Are Large Language Models Good at Lexical Semantics? A Case of Taxonomy Learning

Viktor Moskvoretskii^{1,2}, **Alexander Panchenko**^{1,3}, **Irina Nikishina**^{4}
^{1}Skoltech, ^{2}HSE University, ^{3}AIRI, ^{4} Universität Hamburg
V.Moskvoretskii@skol.tech, A.Panchenko@skol.tech, irina.nikishina@uni-hamburg.de

Abstract

Recent studies on LLMs do not pay enough attention to linguistic and lexical semantic tasks, such as taxonomy learning. In this paper, we explore the capacities of Large Language Models featuring LLaMA-2 and Mistral for several Taxonomy-related tasks. We introduce a new methodology and algorithm for data collection via stochastic graph traversal leading to controllable data collection. Collected cases provide the ability to form nearly any type of graph operation. We test the collected dataset for learning taxonomy structure based on English WordNet and compare different input templates for fine-tuning LLMs. Moreover, we apply the fine-tuned models on such datasets on the downstream tasks achieving state-of-the-art results on the TexEval-2 dataset.

Keywords: taxonomy construction, WordNet, hypernym prediction, LLMs

1. Introduction

Large Language Models (LLMs) are recently considered to be magic pills to every Natural Language Processing (NLP) and real-life task nowadays. People use ChatGPT[1] and other LLM-based systems for recommending books and films, retrieving encyclopedic knowledge, for language learning and teaching, grammar correction, translating, writing letters and sometimes academic papers and many other (Kasneci et al., 2023; Moskvoretskii et al., 2023). At the same time, LLMs show state-of-the-art performance on the NLP benchmarks (Song et al., 2023) and are considered to be the first approach to try.

Therefore, in this paper, we would like to challenge modern LLMs with a lexical semantic task — taxonomy learning, which requires the model to learn not only words and their meanings but also “IS-A” relations between them. Taxonomy organizes concepts into a tree structure summarizing the worldview of a human expert. Indeed, most often, such lexical-semantic resources are constructed and updated manually by highly skilled lexicographers as fully automatic construction of such resources is prone to errors. Previous approaches show that Transformer-based models do not demonstrate high-quality results (Hanna and Mareček, 2021; Radford et al., 2019), yet these did not experiment with the latest LLMs based on in-

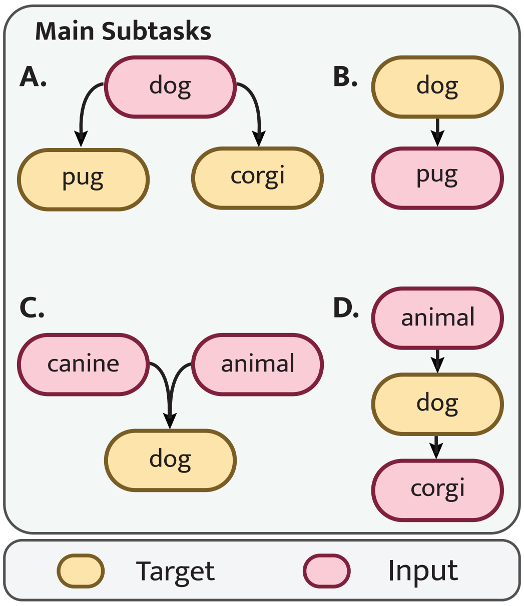

Figure 1: Examples of IS-A relation structures: (A) hyponym prediction, (B) hypernym prediction, (C) synset mixing, and (D) insertion.

[1] https://chat.openai.com

struction tuning. In this work, we aim to address this gap and try to understand if these bring the solution of the task to a new level.

Taxonomies are graph structures, where nodes are words or concepts and IS-A relations between them are denoted as edges. The most popular taxonomy for English is WordNet (Miller, 1998), which consists of synsets — lexical nodes that contain word's lemmas, definition, and sense number specifying meaning. Apart from IS-A relations, WordNet also possesses synonym and meronym relations. Taxonomies are used for various NLP tasks, such as Named Entity Recognition (Toral and Muñoz, 2006), Entity Linking (Corro et al., 2015) and others (Wang et al., 2023; Lenz and Bergmann, 2023). Even though some papers working on taxonomic structures, e.g. Nikishina et al. (2023), Chernomorchenko et al. (2024) and Nikishina et al. (2022b) do consider two-directional relations (hypernyms and hyponyms), however, none of them tackles the ability of LLMs to learn different substructures of taxonomy graph.

To sum up, the contribution of the paper is three-fold:

- we explore the capacities of LLMs to learn taxonomic structures and to predict entities to any level of taxonomy using learned representation;
- we introduce a new dataset creation method that collects different types of taxonomy-related subtasks: hypernym prediction, hyponym prediction, insertion between two existing nodes, and synset mixing, as previous setups considered only hypernym prediction;
- we test the fine-tuned models on the downstream tasks achieving state-of-the-art results on the SemEval 2016 Task-13 (Bordea et al., 2016) on the Environment dataset and provides comparable to SotA results in the Science dataset.

We also make data, code and models publicly available.[2]

2. Related Work

The most prominent directions in the field are Taxonomy Induction (Camacho-Collados et al., 2018), Hypernym Discovery (Bordea et al., 2015, 2016; Velardi et al., 2013) and Taxonomy Enrichment (Jurgens and Pilehvar, 2016; Tanev and Rotondi, 2016; Espinosa-Anke et al., 2016). There exist several studies that cover most previous approaches to taxonomy learning (Nikishina et al., 2022a, 2020; Cho et al., 2020; Takeoka et al., 2021). However, those papers do not cover more recent studies using LLMs which are the most relevant previous work for our research.

Previous methods in taxonomy construction primarily involve either sophisticated graph neural networks, such as Graph2Taxo (Shang et al., 2020), or approaches based on Hearst patterns accompanied by intricate refinement steps, like TAXI+ with Poincaré embeddings (Aly et al., 2019).

To the best of our knowledge, most existing papers do not consider generative transformers for taxonomy learning, but Encoder-based instead, like CTP (Chen et al., 2021), and others (Davies et al., 2023; Hanna and Mareček, 2021). Most existing papers describe the application of LLMs for taxonomy construction. For instance, LM-Scorer Jain and Espinosa Anke (2022) interrogate BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) among masked LMs, and GPT2 (Radford et al., 2019) among causal LMs. The authors use zero-shot taxonomy learning methods which are based on distilling knowledge from language models via prompting and sentence scoring. However, they achieve results that are lower than SotA approaches for the TexEval-2 task. However, there are no current studies that perform taxonomy learning and construction using more recent open-source models to compare with, such as LLaMA-2 (Touvron et al., 2023) and Mistral (Jiang et al., 2023).

3. Taxonomy Learning

In this Section, we describe the whole pipeline of Taxonomy Learning which comprises the

[2] https://github.com/uhh-lt/lexical_llm

algorithm for dataset creation, different model templates, fine-tuning, and evaluation. We also perform an ablation study to understand how different the performance is for widespread common knowledge words and terms.

3.1. Dataset Creation

While constructing our dataset, we primarily rely on the English WordNet 3.0 due to its clean and well-organized structure. Our predominant dependence is on the nouns subgraph, as only the most common class in the WordNet but also a difficult class for LMs to learn, according to Lazaridou et al. (2021).

Category	#Samples	
	Train	Test
Hyponym prediction	16 789	828
Synset mixing	1 461	47
Hypernym prediction	1 338	364
Insertion	648	35
Total	20 236	1 274

Table 1: The statistics of the dataset samples for Taxonomy Learning based on WordNet.

We start the dataset creation with a Directed Acyclic Graph (DAG) from WordNet which is based on the “IS A” relations. Then we randomly sample edges or subsets from the graph into different subsets, considering all possible tree operations. The detailed algorithm for the dataset construction is presented in Subsection 3.2. We assume that such a diverse dataset with various scenarios is beneficial for two reasons:

- diverse dataset will help the model to generalize better and grasp the broader relationships between words from a wider variety of subtasks;
- diverse dataset will equip the model with the capability to employ a range of strategies for constructing taxonomies.

Therefore, we collect four different subsets in order to consider the most possible tree operations within the graph, giving higher priority to hyponym and hypernym prediction. The tasks comprise the following scenarios (Figure 1):

1. **hyponym prediction** (1.A): predicting a list of hyponyms associated with the input synset from taxonomy;
2. **hypernym prediction** (1.B): predicting the hypernym based on the input word;
3. **synset mixing** (1.C): predicting single hyponym based on two synsets.
4. **insertion** (1.D): predicting a word when provided with its hypernym and hyponym.

We guarantee that there is no overlap between our test and training datasets, none of the test nodes is included in any subtask scenario. The statistics for each subset are presented in Table 1.

Algorithm 1 Dataset collection algorithm

```
Input: Sets $A, B, C, D$ sampled from Graph
Output: Train and Test Sets
Train := Empty Array
Test := Empty Array
Collect sets $A, B, C, D$.
while $(A \cup B \cup C \cup D) \neq \emptyset$ do
    cur_set $\sim \mathbb{P}_{data}$
    cur_sample = cur_set.pop()
    if cur_sample$^t$ $\overline{\cap}$ Train $= \emptyset$ then
        to_test $\sim \mathbb{P}_{test}$
        if to_test $== 1$ then
            Test.append(cur_sample)
        else
            Train.append(cur_sample)
        end if
    else
        Train.append(cur_sample)
    end if
end while
```

3.2. Dataset Collection Algorithm

To formulate a precise algorithm, we introduce subtask sets derived from the graph, represented as a collection of the following mini-sets:

$$\begin{aligned}
A_i &= \{p, \{c_j\}_{j=1}^{deg^+ p}\} \in A, \\
B_i &= \{p, c\} \in B, \\
C_i &= \{p_1, p_2, c\} \in C, \\
D_i &= \{g, p, c\} \in D,
\end{aligned}$$

Model	Hyponym	Hypernym	Insertion	Synset Mixing	Mean
GPT2	0.006	0.033	0.018	0.027	0.021
Llama2-7B Numbers	0.099	0.267	0.262	**0.239**	0.162
Llama2-7B Lemmas	**0.127**	0.293	0.329	0.218	0.188
Llama2-7B Definitions	0.123	0.494	**0.436**	0.234	**0.247**
Mistral-7B Definitions	0.085	**0.498**	0.436	0.160	0.218

Table 2: Fine-tuned models MRR scores on the test set. Bold represents the best result, underlined are second-ranked

Here, c dennotes hyponyms, p - hypernyms, and g - hyperhypernyms.

In order to perform comprehensive set intersections, we introduce the concept of "deep intersection", denoted as $\overline{\cap}$. This operation characterizes the intersection between the elements of elements from two sets, not solely the elements themselves, expressed as: $S_1 \overline{\cap} S_2 = \bigcup_{ij}(S_{1i} \cup S_{2j})$

In the following phase, our objective is to create random training and testing sets, ensuring around 1000 samples in the test set and a predominant number of hyponyms predictions and hypernyms prediction in the training set, with other sample types evenly distributed. This task is complex due to possible large intersections among different cases and the order of sample collection. To address this, we introduce a distribution on subtasks denoted as $\mathbb{P}_{data}$, allowing us to manually adjust the probability of sampling each subtask.

We also introduced a Bernoulli distribution $\mathbb{P}_{test}$ with a parameter p to control the probability of samples being assigned to the test set. Optimal values for these probabilities were determined as follows:

For $\mathbb{P}_{data}$: $P(A) = 0.51$, $P(B) = 0.39$, $P(C) = 0.05$, and $P(D) = 0.05$.

For $\mathbb{P}_{test}$: $p = 0.05$ and $q = 0.95$.

During collection, we use the "pop()" operation, that deletes last element from set and returns it.

To handle the intricacies of the prevalent word categories, we employ a topological sort on the graph. Subsequently, we ensure that no vertex within our sets possesses a level lower than a specified parameter denoted as "level". This condition can be expressed as follows: $\forall i, S \quad \forall v \in S_i : \quad TopSort(v) \geq level$. In our collected data $level = 3$

We also establish a "target" vertex for each element within the subtasks. This allows us to track the presence of this specific target vertex in the test set, ensuring that we maintain the integrity of our evaluation.

The breakdown of the definitions for these "target" vertices based on different subtasks can be described as follows:

- $A_i^t = \{c_j\}$: In this case, we need to track all the hyponyms. If we have not encountered hyponyms in the training set, then we cannot determine the target. However, it is permissible to encounter the hypernym in the test set because it is present in the prompt.
- $B_i^t = c$: If we have not seen the hyponym, it means we have not encountered this pair. Otherwise, we would have added the hyponym to the tracking. Therefore, if we haven't seen the hyponym, it implies we haven't seen this edge.
- $C_i^t = c$: If we have not seen the hyponym, it implies we haven't observed the target.
- $D_i^t = p, c$: This scenario is equivalent to restricting two edges: $g{-}p$ and $p{-}c$, which correspond to the cases A and B.

3.3. Model Finetuning

For our research, we utilize latest foundation models language models Llama2-7B and Mistral-7B. Smaller models, such as GPT2, demonstrated negligible performance across all subsets and were consequently excluded from the analysis. These models were optimized using a 4-bit quantization technique. We

	Hyponym	Internal Nodes	Leaves Divided	Only Leaves	Single Leaves	Insertion	Hypernym	Synset Mixing	Mean
Numbers	-0.036	0.001	-0.046	-0.240	0.006	0.079	-0.055	0.089	-0.005
Lemmas	-0.036	-0.019	-0.004	-0.124	-0.016	-0.028	-0.053	0.120	-0.001
Definition	-0.042	-0.044	0.029	-0.194	0.000	0.025	0.006	0.097	0.054

Table 3: MRR Scores difference between easy and hard subsamples (easy$-$hard) for the taxonomy learning subtasks. Green color denotes that scores are higher for the "easy" subset, Red color shows that better results are for the "hard" subset.

further fine-tune them with LoRA (Hu et al., 2022) for one training epoch, using a batch size of 64. We employ the AdamW optimizer with a learning rate of $3e-4$ and a cosine annealing scheduler.

Our inputs include an LLaMA-2 system prompt that looks as follows:

(1) [INST] «SYS» You are a helpful assistant. List all the possible words divided with a comma. Your answer should not include anything except the words divided by a comma «/SYS»

Then we introduce a technical-style input prompt and the expected output format:

(2) hypernym: dog.n.1 | hyponyms: [/INST]

(3) pug, corgi,

We also explore the impact of altering the style of the prompt with numerical representations, lemmas, and definitions: *"dog.n.1"*, *"dog (dog, domestic dog, Canis familiaris)"*, *"dog (a member of the genus Canis that has been domesticated by man since prehistoric times)"*.

3.4. Results

In our study, we assess the quality of our models using Mean Reciprocal Rank (MRR), which reflects the position of the first correct answer. We do not use other possible metrics for ranking as they might be too strict.To evaluate the models, we generate a list of potential candidates, separated by commas, and match them with target words.

As our preliminary experiments, we also conducted a case study to assess ChatGPT performance on the task. We discovered that it failed to provide correct answers, even when employing the few-shot learning technique. For example, for the word "Maltese" candidates from ChatGPT are "dog breed", and "animal" instead of "toy dog" which is the correct hypernym from WordNet; For the phrase "machine translation" possible hypernyms are "automated translation" and "language translation system", while true hypernyms are "artificial intelligence" and "computational linguistics". We can see that the model correctly identifies the area, but is not able to point out the specific synset from WordNet. This was particularly notable in instances where our fine-tuned model excelled.

The results for our fine-tuned models are presented in Table 2. We observe that the best results for hypernym prediction and insertion between the two nodes are quite high. For example, the score of 0.5 for hypernym prediction means that on average the second predicted candidate is the correct one. However, from the manual error analysis, we observe that this score is compiled as the mean of the correctly predicted first candidates for most cases and all incorrect candidates for others. At the same time, we can also see that the results for synset mixing are twice lower than hypernyms or insertion. Quite low scores are achieved for hyponym prediction. Notably, those tasks appear to be more difficult to solve. We hypothesize that the limitations for hyponym prediction may not stem from the amount of data or the model, but rather from the size of the model, which is generally believed to be closely linked to its performance. Initially, we theorized that these limitations could be mitigated through the use of disambiguation via lemmas or definitions. However, our findings suggest that this may not be effective. Additionally, these limitations might arise from the inherent nature of relational and instructional tuning. In cases where there is only a single parent, predictions tend to be more straightforward, and the model is trained to predict a single node. This contrasts with the scenario involving hyponyms, where multiple instances exist. Consequently, the model must predict a sequence, and the loss is calculated across the entire sequence in its precise order.

We can also note that incorporating lemmas yields significantly better results compared to numbers, and the highest scores are achieved when definitions are included. This might happen due to the autoregressive generation. Immersing the model in an appropriate context leads to shifting distribution towards correct answers. In that way, providing definitions makes the shift either stronger or more accurate. We also tested the best setup with the brand new Mistral-7B model which showed higher performance than LLaMA-2 on (Jiang et al., 2023), however, it did not perform better on our dataset.

3.5. Ablation Study

We were surprised to find that LLaMA-2 performed poorly in predicting hyponyms. To better understand the results, we investigate hyponym predictions more thoroughly.

3.5.1. Subtypes of Hyponyms

First, we assume that the results demonstrated on all types of hyponym relations might be not very representative. Therefore, we split the hyponym prediction task into the following subtasks regarding the type of the predicting nodes (See Figure 2 for examples and more detail):

- **Leaves Divided** (2A): all hyponyms are terminal nodes, and 50% of them appear in the input, other part is predicted as a target.
- **Internal Nodes** (2B): Hyponyms are not required to be terminal nodes, but they have to contain at least one internal node.
- **Only Leaves** (2C): all target hyponyms are terminal nodes;
- **Single Leaves** (2D): hyponyms are terminal, and they are the only hyponyms for the node.

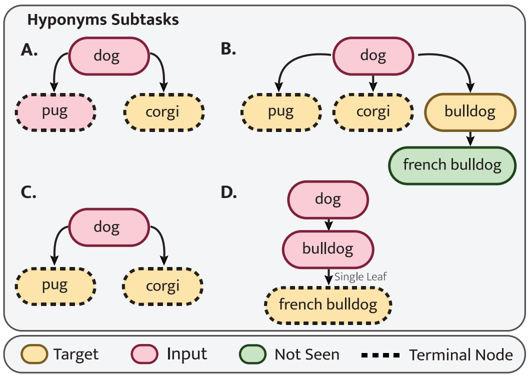

Figure 2: Examples of hyponym subtasks: Leaves Divided (A), Internal Nodes (B), Only Leaves(C), Single Leaves (D).

We further evaluate the model performance, and present the results in Table 4. LlaMA-2 excels in predicting terminal nodes (2C) compared to internal ones (2B). From the manual error analysis, we can conclude that terminal nodes are usually non-ambiguous and have only one meaning. While predicting internal nodes (2B) the model predicts more subsequent nodes (with hop≥ 2) instead of the direct hyponyms. We can also see that (2A) scenario demonstrates lower performance than while predicting all possible hyponyms (2C). This implies that the core issue is not ambiguity, which additional hyponyms can address, but rather the inability to generate appropriate hyponyms. The predictive scope of the model is limited with the present candidates in the input. The scenario featuring a single leaf hyponym (2D), is extremely challenging to predict, even with hyperhypernyms as input. It may be explained by the fact that such instances are more complex and less prevalent in the language.

3.5.2. Common Words VS Terminology

When analyzing the outputs of the models in order to understand low results on average, we notice that the major problem may be in the difficulty of the dataset itself: some synsets in the WordNet taxonomy might be too specific for the model to predict hyponyms or hypernyms for.

Model	2A	2B	2C	2D
#Samples	*117*	*115*	*110*	*486*
Numbers	0.152	0.113	0.220	0.068
Lemmas	**0.179**	0.154	0.220	**0.100**
Definition	0.175	**0.163**	**0.268**	0.081

Table 4: MRR scores for the LlaMA-2 model with a different hyponyms prediction subtasks, with column names correspond to Figure 2.

In order to check this hypothesis, we manually split our dataset into two categories: common knowledge words (*"easy"* category) and terms, jargon, or rare words (*"hard"* category). To do that, we asked three annotators, experts in computational linguistics, to annotate the test set. The assessors were required to mark the whole sample as *"hard"* if there was at least one word that belonged to terms, jargon, or rare words, otherwise, they were supposed to put an *"easy"* label. Krippendorf's alpha score on the annotations reached 0.67, indicating sufficient agreement among annotators to take answers into consideration.

We recalculate the performance on both subsets and present the results in Table 3. Surprisingly, our models tend to perform better on the "hard" instances, particularly when predicting hyponyms. However, for the best model that uses word definitions, "easy" instances achieve higher scores, particularly for the cases that do not involve hyponyms. This pattern, however, doesn't consistently hold for other types of prompts, where "hard" instances are sometimes predicted more accurately, even for hypernyms or internal nodes.

We believe that the outcomes of the current study demonstrate that the model more correctly predicts less common words. This may be explained by the fact that the distribution of candidates for the terms is narrower, which makes it more focused on the correct answers. Moreover, the model encounters such rare words quite infrequently and usually in a consistent and specific context.

4. Downstream Task: TexEval-2 (SemEval 2016 Task 13)

In order to check model abilities to generalize and to learn different strategies of taxonomy creation, we test the fine-tuned models on the downstream task: SemEval 2016 Task 13. We use the Eurovoc taxonomies ("Science" and "Environment") from SemEval-2016 (Bordea et al., 2016). These datasets are commonly used as a benchmark for testing models' abilities of taxonomy construction.

4.1. Taxonomy Construction Procedure

To create the taxonomy, we use perplexity to discover edges between nodes. First, we calculate perplexity for all vertex pairs using two input templates: hyponym prediction 1A and hypernym prediction 1B. After that, we construct the taxonomy by adding edges between vertices with perplexity below a certain threshold. This was done either via considering all possible word pairs (brute-force) or by recursively building the taxonomy from a starting point (root for hyponyms prediction), like a tree (Depth-first search style).

4.2. Results and Discussion

Our experiments show that predicting hypernyms performs significantly better than predicting hyponyms, which is coherent with the scores for the respective subtasks during the fine-tuning step. Furthermore, the brute-force method of building the taxonomy outperformed the DFS-style approach. That could happen due to error accumulation during graph traversal. Incorrect decision on the first couple levels significantly limits our possible edge space. The results for the additional experiments are presented in Table 6.

Table 5 presents the F1-score results for the Science (Sci) and Environment (Env) datasets. We compare our three best-performing models with the previous approaches and the GPT-2 baseline. We deliver results for LlaMA-2 with numerical input and LlaMA-2 with lemmas. For

Model	Sci	Env
TexEval-2 best	0.313	0.300
TAXI+ Aly et al. (2019)	0.414	0.309
Graph2Taxo pure Shang et al. (2020)	0.390	0.370
Graph2Taxo best Shang et al. (2020)	**0.470**	0.400
CTP Chen et al. (2021)	0.291	0.230
LM-Scorer Jain and Espinosa Anke (2022)	0.318	0.264
GPT-2	0.014	
LlaMA-2 with lemma	0.419	0.409
LlaMA-2 with empty lemma	0.426	0.380
LlaMA-2 with numbers	0.416	**0.411**

Table 5: Comparison of the results for the downstream TexEval-2 task.

Approach	Method	Template	Sci	Env
LlaMA-2 with lemma	brute-force	hyper	0.419	0.409
		hypo	0.192	0.115
	dfs	hyper	0.340	0.213
		hypo	0.137	0.142
LlaMA-2 with empty lemma	brute-force	hyper	**0.426**	0.380
		hypo	0.188	0.116
	dfs	hyper	0.338	0.213
		hypo	0.127	0.129
LlaMA-2 with numbers	brute-force	hyper	0.416	**0.411**
		hypo	0.185	0.116
	dfs	hyper	0.186	0.186
		hypo	0.125	0.138

Table 6: Results for the downstream TexEval-2 task comparing different fine-tuned models, methods for graph construction, and templates for model inputs. Hyper approach stands for hypernym prediction and hypo for hyponym prediction

LlaMA-2 with lemmas (as we have no additional lemmas unlike in WordNet), we tried two approaches (duplicate lemma in listing; provide no lemma at all):

(4) “hypernym: cat (cat) | hyponyms:”

(5) “hypernym: cat () | hyponyms:”

Our results show that our method performs better than all other existing models on the Environment dataset and is ranked second on the Science dataset. However, the best-performing approach for “Science”, which is Graph2Taxo (Shang et al., 2020) is reached with a GNN-based cross-domain transfer framework. The best score is achieved during their ablation study. The default setup of the framework does not achieve the best scores (see (Shang et al., 2020) (pure) in Table 5). Moreover, we need to take into account that we did not apply any specific taxonomy-building strategy, which leaves room for further improvement on the taxonomy-creation downstream tasks. At the same time, GPT-2 performed extremely bad on this task, as well as zero-shot methods based on distilling knowledge from language models via prompting and sentence scoring (Jain and Espinosa Anke, 2022), and the pretrained language models (CTP) like BERT for parenthood prediction and tree reconciliation (Chen et al., 2021).

5. Conclusion

Overall, our primary task was to investigate whether Large Language Models are capable of solving purely linguistic tasks, such as Taxonomy Learning. We can conclude that the models do acquire basic skills in different types of taxonomic operations: insertion, node mixing, hypernym, and hyponym prediction. However, the results are far from being perfect on the test split, which demonstrates model confusion on the task. Surprisingly, the model struggles more with common words than with terms. At the same time, the results for internal nodes are lower than for terminal ones. The above-mentioned outcomes and the best scores achieved by the model with definitions demonstrate that the ambiguity problem is still relevant even for LLMs when a small context is given. When considering the downstream task, we can conclude that the fine-tuned LLMs do learn taxonomic relations and could be further used for different applications. For example, we demonstrated that our finetuned LlaMA-2 achieves state-of-the-art results for the TExEval-2 task of taxonomy constructon on the “Science” task. However, the application of such models might still require a more elaborate procedure for taxonomy creation. As for future work, we plan to extend the taxonomy to other languages using Open Multilingual WordNet and do further experiments with input structures and downstream tasks. We believe that the issue could potentially be mitigated either by considering larger models or by modifying the training procedure. To improve the results on the hyponym prediction, we plan to modify the training procedure, involving permuting the sequence of hyponyms and conducting multiple training steps on the same relations, or altering the target to focus on a single hyponym and sampling them in portions.

Limitations

We find the following limitations of our work:

- The full list of operations over taxonomy might also include deletion, moving a synset from one position to another one in a tree. But we do not consider them to assume the taxonomy is “perfect” as input taxonomy is built by humans. However, for automatically constructed taxonomies such operations are essential to use, as some parts of the tree/graph may be not optimally constructed.

- We expect that it is possible to further push the quality reported in our work if larger versions of large pre-trained transformers are used, such as LLaMA2-13B and Vicuna-13b, as was the case for multiple other tasks. However, the general trend is clear from our experiments.

- We are also aware of new experiments from a very recent paper (Chen et al., 2023) where authors present a comparative study for taxonomy construction using LLMs (GPT-NEO and GPT-3.5 for few-shot learning), evaluating on two datasets, different from TexEval-2. Because of the time constraints, we were not able to test our model on their downstream dataset.

- We did not test the multilingual setting of our approach, which is possible if the multilingual version of sequence-to-sequence models and datasets are used. However, preliminary experiments demonstrated negative results and a very low quality of the existing multilingual taxonomies (BabelNet, ConceptNet, Open Multilingual WordNet) (Navigli and Ponzetto, 2010; Speer et al., 2018) for languages distinct from English. This is an important additional experiment to further validation of the method explored in our work.

- Nowadays, dozens of large pre-trained generative models exist and we report results only on a few of them. It may be that some other base models used could further push the results. Our goal however was to show an example of how similar models and not perform an exhaustive search of all models.

- We tried to be exhaustive, but we might not have covered all existing types of taxonomy-related subtasks, which we leave out of the scope of our research.

Ethics Statement

We use in our work large neural models, such as LlaMA-2, pre-trained on real texts including user-generated content. While authors of the models made an effort to filter obviously toxic or biased content, the model itself still can contain certain biases, and as a consequence outputs of our methods may render such biases. Methodologically it is however straightforward to apply our techniques on other pre-trained models that were debiased in a required way. Otherwise, we do not see any other ethical concern in our work to the best of our knowledge.

Acknowledgements

This work was supported by the DFG through the project “ACQuA: Answering Comparative Questions with Arguments” (grants BI 1544/7-1 and HA 5851/2- 1) as part of the priority program “RATIO: Robust Argumentation Machines” (SPP 1999).

The research by Alexander Panchenko was supported by the RSF grant 20-71-10135.

6. Bibliographical References

Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019. Every child should have parents: a taxonomy refinement algorithm based on hyperbolic term embeddings.

He Bai, Tong Wang, Alessandro Sordoni, and Peng Shi. 2022. Better language model with hypernym class prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Dublin, Ireland. Association for Computational Linguistics.

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

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

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

Boqi Chen, Fandi Yi, and Dániel Varró. 2023. Prompting or fine-tuning? A comparative study of large language models for taxonomy construction. *CoRR*, abs/2309.01715.

Catherine Chen, Kevin Lin, and Dan Klein. 2021. 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.

Polina Chernomorchenko, Alexander Panchenko, and Irina Nikishina. 2024. Leveraging Taxonomic Information from

Large Language Models for Hyponymy Prediction. In *Analysis of Images, Social Networks and Texts — 11th International Conference, AIST 2023*, volume 14486 of *LNCS*. Springer.

Yejin Cho, Juan Diego Rodriguez, Yifan Gao, and Katrin Erk. 2020. Leveraging WordNet paths for neural hypernym prediction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3007–3018, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. FINET: context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 868–878. The Association for Computational Linguistics.

Adam Davies, Jize Jiang, and ChengXiang Zhai. 2023. Competence-based analysis of language models. *CoRR*, abs/2303.00333.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Luis Espinosa-Anke, Francesco Ronzano, and Horacio Saggion. 2016. TALN at SemEval-2016 task 14: Semantic taxonomy enrichment via sense-based embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1332–1336, San Diego, California. Association for Computational Linguistics.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Devansh Jain and Luis Espinosa Anke. 2022. Distilling hypernymy relations from language models: On the effectiveness of zero-shot taxonomy induction. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 151–156, Seattle, Washington. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

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.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and

Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, Kris Cao, Tomás Kociský, Susannah Young, and Phil Blunsom. 2021. Pitfalls of static language modelling. *CoRR*, abs/2102.01951.

Mirko Lenz and Ralph Bergmann. 2023. Case-based adaptation of argument graphs with wordnet and large language models. In *Case-Based Reasoning Research and Development*, pages 263–278, Cham. Springer Nature Switzerland.

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.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Viktor Moskvoretskii, Frolov Anton, and Kuznetsov Denis. 2023. Imad: Image-augmented multi-modal dialogue.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Irina Nikishina, Polina Chernomorchenko, Anastasiia Demidova, Alexander Panchenko, and Chris Biemann. 2023. Predicting terms in IS-a relations with pre-trained transformers. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 134–148, Nusa Dua, Bali. Association for Computational Linguistics.

Irina Nikishina, Varvara Logacheva, Alexander Panchenko, and Natalia Loukachevitch. 2020. Studying taxonomy enrichment on diachronic WordNet versions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3095–3106, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Irina Nikishina, Mikhail Tikhomirov, Varvara Logacheva, Yuriy Nazarov, Alexander Panchenko, and Natalia V. Loukachevitch. 2022a. Taxonomy enrichment with text and graph vector representations. *Semantic Web*, 13(3):441–475.

Irina Nikishina, Alsu Vakhitova, Elena Tutubalina, and Alexander Panchenko. 2022b. Cross-modal contextualized hidden state projection method for expanding of taxonomic graphs. In *Proceedings of TextGraphs-16: Graph-based Methods for Natural Language Processing*, pages 11–24, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Linxin Song, Jieyu Zhang, Lechao Cheng, Pengyuan Zhou, Tianyi Zhou, and Irene Li. 2023. Nlpbench: Evaluating large language models on solving NLP problems. *CoRR*, abs/2309.15630.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge.

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.

Hristo Tanev and Agata Rotondi. 2016. Deftor at SemEval-2016 task 14: Taxonomy enrichment using definition vectors. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1342–1345, San Diego, California. Association for Computational Linguistics.

Antonio Toral and Rafael Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

Xiang Wang, Yanchao Li, Huiyong Wang, and Menglong Lv. 2023. Mkbqa: Question answering over knowledge graph based on semantic analysis and priority marking method. *Applied Sciences*, 13(10).