sometimes involves dealing with artificial concepts, whose meaning is defined by the knowledge expressed in the ontology, and cannot accurately be modelled based on the given concept name. Another interesting result is the fact that GNNs with unary templates are competitive (e.g. better than Llama3-8B on average), and more importantly, that their performance is complementary with that of the LLMs. For instance, GNNs perform much bet-

[6] The other possibility is to train a single model on the joint training sets of all ontologies. An analysis of this variant can be found in the appendix.

ter on Wine but much worse on GO.

**Hybrid Strategies** The results of the hybrid models are shown in Table 3. For the conditional hybrid models, we use the notation $X + Y$ to denote the configuration where $X$ is the base model and $Y$ is the fall-back model. We write GCN (UT$_{\geq 3}$) for the variant where $K = 3$, i.e. where only templates occurring at least 3 times in the training data are considered, while GCN (UT) corresponds to the standard variant where $K = 1$. For the simple aggregation strategy, $X + Y$ refers to the model that predicts a rule as positive if it is labelled as such by $X$ or $Y$. Several of the hybrid configurations outperform the best individual models. The conditional hybrid approach slightly underperforms the simple aggregation approach in most cases. We can also see that the variants with $K = 3$ typically outperform the corresponding variants with $K = 1$. Among the LLMs, configurations with Mistral lead to be best results. In Appendix C we show that conditional models can be slightly improved by also considering binary templates.

Our main finding is the surprising effectiveness of the hybrid models, and the simple aggregation strategy in particular. Remarkably, combining the GCN with Mistral even leads to much improved results on ontologies where the GCN alone performs poorly (e.g. Economy and Olympics). Hybrid configurations with RoBERTa are somewhat less effective, especially for the four smaller ontologies, which suggests that the strengths of this model are somewhat in between those of the GCN (i.e. capturing the structure of the ontology) and the LLMs (i.e. capturing richer pre-trained knowledge about lexical entailment). The relatively poor performance of the Mistral + Llama configuration further supports the claim that the success of the hybrid approach (for the other configurations) is due to the complementarity of the methods involved.

**Qualitative Analysis** Below are examples of rules that were identified by the GCN with unary templates but predicted by neither Llama3-8B (IT) nor Mistral-7B:

- Chenin Blanc implies something that has flavor Moderate
- Avocado implies Grocery Produce
- Smoked and Frozen Cod Fillet implies Cod Fillet
- Rings implies Artistic Gymnastics
- Abort implies Computer Process
- Food Distribution Operation implies Military Operation
- Petite Syrah implies something that has sugar Dry
- Railroad Track and Bulkhead implies Contradiction

We see several cases involving domain-specific concepts (e.g. *Rings implies Artistic Gymnastics*), which only make sense within the context of the given ontology (i.e. Olympics). We also see cases involving $\exists$, which often sound less natural when verbalised, e.g. *Petite Syrah implies something that has sugar Dry*. Conversely, let us now consider the following examples of rules that were identified by both Llama3-8B (IT) and Mistral-7B but not by the GCN with unary templates.

- Sauternes implies something located in Sauterne Region
- Chianti implies something located in Chianti Region
- Fire Boat implies Emergency Vehicle
- Canal System implies Water Transportation System
- War implies Violent Contest
- Telegraph implies Electric Device
- Artistic Gymnastics implies Gymnastics
- Summer Games implies Olympic Games
- Plastic implies Manufactured Product
- Coffee Bean implies Plant Agricultural Product

Here we see many examples that are almost tautological, e.g. *Sauternes implies something located in Sauterne Region*. Such rules are easy to identify by NLI models, but the GNN models can fail on such cases if they lack the required template. NLI models also do well on examples that benefit from the general background knowledge captured by LLMs, e.g. *Fire Boat implies Emergency Vehicle*.

|  | **Wine** | **Economy** | **Olympics** | **Transport** | **SUMO** | **FoodOn** | **GO** | **Average** |
|---|---|---|---|---|---|---|---|---|
| Conditional Hybrid Models |  |  |  |  |  |  |  |  |
| GCN (UT) + RoBERTa | 79.9 | 74.8 | 71.0 | 64.8 | 67.6 | 77.1 | 76.4 | 78.5 |
| GCN (UT) + Mistral | 89.5 | 76.6 | 73.0 | 66.7 | 68.0 | 78.9 | 78.6 | 80.1 |
| GCN (UT) + Llama | 83.3 | 74.8 | 75.0 | 64.8 | 66.8 | 78.1 | 77.5 | 79.2 |
| GCN (UT$_{\geq 3}$) + RoBERTa | 83.9 | 85.4 | 80.6 | 81.3 | 81.7 | 79.7 | 77.4 | 81.4 |
| GCN (UT$_{\geq 3}$) + Mistral | **100.0** | 88.3 | 87.9 | 86.3 | 83.2 | 82.1 | 73.7 | 85.9 |
| GCN (UT$_{\geq 3}$) + Llama | 80.0 | 85.4 | 84.4 | 79.7 | 78.1 | 72.2 | 79.2 | 79.9 |
| Simple Aggregation |  |  |  |  |  |  |  |  |
| RoBERTa + Mistral | 79.1 | 87.8 | 87.3 | 79.9 | 83.9 | 86.8 | 87.4 | 84.6 |
| RoBERTa + Llama | 84.8 | 86.2 | 87.5 | 74.8 | **89.6** | **91.2** | **92.1** | 86.6 |
| Mistral + Llama | 72.7 | 72.7 | 70.5 | 67.6 | 71.4 | 67.4 | 67.4 | 70.0 |
| GCN (UT) + RoBERTa | 71.4 | 76.7 | 77.8 | 74.3 | 72.5 | 80.2 | 77.5 | 80.4 |
| GCN (UT) + Mistral | 89.5 | 81.7 | 75.0 | 72.8 | 71.9 | 81.6 | 79.9 | 83.1 |
| GCN (UT) + Llama | 83.3 | 76.6 | 76.9 | 69.4 | 69.4 | 80.9 | 78.8 | 81.2 |
| GCN (UT$_{\geq 3}$) + RoBERTa | 65.0 | 83.8 | 76.5 | 78.0 | 76.4 | 77.4 | 73.1 | 75.7 |
| GCN (UT$_{\geq 3}$) + Mistral | **100.0** | **91.6** | **89.6** | **87.7** | 88.9 | 88.0 | 76.2 | **88.9** |
| GCN (UT$_{\geq 3}$) + Llama | 80.0 | 87.2 | 84.4 | 80.9 | 82.2 | 75.4 | 81.3 | 81.6 |

Table 3: Overview of the main results in terms of F1 (%) for the hybrid models. *RoBERTa* refers to RoBERTa-large, *Mistral* refers to the Mistral-7B base model, and *Llama* refers to the instruction-finetuned Llama3-8B model.

## 6 Related Work

**Knowledge Graph Completion** KG completion is concerned with predicting missing factual assertions and thus fundamentally different from ontology completion. Note, however, that several rule based methods have been proposed for KG completion (Meilicke et al., 2019; Qu et al., 2021; Cheng et al., 2023). Such methods learn rules that capture statistical regularities from the KG (e.g. if somebody works in a company that is based in country X, then they are likely to be a resident of that country). This is again different from our focus in this paper: KG completion rules are typically not universally valid and tied to a particular KG, and they are learned from factual assertions (i.e. the training KG). In contrast, we aim to learn ontological rules that can be used across a wide range of applications, and we infer the plausibility of these rules based on the meaning of the concepts involved.

**Taxonomy Expansion** Another popular knowledge base completion task consists in expanding taxonomies (Jurgens and Pilehvar, 2016; Takeoka et al., 2021), which differs from ontology completion in several respects. First, in the case of ontology completion, due to the presence of complex rules, we need to go beyond modelling relations between natural language terms. Second, most taxonomy expansion benchmarks focus on adding new concepts to the taxonomy, whereas we focus on finding missing rules (involving concepts that already belong to the ontology). Third, in the case of taxonomy enrichment, the input usually consists of a term and a definition, whereas in the case of ontology completion, we need to infer the intended meaning of a concept from the ontology itself. Taxonomy enrichment is thus closely aligned with tasks such as hypernym detection (Hanna and Mareček, 2021) and definition modelling (Noraset et al., 2017). As such, most current approaches primarily rely on language models (Chen et al., 2021a; Takeoka et al., 2021), although Graph Neural Networks have also been leveraged in this context (Shen et al., 2020; Shang et al., 2020).

**Ontology Learning** The ontology completion methods we considered in this paper use NLP models (i.e. concept embeddings or language models) to provide prior knowledge about the meaning of the concepts. Some approaches have been studied which only focus on the structure of the ontology, taking inspiration from knowledge graph embedding models (Kulmanov et al., 2019; Mondal et al., 2021; Xiong et al., 2022; Peng et al., 2022; Jackermeier et al., 2023), but this requires very large ontologies with sufficient regularity. The special case of embedding taxonomies has also received extensive interest (Vilnis et al., 2018; Nickel and Kiela, 2017; Ganea et al., 2018; Le et al., 2019). Yet another line of work has focused on learning concept representations using word embedding models (Mikolov et al., 2013), where the key idea is to view ontology axioms as sentences (Smaili et al., 2018, 2019; Chen et al., 2021b). Finally, when a sufficiently large set of factual assertions is available (i.e. an ABox), we can also find plausible ontology rules by relying on standard rule learning (Iannone et al., 2007; Fanizzi et al., 2008; Bühmann et al., 2016; Sarker and Hitzler, 2019).

**Benchmarking Ontology Completion** Currently there are no standard benchmarks for ontology completion, unlike for the task of KG completion where standardised benchmarks have long been the norm. Chen et al. (2023) and Li et al. (2019) evaluated their models by checking whether they are able to distinguish held-out rules from the ontology from corrupted versions of these rules. However, their evaluation protocols are not compatible and they were tested on different ontologies. Moreover, as already mentioned, testing on randomly corrupted rules has important limitations. He et al. (2023c) introduced ONTOLAMA, a benchmark for testing the ability of language models to recognise subsumption relations between complex concepts. This differs from the benchmark we introduce in this paper, as ONTOLAMA involves predicting the validity of a single rule, without any further ontology context. In contrast, we are specifically interested in methods that can take a given ontology into account.

**Modelling Ontologies with LLMs** Language models are now commonly used for KG completion. Somewhat surprisingly, perhaps, LLMs have not previously been considered for ontology completion, to the best of our knowledge. However, He et al. (2023a) carried out a preliminary study into the potential of LLMs for the related problem of ontology alignment, i.e. mapping the concepts from one ontology onto the corresponding concepts from another ontology. They obtained mixed results with Flan-T5-XXL and ChatGPT, with both models failing to consistently outperform a fine-tuned BERT method (in a zero-shot setting). Giglou et al. (2024) obtained somewhat promising results with Llama 2, Mistral and ChatGPT, but also failed to consistently outperform traditional methods for ontology alignment. Some authors have proposed methods for learning ontologies with LLMs (Giglou et al., 2023; Kommineni et al., 2024), but such works focus on relation extraction tasks (e.g. finding instances of a concept, or relations between concepts) rather than on modelling rules. A key challenge for ontology completion comes from the fact that concepts may be used with a meaning that is more specific than the common understanding of the corresponding natural language term. LLMs with in-context learning tend to struggle in such specification-heavy settings (Peng et al., 2023), which is why we have focused on fine-tuned LLMs in our analysis.

## 7 Conclusions

We have considered the problem of finding plausible rules which are missing from a given ontology. Specifically, we have introduced a new benchmark with hard negatives, which were manually verified by human annotators. We then compared, for the first time, the two main families of ontology completion methods: NLI based methods and GNN based methods. Beyond existing NLI based methods, we also presented the first analysis of LLMs for ontology completion. Finally, we found hybrid strategies to achieve surprisingly strong results, clearly outperforming the current state-of-the-art.

**Acknowledgments** This work was supported by EPSRC grant EP/V025961/1, Leverhulme Trust grant RPG-2021-140, ANR-22-CE23-0002 ERIANA, ANR-20-CHIA-0028 and HPC resources from GENCI-IDRIS (Grant 2024-[AD011013338R2]).

## Limitations

The area of ontology completion is considerably less mature than related areas such as taxonomy expansion and knowledge graph completion. As such, the methods we have analysed in this paper should be seen as baselines for future work. For instance, we expect that much better hybrid strategies can be developed, which combine the knowledge captured by LLMs with models that take into account the structure of the ontology. The results of the LLM models themselves should also be seen as lower bounds. For instance, while we have attempted to construct reasonable prompts, it is likely that better prompting strategies can be found.

In this paper, we have treated ontology completion as a binary classification problem, deciding whether a given candidate rule is valid or not. However, in practice, we also need a mechanism for generating suitable candidate rules. The template based approach can be used in a straightforward way for this purpose. While it is likely that LLMs can also be successfully leveraged for generating rules, rather than classifying them, studying how this can best be done is left for future work.

## References

Dean Allemang and Juan Sequeda. 2024. Increasing the llm accuracy for question answering: Ontologies to the rescue! *arXiv preprint arXiv:2405.11706*.

Franz Baader, Ian Horrocks, and Ulrike Sattler. 2004. Description logics. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 3–28. Springer.

Franz Baader, Ian Horrocks, and Ulrike Sattler. 2009. Description logics. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 21–43. Springer.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Shaked Brody, Uri Alon, and Eran Yahav. 2022. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. 2016. Dl-learner - A framework for inductive learning on the semantic web. *J. Web Semant.*, 39:15–24.

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.

Catherine Chen, Kevin Lin, and Dan Klein. 2021a. Constructing taxonomies from pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4687–4700, Online. Association for Computational Linguistics.

Jiaoyan Chen, Yuan He, Yuxia Geng, Ernesto Jiménez-Ruiz, Hang Dong, and Ian Horrocks. 2023. Contextual semantic embeddings for ontology subsumption prediction. *World Wide Web (WWW)*, 26(5):2569–2591.

Jiaoyan Chen, Pan Hu, Ernesto Jiménez-Ruiz, Ole Magnus Holter, Denvar Antonyrajah, and Ian Horrocks. 2021b. Owl2vec*: embedding of OWL ontologies. *Mach. Learn.*, 110(7):1813–1845.

Kewei Cheng, Nesreen K. Ahmed, and Yizhou Sun. 2023. Neural compositional rule learning for knowledge graph reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. 2008. DL-FOIL concept learning in description logics. In *Inductive Logic Programming, 18th International Conference, ILP 2008, Prague, Czech Republic, September 10-12, 2008, Proceedings*, volume 5194 of *Lecture Notes in Computer Science*, pages 107–121. Springer.

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.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1632–1641. PMLR.

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. 2023. Llms4ol: Large language models for ontology learning. In *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part I*, volume 14265 of *Lecture Notes in Computer Science*, pages 408–427. Springer.

Hamed Babaei Giglou, Jennifer D'Souza, Felix Engel, and Sören Auer. 2024. Llms4om: Matching ontologies with large language models. *CoRR*, abs/2404.10317.

Michael Hanna and David Mareček. 2021. Analyzing BERT's knowledge of hypernymy via prompting. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuan He, Jiaoyan Chen, Hang Dong, and Ian Horrocks. 2023a. Exploring large language models for ontology alignment. *CoRR*, abs/2309.07172.

Yuan He, Jiaoyan Chen, Hang Dong, Ian Horrocks, Carlo Allocca, Taehun Kim, and Brahmananda Sapkota. 2023b. Deeponto: A python package for ontology engineering with deep learning. *CoRR*, abs/2307.03067.

Yuan He, Jiaoyan Chen, Ernesto Jiménez-Ruiz, Hang Dong, and Ian Horrocks. 2023c. Language model analysis for ontology subsumption inference. *CoRR*, abs/2302.06761.

Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. 2007. An algorithm based on counterfactuals for concept learning in the semantic web. *Appl. Intell.*, 26(2):139–159.

Mirjana Ivanović and Zoran Budimac. 2014. An overview of ontologies and data resources in medical domains. *Expert Systems with Applications*, 41(11):5158–5166.

Mathias Jackermeier, Jiaoyan Chen, and Ian Horrocks. 2023. Box2el: Concept and role box embeddings for the description logic EL++. *CoRR*, abs/2301.11118.

Nitisha Jain, Trung-Kien Tran, Mohamed H. Gad-Elrab, and Daria Stepanova. 2021. Improving knowledge graph embeddings with ontological reasoning. In *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings*, volume 12922 of *Lecture Notes in Computer Science*, pages 410–426. Springer.

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.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. 2024. From human experts to machines: An LLM supported approach to ontology and knowledge graph construction. *CoRR*, abs/2403.08345.

Maxat Kulmanov, Mohammed Asif Khan, and Robert Hoehndorf. 2018. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668.

Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. 2019. EL embeddings: Geometric construction of models for the description logic EL++. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6103–6109. ijcai.org.

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3231–3241, Florence, Italy. Association for Computational Linguistics.

Na Li, Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, Qing Gu, and Steven Schockaert. 2021. Modelling general properties of nouns by selectively averaging contextualised embeddings. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3850–3856. ijcai.org.

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, Hanane Kteich, Zied Bouraoui, and Steven Schockaert. 2023. 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.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021a. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021b. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 562–574, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3137–3143. ijcai.org.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Sutapa Mondal, Sumit Bhatia, and Raghava Mutharaju. 2021. Emel++: Embeddings for EL++ description logic. In *Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, California, USA, March 22-24, 2021*, volume 2846 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Ana Ozaki. 2020. Learning description logic ontologies: Five approaches. where do they stand? *Künstliche Intell.*, 34(3):317–327.

Hao Peng, Xiaozhi Wang, Jianhui Chen, Weikai Li, Yunjia Qi, Zimu Wang, Zhili Wu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2023. When does in-context learning fall short and why? A study on specification-heavy tasks. *CoRR*, abs/2311.08993.

Xi Peng, Zhenwei Tang, Maxat Kulmanov, Kexin Niu, and Robert Hoehndorf. 2022. Description logic EL++ embeddings with intersectional closure. *CoRR*, abs/2202.14018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Meng Qu, Junkun Chen, Louis-Pascal A. C. Xhonneux, Yoshua Bengio, and Jian Tang. 2021. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Md. Kamruzzaman Sarker and Pascal Hitzler. 2019. Efficient concept induction for description logics. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3036–3043. AAAI Press.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Nadine Schuurman. 2006. Formalization matters: Critical gis and ontology research. *Annals of the Association of American Geographers*, 96(4):726–739.

Chao Shang, Sarthak Dash, Md. Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, and Alfio Gliozzo. 2020. Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2198–2208, Online. 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 *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 486–497. ACM / IW3C2.

Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. 2018. Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinform.*, 34(13):i52–i60.

Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. 2019. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinform.*, 35(12):2133–2140.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Bilal Succar. 2009. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in construction*, 18(3):357–375.

Kunihiro Takeoka, Kosuke Akimoto, and Masafumi Oyamada. 2021. Low-resource taxonomy enrichment with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2747–2758, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272, Melbourne, Australia. Association for Computational Linguistics.

Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. 2022. Faithful embeddings for EL++ knowledge bases. In *The Semantic Web - ISWC 2022 - 21st International Semantic Web*

*Conference, Virtual Event, October 23-27, 2022, Proceedings*, volume 13489 of *Lecture Notes in Computer Science*, pages 22–38. Springer.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

# A Details of Ontology Completion Strategies

In this section, we provide some additional details about the considered ontology completion strategies.

## A.1 NLI Based Approach with LLMs

To fine-tune the LLMs, we use the following prompt:

> *Classify the text into True or False. Reply with only one word: True or False. Determine if the following statement is valid:* [RULE BODY] *implies* [RULE HEAD].

where [RULE BODY] and [RULE HEAD] are the verbalisations of the rule body (i.e. left-hand side) and head, obtained with DeepOnto. For instance, for the rule $\mathsf{CheninBlanc} \sqsubseteq \exists \mathsf{hasFlavor.Moderate}$, the last part of the prompt becomes: *Chenin Blanc implies something that has flavor Moderate*. For our zero-shot experiments with ChatGPT and GPT-4, we used the same prompt.

## A.2 Concept Embedding Approach

**Typed Binary Templates** Many of the rules in a typical ontology are basic subsumptions of the form $A \sqsubseteq B$. With a naive application of the binary template model, such rules give us the trivial binary template $\rho(X,Y) = X \sqsubseteq Y$. Following Li et al. (2019), we therefore use *typed* binary templates. When typed templates are used, instead of replacing a concept $A$ by a placeholder $X$, we replace it by the conjunction $X \sqcap A'$ where $A'$ is a direct parent of $A$ (i.e. $A'$ is such that we have the rule $A \sqsubseteq A'$ in the ontology). A basic subsumption $A \sqsubseteq B$ then leads to a binary template of the form $\rho(X,Y) = X \sqcap A' \sqsubseteq Y \sqcap B'$, where $A'$ and $B'$ are direct parents of $A$ and $B$. If $A$ and $B$ have multiple direct parents, then we consider each of the corresponding typed templates.

**Using Multi-relational Graphs** To construct the concept graph that is used by the GNN models, the strategy explained in the main paper is to connect two concepts with an edge if they co-occur in some rule. In contrast, Li et al. (2019) used edges of different types, with the types corresponding to binary templates. In other words, their graph structure reflects which kinds of rules two concepts co-occur in. While their graph is more informative, learning with multi-relational graphs is harder, especially considering that the amount of training data that we have available is typically limited. For this reason, it turns out that using our simpler approach performs as well in practice, while having the advantage of being more efficient.

**Loss Function** let $\mathcal{U}$ be the set of unary templates which are witnessed in the given ontology. For each atomic concept $X$ and template $\rho \in \mathcal{U}$, we estimate the probability that $\rho(X)$ is a valid rule as follows

$$\textit{conf}(\rho, X) = \sigma(\mathbf{x} \cdot \mathbf{a}_\rho + b_\rho) \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the final-layer embedding of concept $X$ in the GNN, $\mathbf{a}_\rho \in \mathbb{R}^n$ and $b_\rho \in \mathbb{R}$, and $\sigma$ denotes the sigmoid activation function.

The evaluation carried out by Li et al. (2019) focused on predicting whether a concept $X$ (resp. concept pair $X, Y$) is a valid instance of a unary (resp. binary) template. This makes it possible to compare the performance of different concept embedding strategies, but not to compare the GNN model with other strategies for ontology completion. We therefore adapt the model to make predictions at the level of rules. Specifically, to classify a given rule $r$ as valid or not, we first determine all templates $\rho$ and concepts $X$ for which $r = \rho(X)$. Let us write $\rho_1(X_1), ..., \rho_m(X_m)$ for these template-concept combinations. The probability that $r$ is a valid rule is then estimated as:

$$p(r) = \max_{i=1}^{m} \textit{conf}(\rho_i, X_i)$$

Note that if $m = 0$, i.e. $r$ is not an instance of any of the unary templates, then $p(r) = 0$. We train the model using binary cross-entropy:

$$\mathcal{L} = -\sum_r y_r \log p(r) + (1 - y_r) \log(1 - p(r))$$

where the summation ranges over the rules $r$ in the training set (see Section 3), and we define $y_r = 1$ if $r$ is a positive example and $y_r = 0$ otherwise.

| Model Name | URL |
|---|---|
| Mistral-7B | https://huggingface.co/unsloth/mistral-7b-v0.3-bnb-4bit |
| Mistral-7B (IT) | https://huggingface.co/unsloth/mistral-7b-instruct-v0.3-bnb-4bit |
| Llama3-8B | https://huggingface.co/unsloth/llama-3-8b-bnb-4bit |
| Llama3-8B (IT) | https://huggingface.co/unsloth/llama-3-8b-Instruct-bnb-4bit |
| Phi3-medium (IT) | https://huggingface.co/unsloth/Phi-3-medium-4k-instruct-bnb-4bit |
| Gemma-7B | https://huggingface.co/unsloth/gemma-7b-bnb-4bit |
| Gemma-7B (IT) | https://huggingface.co/unsloth/gemma-7b-it-bnb-4bit |
| Llama2-7B | https://huggingface.co/meta-llama/Llama-2-7b-hf |
| Llama2-7B-Chat | https://huggingface.co/meta-llama/Llama-2-7b-chat-hf |
| Llama2-13B | https://huggingface.co/meta-llama/Llama-2-13b-hf |
| Llama2-13B-Chat | https://huggingface.co/meta-llama/Llama-2-13b-chat-hf |
| Llama2-7B-32K-Instruct | https://huggingface.co/togethercomputer/Llama-2-7B-32K-Instruct |
| Vicuna-13B | https://huggingface.co/lmsys/vicuna-13b-v1.5 |

Table 4: Specification of the LLMs that were used in our experiments.

| Ontology Name | URL |
|---|---|
| Wine | https://www.w3.org/TR/2003/PR-owl-guide-20031215/wine |
| Economy | http://reliant.teknowledge.com/DAML/Economy.owl |
| Olympics | https://swat.cse.lehigh.edu/resources/onto/olympics.owl |
| Transport | http://reliant.teknowledge.com/DAML/Transportation.owl |
| SUMO | https://www.ontologyportal.org/ |
| FoodOn | https://obofoundry.org/ontology/foodon.html |
| GO | http://purl.obolibrary.org/obo/go.owl |

Table 5: Specification of the ontologies that were used for the CONTOR benchmark.

For the binary template model, we need to predict whether $\rho(X, Y)$ is a valid rule. We considered two possible approaches for this. First, we considered the scoring function from DistMult (Yang et al., 2015): In particular, we have:

$$\textit{conf}(\rho, X, Y) = \sigma(\mathbf{x}^T \mathbf{M}_\rho \mathbf{y}) \qquad (6)$$

where $\mathbf{M}_\rho \in \mathbb{R}^{n\times n}$ is a diagonal matrix, and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ are the final-layer embeddings of concepts $X$ and $Y$ in the GNN. Second, we also considered a scoring function inspired by TransE (Bordes et al., 2013):

$$\textit{conf}(\rho, X, Y) = \sigma(\|(\mathbf{y} - \mathbf{x} - \mathbf{a}_\rho\|_2 - b_\rho) \qquad (7)$$

with $\mathbf{a}_\rho \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Unless specified otherwise, the results in this paper are based on the DistMult variant. The binary template model is also trained using binary cross-entropy.

## A.3 Hybrid Strategies

**Conditional Hybrid Strategy** For the ease of presentation, we consider the unary template model for explaining the hybrid strategy. Let $r$ be a given candidate rule and let $\rho_1, ..., \rho_m$ be the corresponding rule templates, where $r = \rho_1(X_1) = ... = \rho_m(X_m)$. For a rule template $\rho$, let us write *freq*$(\rho)$ for the number of times it is witnessed in the training data (i.e. the number of available rules of the form $\rho(X)$). We define the support of $r$ as:

$$\textit{supp}(r) = \max(\textit{freq}(\rho_1), ..., \textit{freq}(\rho_m))$$

Note in particular that *supp*$(r) = 0$ if no templates are available that match $r$. To classify a given rule candidate $r$, we then proceed as follows:

- If *supp*$(r) \geq K$ then we predict the validity of $r$ using the available rule templates
- Otherwise, we use a fall-back strategy, such as an NLI model, to make the prediction.

Entirely analogously we can also consider a hybrid strategy based on the binary template model.

**Simple Aggregation** For the simple aggregation strategy, we consider two ontology completion models (or in some configurations three models) and simply classify a rule as a positive example if it is labelled as such by at least one of the models. Note that this approach, by design, increases recall but possibly at the cost of reduced precision. As such, this approach may perform best if at least one of the models is precision-oriented. For this reason, when the concept embedding based method is used, we also consider the aforementioned variant in which only templates appearing $K$ times in the training data are used.

## B Experimental Details

**Language Models and Ontologies** Table 4 gives an overview of the language models that were used in our experiments, together with information about where they can be obtained. In addition to the models that were considered in the main results table (Table 2), we show details of some models which we will additionally consider in this appendix. Details about the ontologies that are included in our CONTOR benchmark are provided in Table 5. Note that these ontologies are provided under a CC BY license.

**Generating Training Examples** As explained in the main paper, the negative examples in the training data were obtained by randomly corrupting the positive rules from the training split. To this end, we used the following strategies, where we use notations such as $\alpha$ and $\beta$ to denote arbitrary rule bodies and heads, and notations such as $C$ and $D$ to denote concept names: (i) For each rule of the form $C \sqsubseteq D$ in the ontology, we add $D \sqsubseteq C$ as a negative rule. (ii) For each rule of the form $\alpha_1 \sqsubseteq \beta_1$, we randomly select another rule from the ontology of the form $\alpha_2 \sqsubseteq \beta_2$ and we generate the corrupted rules $\alpha_1 \sqsubseteq \beta_2$ and $\alpha_2 \sqsubseteq \beta_1$. (iii) For each rule of the form $C \sqsubseteq D$, we randomly replace $C$ or $D$ by another concept, which is randomly sampled from all concepts appearing in the ontology. (iv) For each rule of the form $C \sqsubseteq D$ we generate the constraint $C \sqcap D \sqsubseteq \bot$ (encoding that $C$ and $D$ are disjoint).

**Training Details** We use the rule-based verbalizer provided by the DeepOnto library to convert the rules into textual inputs. For instance, the concept RedWine is converted to the term "red wine", while the concept ∃hasColor.Red is converted to the phrase "something that has color red". For training with DeepOnto, we set the learning rate to 1e-5, weight decay to 1e-2, the number of epochs to 3, the batch size of the training and development sets to 8, and the batch size of test sets to 16.

For tuning the GNN models, we select the number of layers from {2, 3, 4, 5}. For GAT and GATv2, we select the number of attention heads from {4, 8, 16} and fix the negative slope of the LeakyReLU activations as 0.2. In all GNN models, we use dropout to avoid over-fitting. For GAT and GATv2, the dropout rate of attention layers is set to 0.2. For all GNN models, the dropout rate of non-attention layers is set to 0.5. We select the dimension of the hidden layers from {8, 16, 32, 64}. All GNN models are optimised using AdamW, with a learning rate of 1e-2 and weight decay of 5e-2. We train the models for 200 epochs and select the best checkpoint based on the validation split.

For the baselines based on feedforward networks (FFN), we learn encoders consisting of several ReLU layers to transform the pre-trained concept embeddings. On top of the encoder, we add a classification layer of the form (5) for the unary template variant and a classification layer of the form (6) for the binary template variant. The number of layers is tuned from {3, 4, 5} for FoodOn and GO, and {2, 3, 4} for the other ontologies. The dimensionality of the hidden states is tuned from {32, 64, 128} for FoodOn and GO, and {16, 32, 64} for the others. We use AdamW as the optimizer with a learning rate of 0.01. We use dropout, and the dropout rate is set to 0.5. As for the GNNs, we use binary cross entropy as for training the models.

To fine-tune the LLMs, we rely on QLoRA, which combines 4-bit quantization via BitsAndBytes with Low-Rank Adaptation (LoRA) to enable efficient model optimization. In particular, we use the implementation from Unsloth AI[7], which has a number of further optimisations for efficient memory management.

## C Additional Analysis

**Comparison of Concept Embeddings** For the GNN models, Table 2 only reports results for the ConCN concept embeddings. To analyse the impact of this choice, Table 6 compares the results for different concept embedding choices. We experiment with several types of pre-trained concept embeddings. First, we consider standard word embeddings models: Skip-gram (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and Numberbatch (Speer et al., 2017)[8]. Second, we consider methods which rely on fine-tuned LM encoders to map the name of a concept to their embedding: Mirror-

[7] https://github.com/unslothai/unsloth

[8] Glove trained on Common Crawl (https://nlp.stanford.edu/projects/glove/), Skip-gram trained on Google News (https://code.google.com/archive/p/word2vec/), and Numberbatch from https://conceptnet.s3.amazonaws.com/downloads/2019/numberbatch/numberbatch-en-19.08.txt.gz

BERT[9] (Liu et al., 2021a), which was trained in a self-supervised fashion, and the bi-encoder model from Gajbhiye et al. (2022), which we refer to as BiEnc[10]. Finally, we use two methods which obtain embeddings by finding mentions of the concept name in Wikipedia and aggregating their contextualised representation: MirrorWiC[11] (Liu et al., 2021b), which is self-supervised, and ConCN[12](Li et al., 2023), which was trained using distant supervision from ConceptNet. For this analysis, we have used the GCN model.

As can be seen, the results are highly sensitive to the quality of the concept embeddings, with traditional word embeddings models such as Skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) leading to considerably weaker results. However, the performance of Numberbatch (Speer et al., 2017), a method which uses the ConceptNet[13] knowledge graph to improve Skip-gram, is surprisingly strong. MirrorBERT (Liu et al., 2021a), MirrorWiC (Liu et al., 2021b), BiEnc (Gajbhiye et al., 2022) and ConCN (Li et al., 2023) all rely on fine-tuned BERT models for obtaining concept embeddings. The best results are obtained with ConCN, which also relies on ConceptNet for distant supervision.

**Additional Language Models** Table 7 provides results for a number of LLMs that were not considered in the main experiments. These models clearly underperform the best models from Table 2.

**Usefulness of Binary Templates** While the binary template model underperforms the other approaches when used in isolation, it has complementary strengths that can be exploited by the hybrid strategies. Table 8 shows an analysis of hybrid models that take advantage of this. For the conditional hybrid models, the notation $X+Y+Z$ refers to a configuration where $X$ is the base model and the conditional hybrid model $Y+Z$ is the fall-back model. As can be seen, in most cases adding the binary template model leads to slightly better results, compared to the corresponding configurations in Table 3.

**Joint Training** In the main experiments (Table 2), the NLI based models were trained on each ontology separately. This has the advantage that the resulting models are specialised towards the given ontology, which can be important if ontologies use concepts in idiosyncratic ways, among others. However, jointly training these models on all ontologies together also has some possible advantages. For instance, some of the smaller ontologies may not have enough examples to enable successful fine-tuning of LLMs. Moreover, the models might generalise better by being exposed to a more diverse set of training examples. Table 9 shows the results we obtained with this joint training strategy. We can see that this generally leads to worse results than fine-tuning on individual ontologies. However, some configurations benefit from this joint training strategy, namely Gemma-7B and (to a lesser extent) Llama3-8B, as well as several variants of Llama2. Interestingly, joint fine-tuning only benefits the base models of Gemma and Llama3. We can also observe some differences between the different ontologies. For instance, for SUMO it is almost always beneficial to fine-tune on this ontology alone, with the Llama2-7B variants as the only exceptions.

**Scoring Function** For the binary template model, in our main experiments we have relied on a bilinear scoring function (6). Another possibility, inspired by TransE, is to use (7) instead. A comparison between both alternatives is shown in Table 10. As can be seen, the results are broadly similar.

**Effect of QLoRA** The use of 4-bit quantization and low-rank adaption makes the process of fine-tuning LLMs significantly more efficient. However, this efficiency may come at the cost of reduced performance. To analyse this, Table 11 compares the results for Llama-3 based on QLoRA (from the main experiments) with a model that was trained using standard fine-tuning in full precision. For the full fine-tuning experiment, we relied on the Llama-3 model provided by Meta[14]. As can be seen, full fine-tuning performs slightly better on average, but the improvement is insufficient to justify the significantly higher computational cost, especially since other models perform better even with 4-bit QLoRA (including the instruction fine-tuned variant of Llama-3). In particular, for the full fine-tuning experiment, we needed eight A100 GPUs

[9] https://huggingface.co/cambridgeltl/mirror-bert-base-uncased-word

[10] https://github.com/amitgajbhiye/biencoder_concept_property

[11] https://huggingface.co/cambridgeltl/mirrorwic-bert-base-uncased

[12] https://github.com/lina-luck/semantic_concept_embeddings

[13] https://conceptnet.io

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

|  |  | **Wine** | **Economy** | **Olympics** | **Transport** | **SUMO** | **FoodOn** | **GO** | **Average** |
|---|---|---|---|---|---|---|---|---|---|
| **Unary templates** | Skip-gram | 50.3 | 52.2 | 47.7 | 48.3 | 51.6 | 56.4 | 55.1 | 51.7 |
|  | GloVe | 51.2 | 53.4 | 48.9 | 50.2 | 53.7 | 58.7 | 57.6 | 53.4 |
|  | Numberbatch | 82.8 | 72.3 | 67.5 | 75.2 | 66.9 | 70.8 | 70.5 | 72.3 |
|  | MirrorBERT | 81.3 | 71.5 | 65.4 | 70.3 | 63.4 | 68.1 | 67.1 | 69.6 |
|  | MirrorWiC | 82.4 | 71.9 | 66.2 | 71.6 | 64.5 | 68.8 | 68.3 | 70.5 |
|  | BiEnc | 83.2 | 72.9 | 68.2 | 75.4 | 66.8 | 71.2 | 70.9 | 72.7 |
|  | ConCN | 84.8 | 73.4 | 68.9 | 76.9 | 67.2 | 71.5 | 71.9 | 73.5 |
| **Binary templates** | Skip-gram | 47.8 | 13.1 | 19.9 | 42.8 | 2.6 | 18.7 | 18.3 | 23.3 |
|  | GloVe | 49.1 | 13.2 | 20.4 | 43.2 | 2.6 | 19.2 | 18.9 | 23.8 |
|  | Numberbatch | 77.3 | 16.1 | 27.9 | 51.6 | 3.7 | 25.9 | 27.9 | 32.9 |
|  | MirrorBERT | 74.5 | 15.7 | 26.4 | 50.3 | 3.6 | 23.7 | 24.8 | 31.3 |
|  | MirrorWiC | 75.2 | 15.9 | 26.5 | 50.9 | 3.6 | 24.5 | 25.4 | 31.7 |
|  | BiEnc | 76.3 | 16.2 | 28.1 | 52.6 | 3.7 | 25.8 | 28.1 | 33.0 |
|  | ConCN | 78.9 | 16.6 | 28.5 | 52.8 | 3.7 | 26.3 | 28.7 | 33.6 |

Table 6: Analysis of different pre-trained concept embeddings. All results are obtained with the GCN model.

|  | **Wine** | **Economy** | **Olympics** | **Transport** | **SUMO** | **FoodOn** | **GO** | **Average** |
|---|---|---|---|---|---|---|---|---|
| Llama2-7B | 45.0 | 63.3 | 55.1 | 60.0 | 64.0 | 75.1 | 72.7 | 62.2 |
| Llama2-7B-Chat | 50.8 | 56.0 | 50.2 | 53.5 | 60.6 | 74.5 | 69.0 | 59.2 |
| Llama2-13B | 52.2 | 68.2 | 55.6 | 62.0 | 69.5 | 78.2 | 77.2 | 66.1 |
| Llama2-13B-Chat | 54.4 | 66.0 | 53.6 | 55.0 | 70.1 | 76.8 | 75.6 | 64.5 |
| Llama2-7B-32K-Instruct | 45.8 | 66.2 | 64.5 | 60.4 | 69.0 | 75.7 | 70.1 | 64.5 |
| Vicuna-13B | 54.1 | 78.4 | 73.0 | 69.4 | 72.5 | 77.3 | 76.8 | 71.6 |

Table 7: Analysis of additional LLMs for the main experiments in terms of F1 (%). The models are trained on each ontology separately.

|  | **Wine** | **Economy** | **Olympics** | **Transport** | **SUMO** | **FoodOn** | **GO** | **Average** |
|---|---|---|---|---|---|---|---|---|
| CONDITIONAL HYBRID MODELS |  |  |  |  |  |  |  |  |
| GCN (UT) + GCN(BT) + RoBERTa | 78.0 | 79.5 | 84.5 | 77.0 | 77.7 | 78.5 | 77.6 | 80.0 |
| GCN (UT) + GCN(BT) + Mistral | 81.0 | 83.2 | 89.2 | 82.3 | 78.7 | 80.9 | 79.7 | 81.2 |
| GCN (UT) + GCN(BT) + Llama | 78.0 | 78.3 | 84.5 | 74.9 | 74.8 | 80.2 | 78.8 | 80.7 |
| GCN (UT$_{\geq 3}$) + GCN(BT$_{\geq 3}$) + RoBERTa | 78.9 | 87.6 | 86.6 | 79.7 | 87.6 | 83.4 | 78.7 | 83.2 |
| GCN (UT$_{\geq 3}$) + GCN(BT$_{\geq 3}$) + Mistral | **87.8** | **91.3** | **94.4** | **84.6** | 89.8 | **86.1** | 77.2 | 87.3 |
| GCN (UT$_{\geq 3}$) + GCN(BT$_{\geq 3}$) + Llama | 75.7 | 86.4 | 88.2 | 78.6 | 83.2 | 75.0 | 78.6 | 80.8 |
| SIMPLE AGGREGATION |  |  |  |  |  |  |  |  |
| GCN (UT) + GCN (BT) + RoBERTa | 72.3 | 75.7 | 80.0 | 74.1 | 72.4 | 80.7 | 80.1 | 81.1 |
| GCN (UT) + GCN (BT) + Mistral | 80.9 | 83.2 | 89.2 | 82.3 | 78.7 | 82.4 | **82.3** | 84.5 |
| GCN (UT) + GCN (BT) + Llama | 78.0 | 78.3 | 84.5 | 74.9 | 74.8 | 81.8 | 80.3 | 83.1 |
| GCN (UT$_{\geq 3}$) + GCN (BT$_{\geq 3}$) + RoBERTa | 68.2 | 82.6 | 80.6 | 76.4 | 76.8 | 77.0 | 72.9 | 76.4 |
| GCN (UT$_{\geq 3}$) + GCN (BT$_{\geq 3}$) + Mistral | **87.8** | **91.3** | **94.4** | **84.6** | **90.0** | 86.0 | 77.6 | **88.0** |
| GCN (UT$_{\geq 3}$) + GCN (BT$_{\geq 3}$) + Llama | 75.7 | 86.4 | 88.2 | 78.6 | 83.1 | 75.0 | 78.6 | 80.8 |

Table 8: Overview of the main results in terms of F1 (%) for the hybrid models with binary templates. *RoBERTa* refers to RoBERTa-large, *Mistral* refers to the Mistral-7B base model, and *Llama* refers to the instruction-finetuned Llama3-8B model.

for about twenty hours for the biggest ontologies. In contrast, to fine-tune the LLMs with QLoRA (Unsloth AI), we used a single A6000 GPU for less than one hour per ontology.

**Prompt Analysis** In Table 12, we compare the performance of different prompts. For this analysis, we use a test set that consists of 100 examples from each of the seven ontologies (50 positive and 50 negative examples) The prompts that we considered are as follows:

- Prompt 1: Classify the text into True or False. Reply with only one word: True or False. Determine if the following statement is valid:
- Prompt 2: Assess the validity of the following statement. Reply with only one word: True or

|  | Wine | Economy | Olympics | Transport | SUMO | FoodOn | GO | Average |
|---|---|---|---|---|---|---|---|---|
| Mistral-7B | 54.7 | 77.7 | 79.5 | 79.2 | 69.8 | 79.7 | **82.9** | 74.8 |
| Mistral-7B (IT) | 53.0 | 80.8 | 69.6 | 77.6 | 67.4 | **81.6** | 80.9 | 73.0 |
| Llama3-8B | **69.0** | **63.8** | **57.7** | **76.3** | 61.9 | 63.0 | 66.4 | **65.4** |
| Llama3-8B (IT) | 63.5 | **80.5** | 73.8 | **83.0** | 76.4 | **81.8** | 80.9 | 77.1 |
| Phi3-medium (IT) | 52.5 | 80.4 | 71.0 | **80.4** | 76.1 | 80.5 | **81.7** | 74.6 |
| Gemma-7B | **66.4** | 75.4 | **78.6** | **81.5** | 72.2 | 77.8 | **80.6** | **76.1** |
| Gemma-7B (IT) | **63.6** | **75.9** | 68.2 | 78.5 | 73.1 | 79.9 | 77.8 | 73.9 |
| Llama2-7B | **53.8** | **71.4** | **71.5** | **60.2** | **69.9** | **75.3** | **73.4** | **67.9** |
| Llama2-7B-Chat | 37.1 | **68.2** | **66.7** | **59.7** | **68.0** | 73.5 | **72.3** | **63.7** |
| Llama2-13B | 45.6 | **69.8** | **68.1** | 58.5 | 68.2 | 75.3 | 75.5 | 65.9 |
| Llama2-13B-Chat | 48.3 | **71.9** | **63.3** | **60.8** | 67.7 | 74.3 | 74.4 | **65.8** |
| Llama2-7B-32K-Instruct | **51.3** | 65.0 | **74.6** | 58.2 | 65.8 | 72.2 | 69.3 | **65.2** |
| Vicuna-13B | 48.6 | 72.3 | 63.5 | 59.7 | 71.8 | 77.1 | 76.3 | 67.0 |

Table 9: Results for the NLI models when jointly trained on all ontologies. Results which are better than the corresponding result in Table 2 and 7 are highlighted in bold.

|  | DistMult | TransE |
|---|---|---|
| Wine | 72.2 | 70.8 |
| Economy | 14.6 | 15.7 |
| Olympics | 27.9 | 27.3 |
| Transport | 47.0 | 45.9 |

Table 10: Comparison between DistMult and TransE as scoring function for the Binary Template model (F1%).

False. Determine if the following statement is valid:

- Prompt 3: Assess the validity of the following rule. Reply with only one word: True or False. Determine if the following rule is valid:
- Prompt 4: Classify the text into True or False. Reply with only one word: True or False. Determine if the following is a valid rule:
- Prompt 5: Classify the text into True or False. Reply with only one word: True or False. Determine if the following is valid statement:

In all cases, the prompt is followed by a statement of the form RULE BODY *implies* RULE HEAD. As we can see in Table 12, the performance of these prompts is comparable. Furthermore, the instruction fine-tuned models are generally more robust against changes in the prompt.

| | Wine | Economy | Olympics | Transport | SUMO | FoodOn | GO | Average |
|---|---|---|---|---|---|---|---|---|
| QLoRA 4-bit (Llama3-8B) | 57.9 | 57.9 | 51.8 | 62.0 | 75.5 | 72.5 | 77.4 | 65.0 |
| Full fine-tuning (Meta-Llama-3-8B) | 53.6 | 72.5 | 68.9 | 77.1 | 66.1 | 77.9 | 79.4 | 70.8 |

Table 11: Comparing fine-tuning in 4-bit precision using QLoRA with standard fine-tuning in full prevision.

| | Prompt 1 | Prompt 2 | Prompt 3 | Prompt 4 | Prompt 5 |
|---|---|---|---|---|---|
| Mistral-7B | 80.3 | 77.6 | 78.5 | 75.6 | 78.1 |
| Mistral-7B (IT) | 78.3 | 78.5 | 77.4 | 78.8 | 78.6 |
| Llama3-8B | 53.4 | 57.2 | 58.8 | 50.8 | 52.9 |
| Llama3-8B (IT) | 75.8 | 74.9 | 75.1 | 74.9 | 75.9 |
| Phi3-medium (IT) | 70.9 | 73.6 | 70.9 | 69.9 | 69.4 |
| Gemma-7B | 69.4 | 72.2 | 63.1 | 65.5 | 62.2 |
| Gemma-7B (IT) | 76.3 | 76.9 | 77.3 | 77.6 | 76.1 |
| Llama2-7B | 66.4 | 61.9 | 62.9 | 67.2 | 67.1 |
| Llama2-7B-Chat | 61.3 | 63.0 | 63.7 | 65.0 | 64.8 |
| Llama2-13B | 69.5 | 69.9 | 71.2 | 70.4 | 70.9 |
| Llama2-13B-Chat | 65.4 | 65.4 | 63.3 | 65.2 | 65.8 |
| Llama2-7B-32K-Instruct | 67.0 | 67.4 | 67.8 | 63.2 | 65.8 |
| Vicuna-13B | 69.7 | 69.0 | 68.4 | 69.8 | 68.6 |

Table 12: Analysis of the performance of different prompts in terms of F1 (%) on a combined test set containing 100 examples from each of the ontologies.