End-to-End Ontology Learning with Large Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Ontologies are useful for automatic machine processing as they represent knowledge in a structured format. Yet, constructing ontologies requires substantial manual effort. To automate part of this process, large language models (LLMs) have been applied to solve various subtasks of ontology learning. However, this partial ontology learning does not capture the interactions between subtasks. We address this gap by introducing OLLM, a general and scalable method for solving the *full* task of building an ontology from scratch. Rather than focusing on subtasks, like individual relations between entities, we model entire subcomponents of the target ontology by finetuning an LLM with a custom regulariser that reduces overfitting on high-frequency concepts. We introduce a novel suite of metrics for evaluating the quality of the generated ontology by measuring its semantic and structural similarity to the ground truth. Our metrics stem from modern deep learning evaluation techniques, but make fewer assumptions about the ontologies than standard ontology metrics. Our results on Wikipedia show that OLLM outperforms subtask composition methods, producing more semantically accurate ontologies while maintaining structural integrity. We further demonstrate that our model can be effectively adapted to a new domain, like arXiv, needing only a small number of training examples.

1 Introduction

An ontology is a formal and structural way of representing domain-specific concepts and their relations [**?**]. They can be simplistic consisting of *concepts* and only a small number of types of *taxonomic relations* (e.g., *is-a* relationships). Or they can be complex consisting of axioms and many types of relations. For example, a simple ontology for programming languages might contain two concepts "Dynamically-typed language" and "Python", and one relation "Dynamically-typed language → Python", representing the knowledge that Python is a dynamically-typed language. A more complex ontology might contain axioms too, for example, "all programming languages are either dynamically or statically typed". In this paper we focus on ontologies of the simpler type. Compared to typical deep learning models which represent knowledge implicitly in its weights, ontologies capture knowledge in a structured and explicit manner, making them reliable, easy to edit and human-interpretable. Such benefits of ontologies have led to their wide adoption in practice such as the Schema.org [**?**] ontology which is part of the Semantic Web [**?**] initiative.

While ontologies are useful, building ontologies often requires substantial manual effort. Ontology learning (OL) is the study of automating the construction of high-quality ontologies at scale. For a simplistic ontology, this amounts to discovering the concepts and taxonomic relations, usually based on a source corpus. In this paper we aim to develop domain-independent methods for OL that are scalable and produce better ontologies.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

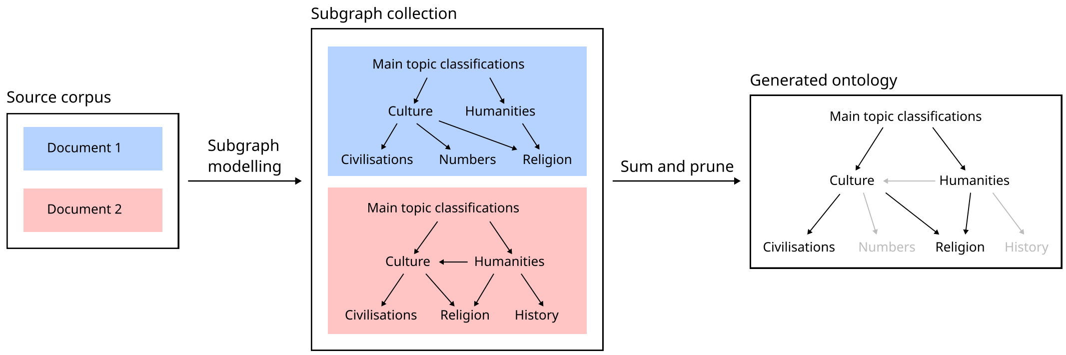


Figure 1: Overview of OLLM. A finetuned LLM is used to model the relevant subgraph for each document in the source corpus. The generated subgraphs (sub-ontologies) are then summed into a weighted graph, and pruning is applied to obtain the final output ontology.

Traditionally, OL is viewed as a composition of subtasks [**?**], such as concept discovery and relation extraction. In particular, prior works have demonstrated that state-of-the-art large language models (LLMs) can solve such subtasks effectively [**?**]. While studying subtasks permits fine-grained analysis and evaluation, it does not directly reflect the downstream impact on the final ontology. Moreover, there is potential room for improvement by combining several subtasks into one. In this paper, we instead develop and evaluate methods that construct ontologies in an end-to-end fashion to answer the following research questions:

1. How can we leverage LLMs' knowledge base to build ontologies from scratch?
2. Does our method scale efficiently to practical problem sizes?
3. How well does our method generalise to new domains?

We introduce OLLM, an end-to-end method for using LLMs to construct ontologies at scale. Rather than focusing on individual relations between concepts, we finetune an LLM to model entire sub-components of the target ontology. The output ontology is generated by taking the sum of generated sub-components and applying simple post-processing. An overview of the pipeline is shown in Fig. 1. To train OLLM, we collect the categorisation metadata for a subset of Wikipedia articles. We attempt to adapt an LLM to model the relevant categorisation subgraph for a particular Wikipedia article, but discover that direct finetuning leads to poor generalisation due to overfitting to high-level, frequently occurring concepts. Instead, we propose a custom regulariser that reweights each concept based on its frequency of occurrence, which substantially improves generalisation.

We evaluate OLLM by measuring the similarity of the generated ontology with the ground truth. Current approaches for comparing ontologies rely on mapping classes of the two ontologies onto each other, most commonly by literal text matching. TODO: Add citation This is unreliable when the two ontologies are not already sufficiently similar. Instead, we propose a suite of evaluation metrics suitable for comparing arbitrary labelled graphs. These metrics compare edges and subgraphs of the two ontologies using pretrained text embedders to test for semantic and structural similarity. The results reveal that an LLM can already outperform existing extraction-based methods out of the box, and the performance can be further improved by finetuning with our custom regulariser. We additionally demonstrate that OLLM can be adapted to build the arXiv ontology using only a small number of training examples, suggesting that our model can be applied to new domains in a data-efficient way.

Contributions

1. We constructed two datasets based on Wikipedia and arXiv, which can serve as standard datasets for future work studying end-to-end OL.
2. We created OLLM, a method that utilises LLMs to build ontologies from scratch. OLLM produces high-quality ontologies and serves as a strong baseline for end-to-end OL.
3. We developed new evaluation metrics for end-to-end OL.

2 Background

An ontology is a structured way of representing concepts and relations of a shared conceptualisation, i.e. domain knowledge [**?** **?**]. In this paper, we focus on simplistic ontologies that only consist of concepts and taxonomic relations which represent *is-a* or *is-subclass-of* relationships between concepts. In some cases, the *is-part-of* relation is also considered a taxonomic relation. We treat such an ontology as a rooted labelled directed graph where nodes represent concepts, edges represent taxonomic relations and the root node is the special concept of all concepts. A strict ontology asserts that the taxonomic relation is asymmetric and thus the graph must be acyclic, though in practice some ontologies, such as the Wikipedia ontology studied in this paper, may contain cycles. We therefore do not assume that an ontology graph is necessarily acyclic. Examples of ontologies include WordNet [**?**] with 117,659 concepts and 89,089 taxonomic relations and the Gene Ontology [**?**] with 42,255 concepts and 66,810 taxonomic relations.

Ontology learning is the automatic extraction of ontological elements [**?**]. The most studied source of input is unstructured text, though there are also works on OL on semi-structured data like HTML [**?**]. In this paper, the input is a set of documents, each consisting of some unstructured text. We additionally assume each document is associated with one or more concepts in the ground truth ontology which we utilise for training. The goal is to reconstruct the ground truth ontology given the set of documents.

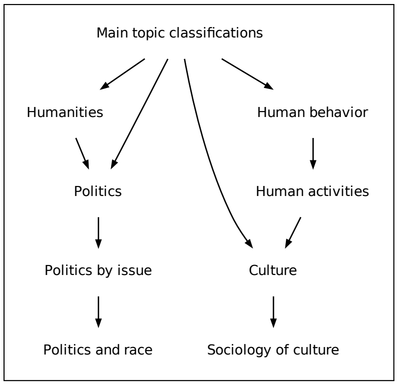
Prior works view OL as a composition of subtasks and study each subtask in isolation [**?** **?**]. A typical pipeline for building a simple ontology is to first perform concept discovery (identify the nodes) and then relation extraction (identify the edges) [**?** **?**]. A notable approach for relation extraction is Hearst patterns [**?**]. Hearst patterns are hand-crafted lexico-syntactic patterns that exploit natural language structure to discover taxonomic relations. For example, the pattern "NP such as NP" matches phrases like "dogs such as chihuahuas" and thus can be processed by regular expressions to identify the relation "dog → chihuahua". Hearst patterns suffer from low recall as the relations must occur in exact configurations to be matched by rules. More recent works have suggested smoothing techniques to alleviate this issue [**?**].

Recent research has transitioned to using language models for OL. REBEL [**?**] treats relation discovery as a translation task and finetunes encoder-decoder LLMs to extract both taxonomic and non-taxonomic relations. **?**] benchmarked a wide family of LLMs for concept and relation discovery and showed promising results. There are also proof-of-concept works for building ontologies end-to-end with LLMs. **?**] proposes to build an ontology by recursive prompting an LLMs while **?**] generates the entire ontology in one completion. However, both studies are limited in the scale of the task and evaluation. The authors only considered ontologies of up to 1000 concepts and relied on manual qualitative evaluation. We bridge this gap by proposing a method that can scale to practical problem sizes and new metrics for systematic qualitative evaluation.

The evaluation of ontologies is also an open research area. The main approaches are gold-standard evaluation, which matches elements of the generated ontology with a predefined target ontology; task-based evaluation, which measures the usefulness of the ontology on a specific application; and human evaluation [**?** **?**]. In this paper, we evaluate by the gold standard as it is the most straightforward approach when such ground-truth ontology exists. Prior works have considered matching concepts [**?**] and direct and indirect relations [**?** **?**] by literal text comparison. Other works have also considered edit-distance [**?**] or bag-of-words distributional similarity for text comparison [**?**]. These techniques may be considered unreliable and have been superseded by current methods [**?**]. We instead rely on more modern techniques like pretrained text embedders [**?**] and graph convolutions [**?**] to match substructures between the two ontologies.

3 OLLM

This section introduces OLLM, a simple and scalable method for end-to-end OL with LLMs. On a high level, OLLM uses an LLM to model linearised subgraphs of the target ontology. In contrast to learning individual edges, modelling subgraphs allows the model to learn higher-order structures, such as the interactions between three or more nodes. To create the training dataset, OLLM relies on the assignment of documents to concepts which induces a relevant subgraph for each document. Such subgraphs are much smaller than the complete graph so they can be learned by the model more




```
<s>[INST] Title: Hybridity
Hybridity, in its most basic sense ... [/INST]
Main topic classifications -> Human behavior
    -> Human activities -> Culture ->
    Sociology of culture
Main topic classifications -> Humanities ->
    Politics -> Politics by issue -> Politics
    and race
Main topic classifications -> Politics ->
    Politics by issue -> Politics and race
Main topic classifications -> Culture ->
    Sociology of culture</s>
```

Figure 2: Example subgraph induced by the Wikipedia page "Hybridity" (left), where $N = 4$ and $C = \{\text{Politics and race}, \text{Sociology of culture}\}$. The corresponding training text sequence (right), where text coloured in grey is ignored as training targets but is still present as context for later tokens.

easily. The generated subgraphs for each document are summed into a weighted graph and simple post-processing is applied to obtain the final predicted ontology.

3.1 Subgraph modeling

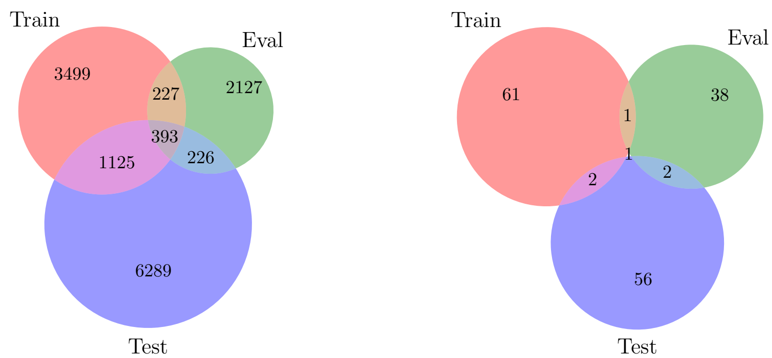
Here, we describe the method for creating document-subgraph pairings. Given a document and its associated set of concepts C, we define the *relevant paths* as the set paths of at most length N from the root to any of the concepts in C. The *relevant subgraph* is the set of nodes and edges that occur at least once in the relevant paths. An example is shown in the left subfigure of Fig. 2. The choice of N is task-specific and we describe our method for choosing N in Section 5.1.

To employ LLMs to model the subgraphs, we must linearise the graph into a string for sequence modelling. Existing methods for autoregressive graph generation employ BFS [**?**] or DFS [**?**] ordering starting at an arbitrary node. We instead choose to linearise the subgraph as a list of relevant paths that produced the subgraph in the first place. We do so for three reasons: Firstly, the subgraph is defined from such a collection of paths which makes them the most natural representation; Secondly, we hypothesise the hierarchy of concepts on each path is a desirable inductive bias for the hierarchical nature of an ontology; Thirdly, the path-based representation is much easier to describe in natural language instructions so that our LLM prompting-based baselines may produce reasonable results without finetuning. The linearisation template can be found in Appendix A.2.

3.2 Post-processing

The final output graph is obtained by summing all generated subgraphs for each document and pruning low-weighted components. Given the generated subgraphs $G_1 = (V_1, E_1), \dots, G_n = (V_n, E_n)$, the raw output graph is defined as $G_{\text{raw}} = (V_{\text{raw}}, E_{\text{raw}})$ where $V_{\text{raw}} = \cup_{i=1}^{n} V_n$ and $E_{\text{raw}} = \cup_{i=1}^{n} E_n$. Each edge $(u, v) \in E_{\text{raw}}$ is additionally weighted by the number of times they occur in the collection of subgraphs: $w_{u,v} = \sum_{i=1}^{n} \mathbb{1}[(u, v) \in E_n]$. A few simple post-processing steps are then applied to G_{raw}:

1. Self-loop pruning: All edge $(u, u) \in E_{\text{raw}}$ are removed.
2. Inverse-edge pruning: All edges $(u, v) \in E_{\text{raw}}$ where $(v, u) \in E_{\text{raw}}$ and $w_{v,u} > w_{u,v}$ are removed.
3. Absolute thresholding: Edges in E_{raw} with weight below the α-th quantile are removed, where $0 \leq \alpha \leq 1$ is a hyperparamter.
4. Relative thresholding: For each vertex $u \in V_{\text{raw}}$, let $e_1, \dots, e_k$ be the outgoing edges from u sorted by weight in ascending order. Define the cumulative weight as $C(e_i) = \sum_{j=1}^{i} w_{e_j} / \sum_{j=1}^{k} w_{e_j}$. The edges $\{e_i \mid C(e_i) \leq \beta\}$ are pruned, where $0 \leq \beta \leq 1$ is a hyperparameter.
5. Clean up: After pruning all edges, nodes with no incoming or outgoing edges are removed.



(a) Wikipedia (b) arXiv

Figure 3: Intersection of nodes in the train, eval and test split of the datasets.

In our implementation, we choose the hyperparameters α and β by tuning on the validation set.

4 Evaluating end-to-end OL

Since our problem setup is uncommon in existing literature, we also develop new evaluation methods. Ontology evaluation is a hard problem as there are no quantitative definitions of what constitutes a "good ontology" and metrics generally only capture one aspect of an ontology. We approach evaluation by treating the ground truth as a proxy for a good ontology and comparing the generated ontologies against the ground truth. This section describes how the ground truth is obtained and what metrics are used for measuring ontology similarity.

4.1 Dataset

We collect the datasets for the two ontologies considered in this paper: Wikipedia categories and the arXiv taxonomy. We use Wikipedia for learning and in-domain evaluation and arXiv for out-of-domain evaluation. To build the Wikipedia dataset, we perform a BFS traversal from its root category "Main topic classifications" up to depth 3. For every category encountered, we retrieve the title and summary (the text before the first section) of up to 5000 pages that belong in that category. The source data is obtained from the Wikipedia API.[1] The arXiv taxonomy is available from its home page and the source corpus is constructed from the title and abstract of all the papers uploaded to arXiv in the years 2020–2022 with more than or equal to 10 citations.[2] In total, the Wikipedia dataset has 13,886 concepts, 28,375 taxonomic relations and 362,067 documents, while the arXiv dataset has 161 concepts, 166 taxonomic relations and 126,001 documents.

Generating the train and test splits from the datasets is also a non-trivial problem. As described in Section 3.1, each training example consists of a document and its induced subgraph. The naive approach of randomly selecting a subset of documents for the training set likely leads to data leakage as there might be a significant overlap between subgraphs in the training set and the test set. Instead, we propose to first split the full ontology in train and test graphs and then generate the training document-subgraph pairs. Our method is as follows:

1. Let V^{top} be the set of top-level nodes, i.e. children of the root node. Randomly partition V^{top} into train $V^{\text{top}}_{\text{train}}$, validation $V^{\text{top}}_{\text{val}}$, and test $V^{\text{top}}_{\text{test}}$ splits in 7:3:10 ratio.
2. Let d be the depth on the full graph, i.e. the distance of the furthest node from the root. The nodes of the train graph are taken as the union of all the nodes that are within distance $d - 1$ from any node in $V^{\text{top}}_{\text{train}}$, plus $V^{\text{top}}_{\text{train}}$ and the root. The edges are all the edges in the full graph that have both endpoints in the train graph. Similar applies for $V^{\text{top}}_{\text{val}}$ and $V^{\text{top}}_{\text{test}}$.

[1] https://en.wikipedia.org/w/api.php
[2] Citation counts obtained from https://api.semanticscholar.org/.

Our methods ensure that there are sufficiently many unseen concepts (and thus relations) in the test split, as shown in Fig. 3.

4.2 Metrics

Existing methods for measuring similarity between ontologies rely on outdated techniques such as edit distance or document co-occurrence statistics for text comparison. To obtain more reliable evaluation results, we propose a suite of similarity metrics that uses more modern methods like text embeddings. Multiple metrics are used as they trade off interpretability with comprehensiveness, and we aim to make them complementary by capturing different aspects of an ontology. In this section, we denote the ground truth ontology graph as $G = (V, E)$ and the generated graph as $G' = (V', E')$.

Literal F1 [?] While literal text matching is unreliable, it is also the simplest and the most interpretable. The Literal F1 metric is given by the harmonic mean of the precision and recall of the edges:

$$\text{Literal precision} = \frac{|E \cap E'|}{|E'|} \qquad \text{Literal recall} = \frac{|E \cap E'|}{|E|}$$

Fuzzy F1 The literal F1 metric puts a strong emphasis on using the correct wording, while in practice, we are interested in evaluating the semantics of an ontology. For example, using a synonymous phrase for a concept should not be penalised. We utilise embeddings from a pretrained sentence transformer and use the cosine similarity of the embeddings to measure semantic similarity. Specifically, let $\text{NodeSim}(u, u') \in V \times V' \to [-1, 1]$ be the cosine similarity between the sentence embeddings for u and u'. The Fuzzy F1 score is obtained from the fuzzy precision and recall, defined as:

$$\text{Fuzzy precision} = \frac{|\{(u', v') \in E' \mid \exists (u, v) \in E.\, \text{NodeSim}(u, u') > t \wedge \text{NodeSim}(v, v') > t\}|}{|E'|}$$

$$\text{Fuzzy recall} = \frac{|\{(u, v) \in E \mid \exists (u', v') \in E'.\, \text{NodeSim}(u, u') > t \wedge \text{NodeSim}(v, v') > t\}|}{|E|}$$

where t is the matching threshold. We use all-MiniLM-L6-v2 [? ?] as the embedding model and choose t as the median cosine similarity between the synonyms in WordNet [?], computed to be 0.436.

Continuous F1 With fuzzy comparisons, the matches between the edges of the generated and the ground truth graph are no longer one-to-one. This is problematic: Consider two graphs $A \to B$ and $B \leftarrow A \to B'$, where B and B' match fuzzily. Such graphs will achieve a perfect fuzzy F1 score yet they significantly differ. Additionally, we found that the previous metrics fail to provide a useful signal for hyperparameter tuning, particularly for our baselines where the generated graphs are poor. The continuous F1 metric solves these issues by computing the highest-scoring edge matching between the two graphs, where the similarity score between (u, v) and (u', v') is given by $\min(\text{NodeSim}(u, u'), \text{NodeSim}(v, v'))$. Obtaining such matching is equivalent to solving the linear assignment problem [?], which can be computed by the Hungarian algorithm [?]. The Continuous F1 is obtained from the continuous precision and recall, given by:

$$\text{Continuous precision} = \frac{s_{\text{cont}}}{|E'|} \qquad \text{Continuous recall} = \frac{s_{\text{cont}}}{|E|}$$

where s_{cont} is the score achieved by the best edge matching.

Graph F1 Instead of individual edges, this metric aims to capture the wider structure of the two graphs. Intuitively, we want to know how concepts are related to their local neighbourhood. We do so by using simple graph convolutions [?] with $K = 2$ to compute graph-aware node embeddings after embedding each node with the pretrained embedder. Such embeddings in G are compared against those in G' by cosine similarity, and the highest-scoring node matching, similar to the continuous F1 metric, gives the graph similarity score. The Graph F1 is computed from the graph precision and recall, defined to be:

$$\text{Graph precision} = \frac{s_{\text{graph}}}{|V'|} \qquad \text{Graph recall} = \frac{s_{\text{graph}}}{|V|}$$

where s_{graph} is the score achieved by the best node matching.

```
<s>[INST] Title: List of general awards in the humanities
This list of general awards in the humanities ... from that country. [/INST]
Main topic classifications -> Society -> Humanities -> Humanities awards
Main topic classifications -> Academic disciplines -> Humanities -> Humanities awards
Main topic classifications -> Society -> Culture -> Cultural lists -> Lists of awards
Main topic classifications -> Culture -> Cultural lists -> Lists of awards
Main topic classifications -> Lists -> Cultural lists -> Lists of awards
Main topic classifications -> Humanities -> Humanities awards
Main topic classifications -> Academic disciplines -> Liberal arts education -> Humanities -> Humanities awards
```

(a) Direct finetuning

```
<s>[INST] Title: List of general awards in the humanities
This list of general awards in the humanities ... from that country. [/INST]
Main topic classifications -> Society -> Humanities -> Humanities awards
Main topic classifications -> Academic disciplines -> Humanities -> Humanities awards
Main topic classifications -> Society -> Culture -> Cultural lists -> Lists of awards
Main topic classifications -> Culture -> Cultural lists -> Lists of awards
Main topic classifications -> Lists -> Cultural lists -> Lists of awards
Main topic classifications -> Humanities -> Humanities awards
Main topic classifications -> Academic disciplines -> Liberal arts education -> Humanities -> Humanities awards
```

(b) Finetuning with masked loss

Figure 4: Per token loss on an example from the test set of the final model trained with and without the custom masked loss objective. We observe that using the masked loss objective improves generalisation on the high-level relations while maintaining performance on lower-level relations.

Motif distance Taking inspiration from classical network analysis, we use *network motifs* [**? ?**] to evaluate the structural integrity of the generated graphs. Network motifs are reoccurring subgraphs in a larger graph, most commonly 3-vertex subgraphs. They are typically indicative of the structural characteristics of the full graph. We define the motif distance as the 1-Wasserstein distance between the distribution of all 3-vertex subgraphs in G and G'.

5 Experiments

We design our experiments to answer the following research questions:

1. Does OLLM produce better ontologies than traditional methods by subtask composition?
2. Can OLLM be easily adapted to a new domain?

We approach the questions by training OLLM on the Wikipedia dataset and further transfer the model to arXiv with a small number of arXiv samples. As baselines, we use two relation extraction methods, Hearst patterns [**? ?**] and REBEL [**?**]. Relation extraction depends on successful concept discovery to produce high-quality ontologies. To estimate a ceiling to such baselines, *we give the baselines a substantial advantage* by providing them with the ground truth concepts in the test graph. The results show that even with such an advantage, OLLM outperforms the baselines on many metrics, demonstrating the potential of OLLM for end-to-end OL.

5.1 Implementation details

We discover that directly finetuning an LLM on the sequences defined in Section 3.1 produces poor results due to overfitting. Analysing the per-token loss of a naively finetuned model on the test split shows that the model tends to memorise high-level relations from the training set, leading to poor generalisation as shown in Fig. 4 (top). This occurs because high-level relations are present in many relevant subgraphs and thus repeated many times in the training set. This problem is not solvable by early stopping since terminating training early will result in a model that massively underfits lower-level relations.

This issue is akin to multi-task learning [**?**] where the standard solution is to apply some loss weighting factor to rebalance training objectives [**? ?**]. We draw inspiration from this connection and propose a new training objective that randomly masks the loss contribution from frequently occurring relations. Suppose a relation $u \to v$ is present n times in the training set. During training, when $u \to v$ appears in one of the relevant paths, we mask the tokens for v with probability $\max(1 - M/n, 0)$, where M is a constant for the average number of times a relation is present in the training set. Note that while v is masked from the target, its tokens are still present in the input sequence as context for later tokens. A concrete example is shown in Fig. 2 (right).

We finetune Mistral 7B v0.2 [**?**] with Low-Rank Adaptation [**?**] on the masked loss objective. The model is trained on the Wikipedia dataset for two epochs with Adam. During inference, the outputs are generated with temperature 0.1 and nucleus sampling [**?**] top-p of 0.9. The weight of each edge is given by the number of generated subgraphs in which it appears. We include a finetuning baseline without the masked loss objective, denoted as **Finetune**. To adapt OLLM for arXiv, we further finetune the model on 2048 document-subgraph pairs from arXiv. We initialise new low-rank adaptors and train until the loss stops improving on the validation set. We name these models **OLLM (transfer)** and **Finetune (transfer)** for training without and without the masked loss objective respectively. Full details for the Wikipedia and arXiv experiments can be found in Appendix A.1.1.

The hyperparameters for the post-processing steps are tuned by grid search on the validation set. We sweep over $\alpha \in 1 - \text{geomspace}(1/|E_{\text{raw}}|, 1, 21)$ and $\beta \in \text{geomspace}(0.1, 1, 21) - 0.1$ and use the values that maximises the continuous F1 metric. For Wikipedia, we choose the subgraph modelling path length $N = 4$ as it is the smallest N such that almost all edges ($> 99\%$) occur in at least one induced subgraph. Such criterion is used as smaller N results in smaller subgraphs which we expect to be easier to model accurately. We choose $N = 3$ for arXiv for the same reason.

5.2 Baselines

We give a brief overview of the baseline methods here. The full implementation details can be found in Appendix A.1. All baselines produce weighted directed graphs which we apply the same post-processing steps as OLLM (Section 3.2) to obtain the final predicted graph.

Memorisation Simply memorising the train graph is a surprisingly strong baseline due to the overlap between train and test graphs, especially for Wikipedia. The weight of each edge is given by the number of relevant subgraphs in which it appears.

Hearst We follow the improved implementation of Hearst patterns by **?**]. The authors propose spmi, a method which uses low-rank approximations to smooth the relation matrix so that two concepts can be compared even if there are no direct matches between them. We use the smoothed relation matrix to weigh the relations between the ground truth concepts. The additional hyperparameter for the rank of the smoothed matrix is tuned by grid search over the validation set.

REBEL The REBEL-large model [**?**] is an encoder-decoder LLM trained to extract many types of relations from Wikipedia articles. We only take the "subclass of", "instance of", "member of" and "part of" relations that were extracted. Similar to **Hearst**, we find that it fails to find many direct relations between ground truth concepts. The same low-rank smoothing technique is applied to give a higher recall.

Prompting We test the **zero/one/three-shot** performance of instruction-tuned LLMs on the subgraph modelling task described in Section 3.1. We use Mistral 7B Instruct v0.2 [**?**] as the instruct model. We perform manual prompt engineering to describe the task and steer the model to return outputs of the same format as that described in Section 3.1. The prompt can be found in Appendix A.2.

5.3 Results

Our evaluation results reveal that OLLM produces both semantically and structurally more accurate ontologies than our baselines. Inspecting the metrics for the Wikipedia task in Table 1, we see that although OLLM is outperformed by the **Memorisation** and **Finetune** on Literal F1, it is much better at the Fuzzy, Continuous and Graph F1 metrics. This suggests that while OLLM produces ontologies that are *syntactically* less aligned to the ground truth, it better captures the overall semantics. In fact, our prompting baselines following the same task format as OLLM also outperform **Hearst** and **REBEL** in the semantics-aware metrics, though they suffer in structural integrity as reflected by the high Motif Distance. The results also hint at the potential pitfalls of syntax-based evaluation metrics as we see syntactic similarity does not generally entail semantic similarity.

Table 1: TODO: End with a take home message

Dataset	Method	Literal F1 ↑	Fuzzy F1 ↑	Cont. F1 ↑	Graph F1 ↑	Motif Dist. ↓
Wikipedia	Memorisation	**0.134**	0.837	0.314	0.419	**0.063**
	Hearst	0.003	0.538	0.350	0.544	0.163
	Rebel	0.004	0.624	0.356	0.072	0.132
	Zero-shot	0.007	0.871	0.455	0.639	0.341
	One-shot	0.031	0.888	0.477	0.610	0.314
	Three-shot	0.031	0.880	0.475	0.622	0.354
	Finetune	0.124	0.884	0.470	0.588	0.050
	OLLM	0.093	**0.915**	**0.500**	**0.644**	0.080
arXiv	Memorisation	0.000	0.207	0.257	0.525	**0.037**
	Hearst	0.000	0.000	0.151	0.553	0.098
	Rebel	0.000	0.060	0.281	0.546	0.088
	Zero-shot	0.025	0.450	0.237	0.414	0.145
	One-shot	**0.072**	0.460	0.290	0.433	0.293
	Three-shot	0.051	0.405	0.212	0.385	0.124
	Finetune (transfer)	0.000	0.440	0.225	0.441	0.148
	OLLM (transfer)	0.040	**0.570**	**0.357**	**0.633**	0.097

The arXiv task differs from the Wikipedia task as it has much fewer relations and there is even less overlap between the train and test split. This imposes a great challenge on **Finetune** and OLLM as they need to generalise with a limited diversity of training samples. Despite such constraints, OLLM is substantially better than other methods in modelling the semantics of the test graph. Inspecting the generated outputs, we observe prompting baselines tend to produce repetitive concepts such as "Machine Learning and Artificial Intelligence" and "Artificial Intelligence and Machine Learning" while **Hearst** and **REBEL** put "Machine Learning" as the parent concept of almost all ground truth concepts. Plots for the generated graphs can be found in Appendix A.3.

6 Discussion

In this paper, we introduce a general method for building ontologies in an end-to-end fashion. We propose a set of metrics for end-to-end OL that measures the semantic and structural similarity between arbitrary labelled graphs. Our model, OLLM, outperforms traditional subtask composition methods in reconstructing the Wikipedia categories and can be transferred to build ontologies for arXiv after finetuning on a small number of examples. Using LLMs as the backbone for subgraph modelling opens up exciting avenues for future research. For example, one may generate ontologies from corpora with images using vision language models [**?**].

We only study and evaluate the construction of simple ontologies with only concepts and taxonomic relations. A potential approach to extend OLLM to produce non-taxonomic relations is to add tags indicating the relation type to each edge when linearising the subgraphs for sequence modelling. New evaluation metrics might also be required to handle multiple types of relations. Another limitation is that we are unable to fully control for data contamination as the pretraining dataset of Mistral 7B is not publically known. We do, however, observe that the generated ontologies are sufficiently different from the ground truth, indicating that OLLM is not simply remembering samples from its pretraining stage.

A Appendix / supplemental material

A.1 Experiment details

A.1.1 OLLM training

For the Wikipedia experiment, we use Mistral 7B v0.2 (not instruction-tuned) [**?**] as the base model. We attach LoRA [**?**] adaptors to all attention and feed-forward layers with parameters $r = 32$ and $\alpha = 16$. The model is trained for 2 epochs ($\approx$ 17K steps) with batch size 16, context length 2048, and is optimised with Adam using a constant learning rate of 1e-5 with warm-up from zero for the first 100 steps. **Finetune** uses the same configuration. Training on two A100 GPUs takes $\approx$ 6 hours.

For the arXiv experiment, we further finetune the model trained on Wikipedia with masked loss objective on 2048 document-subgraph pairs from the arXiv training set. We merge the LoRA adaptors from the Wikipedia experiment and initialise new ones with $r = 8$ and $\alpha = 8$. The model is trained with batch size 16 and Adam with constant learning rate 3e-6 and warp-up from zero for the first 10 steps. Early stopping is used to terminate training when the loss stops improving on the evaluation set, which happened at step 288. **Finetune (transfer)** uses the same configuration. Eearly stopping happened at step 192.

A.1.2 Hearst

The **Hearst** baseline follows the implementation by **?**]. We give a description of the implementation here. TODO: CoreNLP, blah blah blah...

A.1.3 REBEL

A.1.4 Prompting

We sample the one/three-shot examples from the training set for each query. The output is parsed using regex and results that do not match the regex are discarded.

A.2 Prompt templates

OLLM finetuning template

```
<s>[INST]\
Title: {{ title }}
{{ abstract }}[/INST]\
{% for path in paths %}
{{ path | join(" -> ") }}
{% endfor %}\
</s>
```

Zero, one, and three-shot prompt template

```
The following is an article's title and abstract. Your task is to assign this
    article to suitable category hierarchy. A category is typically represented by
    a word or a short phrase, representing broader topics/concepts that the article
     is about. A category hierarchy represented by a collection of paths from the
    generic root category "Main topic classifications" to a specific category
    suitable for the article. The topics titles should become more and more
    specific as you move from the root to the leaf.

{% if examples|length > 0 %}
{% for example in examples %}
### EXAMPLE {{ loop.index }} ###
### ARTICLE ###
Title: {{ example['title'] }}
{{ example['abstract'] }}
### END ARTICLE ###
{% for path in example['paths'] %}
{{ path | join(" -> ") }}
{% endfor %}
### END EXAMPLE {{ loop.index }} ###
{% endfor %}
{% else %}
You must answer in the format of:
Main topic classifications -> Broad topic 1 -> Subtopic 1 -> ... -> Most specific
    topic 1
Main topic classifications -> Borad topic 2 -> Subtopic 2 -> ... -> Most specific
    topic 2
...
{% endif %}

### ARTICLE ###
Title: {{ title }}
{{ abstract }}
### END ARTICLE ###

Provide a category hierarchy for the above article. \
{% if examples|length > 0 %}
Use the same format as the examples above.
{% else %}
Use the format described above.
{% endif %}
```

A.3 Visualisation of generated ontologies

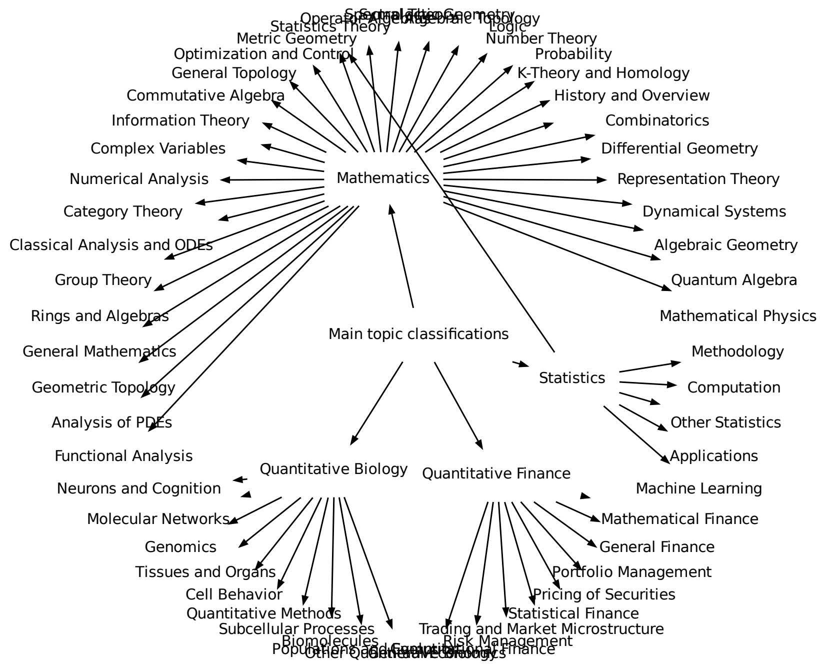


Figure 5: Ground truth test split ontology for arXiv

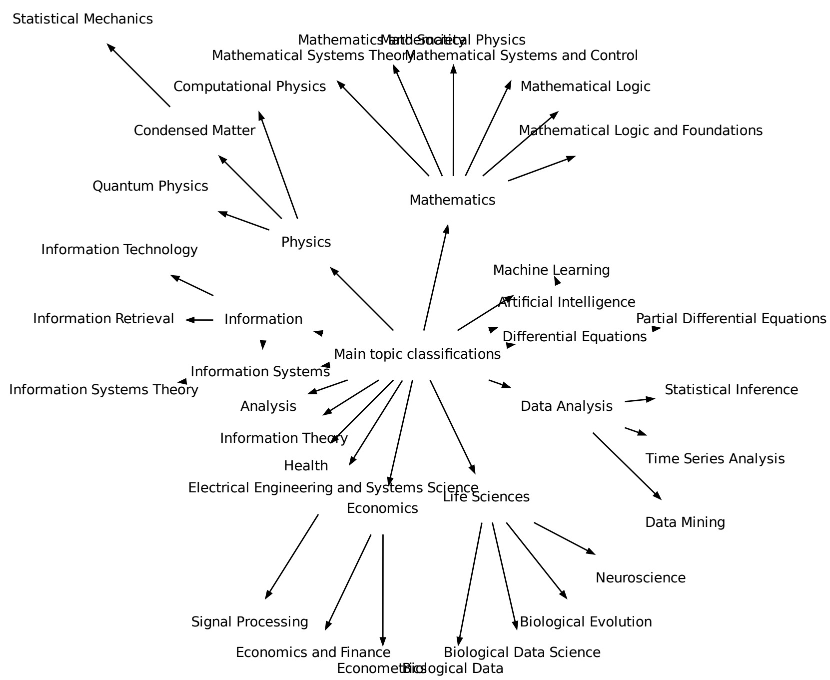


Figure 6: Ontology for arXiv generated by OLLM

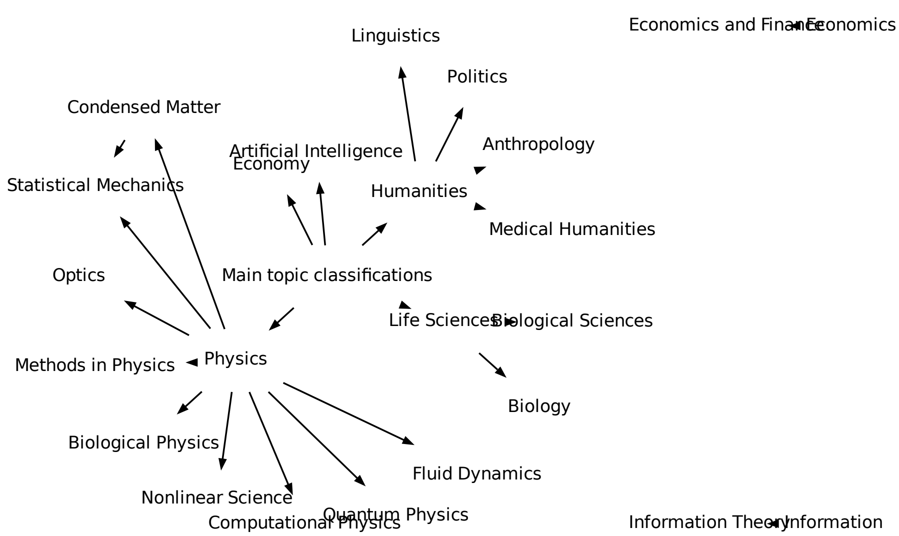


Figure 7: Ontology for arXiv generated by Finetune

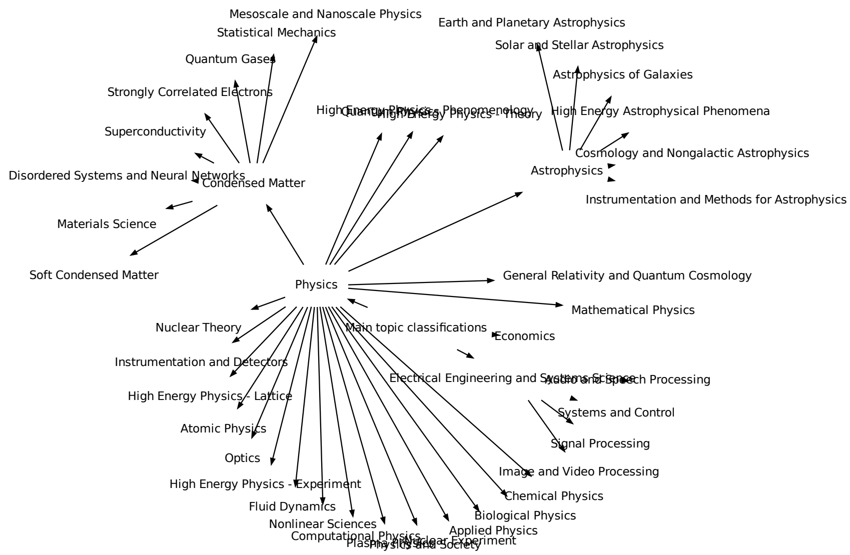


Figure 8: Ontology for arXiv generated by Memorisation

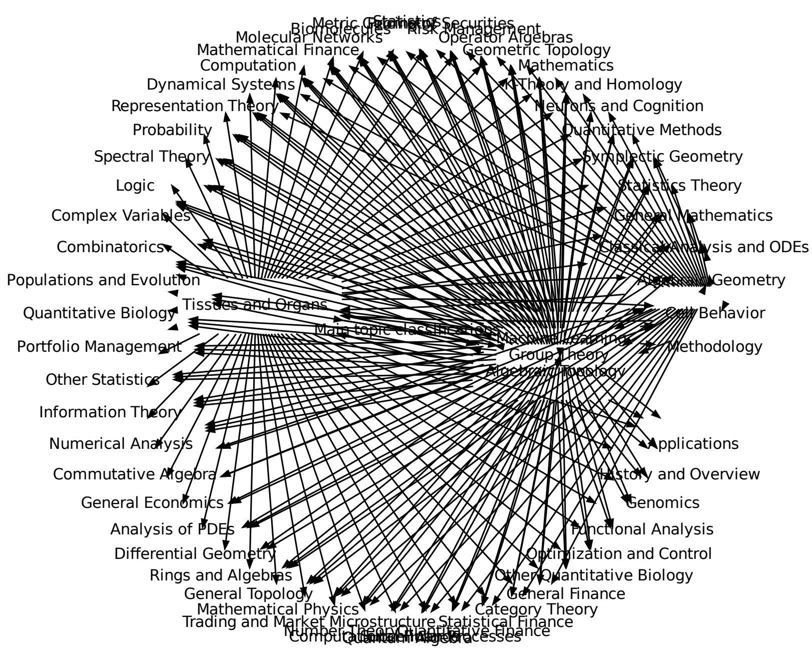


Figure 9: Ontology for arXiv generated by Hearst

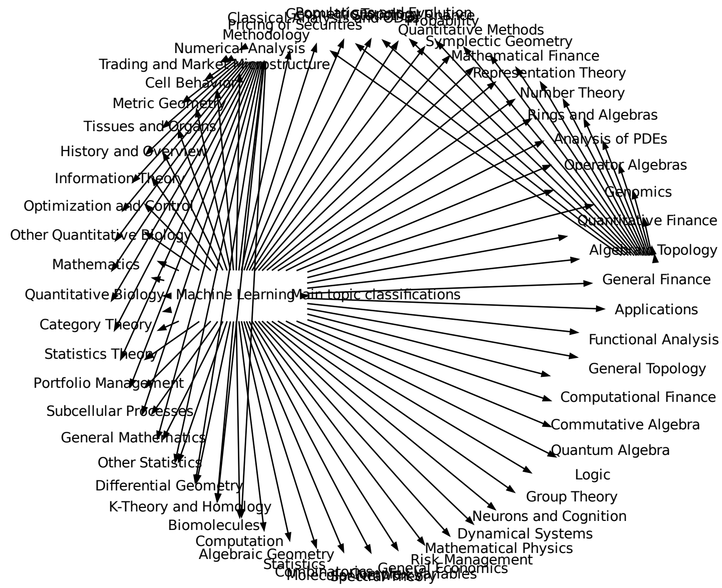


Figure 10: Ontology for arXiv generated by REBEL

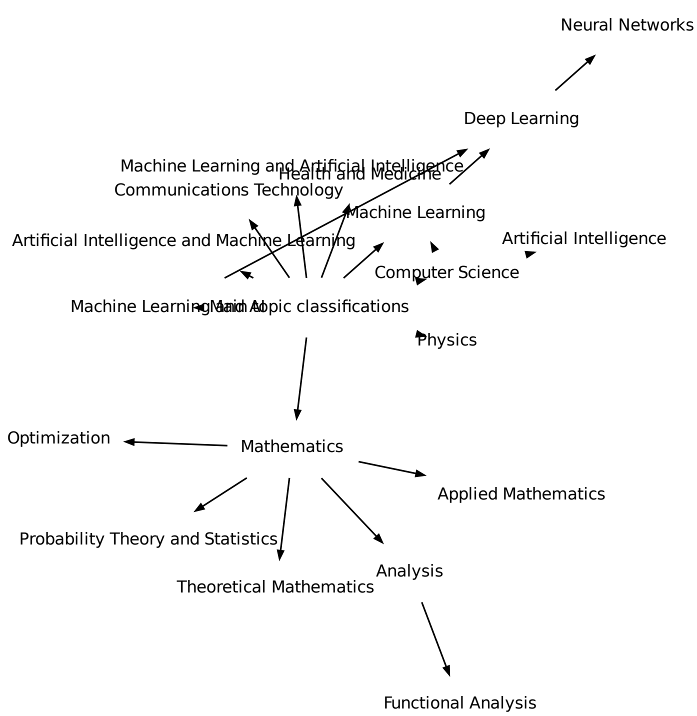


Figure 11: Ontology for arXiv generated by zero-shot

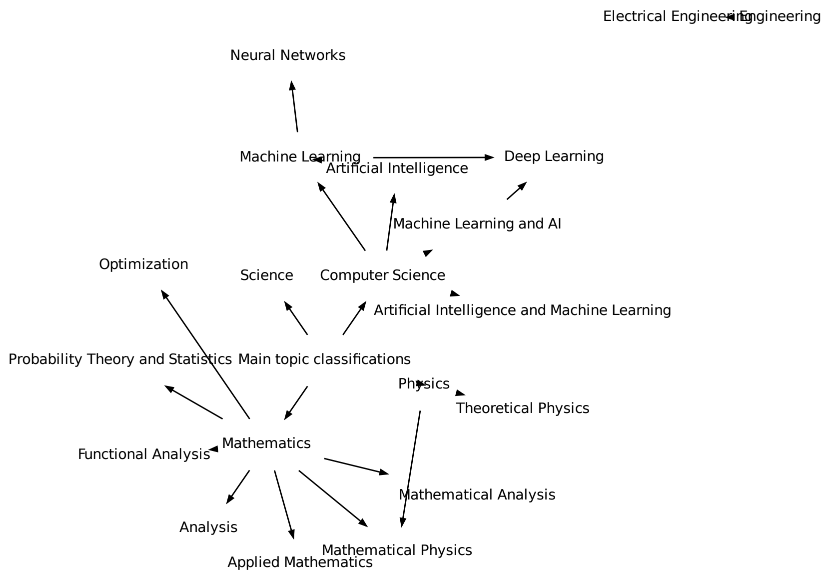


Figure 12: Ontology for arXiv generated by one-shot

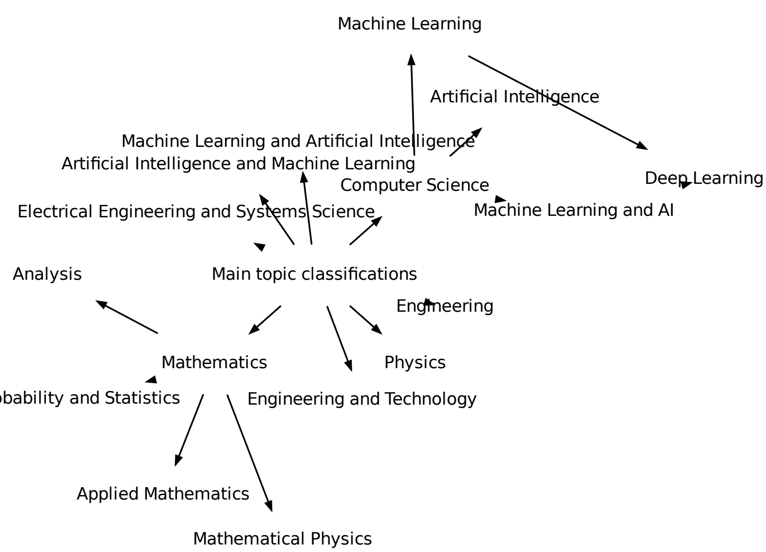


Figure 13: Ontology for arXiv generated by three-shot

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.