LLMs4OL 2024: The 1st Large Language Models for Ontology Learning Challenge at the 23rd ISWC
LLMs4OL 2024 Task Participant Papers
https://doi.org/10.52825/ocp.v4i.2483

Published: 02 Oct. 2024

SKH-NLP at LLMs4OL 2024 Task B: Taxonomy Discovery in Ontologies Using BERT and LLaMA 3

Seyed Mohammad Hossein Hashemi[1], Mostafa Karimi Manesh[1], and Mehrnoush Shamsfard[1]

[1]Shahid Beheshti University, Tehran, Iran

seyedmo.hashemi@mail.sbu.ac.ir, {m_karimimanesh,m-shams}@sbu.ac.ir

*Correspondence: Seyed Mohammad Hossein Hashemi, seyedmo.hashemi@mail.sbu.ac.ir

Abstract: Taxonomy discovery in ontologies refers to extracting the parent class from the child class. By modeling this task as a classification problem, we addressed it using two different approaches. The first approach involved fine-tuning the "BERT-Large" model with various prompts and using it in a classification system. In the second approach, we utilized the "LLaMA 3 70B" model, experimenting with different prompts and modifying them to achieve the best results. Additionally, we evaluated the correctness of the answers using substring and Levenshtein distance functions. The results indicate that, with appropriate fine-tuning, the BERT model can achieve performance levels comparable to those of more recent and significantly larger language models, such as LLaMA 3 70B. However, with appropriate prompts, LLaMA 3 70B performs slightly better than BERT, highlighting the importance of prompt quality. Ultimately, further experiments on different settings for fine-tuning BERT, few-shot learning, and using knowledge graphs for validating the model's answers for LLaMA are recommended to improve the results. Additionally, testing other models and examining the results of various encoder-based and decoder-based models can be employed.

Keywords: Large Language Models, LLMs, Ontologies, Ontology Learning, Fine-tuning, Prompting, Prompt-based Learning, BERT, LLaMA 3

1 Introduction

One of the applications of large language models (LLMs) is learning ontologies from input text. This process is divided into three sections: *term typing*, *taxonomy discovery*, and *non-taxonomic relationship extraction* [1]. In the LLMS4OL challenge [2], the goal is to develop models to perform these tasks automatically. The task selected by the authors of this paper is Task B, i.e., taxonomy discovery. Among the available datasets in the challenge [3], the GeoNames dataset was chosen for this project. This dataset is extracted from the GeoNames ontology [4], which is a geographical database that provides information about locations and geographical features around the world. This data includes place names, geographic coordinates (latitude and longitude), place types (such as city, village, river, mountain, lake, etc.), elevation above sea level, postal codes, population, and other attributes. Among the features available in this ontology, the place name along with its type has been extracted.

For solving this problem, the language models BERT-Large and LLaMA 3 70B are being used. The following reasons were involved in choosing these two models:

- BERT performs very well in many traditional NLP tasks such as classification and information extraction.
- In the studies of BabaeiGiglouet al.[1], BERT was able to achieve remarkable results without the need for fine-tuning, with its results in this task being close to other mentioned LLMs like GPT-3.
- Fine-tuning BERT model was doable for the authors of the article given their hardware access.
- LLaMA 3 70B has a larger size and more modern architecture comparing to BERT.

As shown by BabaeiGiglou et al. [1], in Task B with the GeoNames dataset, the best result was achieved by the GPT-3.5 model, with an F1-score of 67.8. Among the fine-tuned models, FLAN-T5 Large obtained the best result with an F1-score of 62.5. The BERT and LLaMA 7B models also reached scores of 54.5 and 33.5, respectively, without fine-tuning. However, these scores were based on the model’s performance in binary classification, determining whether there is a relationship between a child and parent or not. In contrast, the task in this project is to identify the parent of each child.

In this project, the competition training dataset was first received, and additional necessary data was generated. Then, with the help of two different approaches using the mentioned language models, various methods were presented for solving the problem. The final results of each method and their analysis are discussed, and finally, ideas for improving these models are provided.

2 Augmentation of Training Data

The dataset provided to participants for this challenge includes 476 records of (child, parent) pairs from the GeoNames ontology. In this dataset, the parent column contains 9 distinct values; therefore, this dataset can be considered a classification dataset with 9 classes. To train a classifier using BERT, in addition to this dataset, we also needed a negative dataset, which we generated through the procedure described below.

From the 476 records in the initial dataset, 76 records were separated for validation to ensure no overlap between the training and validation datasets. Then, using a consistent pattern, negative data was generated for both the training and validation datasets. Two approaches were considered for generating negative data:

1. Generating reversed records: Reversed records are records that are exactly copied from the positive dataset, with the parent and child swapped.
2. Generating manipulated records: Manipulated records are also exactly copied from the positive dataset, with one of the other 8 parents randomly replacing the original parent in each record.

The number of records in the generated negative dataset for each dataset equals the number of records in the original dataset. Approximately one-third of the negative dataset consists of reversed records, and two-thirds are manipulated records. For example, for a set of 400 positive records in the training data, 133 reversed records and 267 manipulated records were generated. In the final dataset, we have a set of records where the positive or negative status is indicated by a column titled “label” that can be True or False.

3 Proposed Methods for Taxonomy Discovery

As mentioned earlier, two different approaches were used to solve this problem. One was fine-tuning BERT, and the other was using the LLaMA 3 70B model. Details of each approach are explained below. Before continuing, since the discussed problem can also be considered a classification problem, from here onward, the concepts of class and parent (destination of the is-a relationship) and also instance and child (source of the is-a relationship) are considered synonymous. Additionally, since the main focus is on identifying the class (in classification problem) or the parent (in ontology hierarchy), any mention of class refers to the parent and vice versa.

3.1 BERT-Based Approach

To solve this problem using BERT, we modeled it as a multi-class classification task with 9 classes. The method involves using a single binary classifier iteratively for each of the 9 classes. The classifier determines whether an instance belongs to a specific class or not by receiving the instance and class as inputs. For example, when predicting a child’s parent, the classifier first determines if the child belongs to class 1 or not, then class 2, and so on for all subsequent classes.

Other approaches could have been applied, such as using a separate classifier for each class or employing a single 9-class classifier for all classes. Our method in using a single binary classifier is scalable to larger datasets and a greater number of classes without requiring the training of multiple models or designing a separated multi-class classifier when the classes are changed.

We used the BERT-Large model to solve this problem. Initially, this model was fine-tuned on the training dataset as a binary classifier. This means that the final model can determine whether there is an is-a relationship between the given (parent, child) pair or not. To use this model for a 9-class classification problem, we need to check whether a child belongs to a parent once for each parent, and based on the output, the relevant class is extracted. Details of this method are explained below.

During the fine-tuning and testing of the models, an additional prompt (the ninth in the following list) was added to the 8 prompts used by BabaeiGiglou et al. [1], resulting in a total of 9 prompts. The complete set of prompts is as follows:

1. parent is the superclass of child. This statement is [MASK].
2. child is a subclass of parent. This statement is [MASK].
3. parent is the parent class of child. This statement is [MASK].
4. child is a child class of parent. This statement is [MASK].
5. parent is a supertype of child. This statement is [MASK].
6. child is a subtype of parent. This statement is [MASK].
7. parent is an ancestor class of child. This statement is [MASK].
8. child is a descendant class of parent. This statement is [MASK].
9. ”parent” is the superclass of ”child”. This statement is [MASK].

In these prompts, *parent* and *child* are replaced with the appropriate parent and child, and *[MASK]* is the token that the model needs to predict. In this set, there are 4 superclass statements, 4 corresponding subclass statements, and 1 additional superclass statement grouped together. This set of prompts has been used in different ways to fine-tune BERT, which will be discussed in section 4.1.

To determine the parent class for a child class, we initially ask the model nine questions, each corresponding to one of the potential parent classes. These questions are posed in the format of the first prompt. If the model answers ”True” to only one of these nine questions while answering ”False” to the remaining eight, the parent class associated with the ”True” response is selected. If no single parent class stands out, we proceed with two additional prompts in sequence. At each stage, only parents with the highest score, meaning those for which the model has returned True for more prompts, advance to the next stage. This process continues until the end of the prompt list. In the final step, if multiple parent classes still have the highest score, the system randomly selects one from these parents. The percentage of instances where random selection was used relative to the total number of instances can be a criterion for evaluating different systems.

3.2 LLaMA-Based Approach

After evaluating Task B using the fine-tuned Bert-Large model, it was decided to perform the evaluation using the LLaMA 3 70B as well. In this phase, the focus was mainly on the prompts. The general structure of the prompts follows two main concepts:

1. Classification Concept (instance and class)
2. Hierarchy Concept (is-a) (parent and child)

In the first category, the prompts contain a classification definition, asking the model to identify the class based on the given input. However, in the second category, the problem is defined as an is-a hierarchy, and the model is asked to identify the destination of the relationship (parent) based on the input (child).

After observing the model’s responses, one challenge identified was the class names. Each class title is a combination of several terms (e.g., ”mountain, hill, rock”). Despite mentioning the class titles in the prompts, in some cases, the model only used part of the class title in its responses. For example, in response to the question about ”cattle dipping tank,” which corresponds to the class ”spot, building, farm,” the model only used ”spot” as the answer. Given these conditions, during the evaluation phase, in addition to evaluating the model’s output separately, the substring function and Levenshtein distance [5] were applied to the model’s output.The substring function returns the class title that the output of the model is a part of. The Levenshtein function returns the class title that is closer to the output of the model based on the Levenshtein distance.

In addition to the mentioned actions, to save time, instead of providing samples one by one, a set of samples formatted in a specific way was fed to the model, and responses were received in batches. To manage this issue, each sample was assigned a unique number, and the model was asked to separate the response sections and include the number of each question alongside its response. Subsequent results indicate that batch questions are not as accurate as individual questions.

4 Experiments

In this section, we present the experiments conducted on the two groups of systems discussed: BERT-based systems and LLaMA-based systems. Implementations and datasets used in these experiments are available in a GitHub repository[1].

4.1 Experiments on the BERT-Based Systems

In this section, we examine the details of the systems implemented using BERT. All systems are fine-tuned using the methods mentioned in the previous section, with the goal of predicting the parent of each child. Due to time and hardware constraints during the competition, the BERT model was fine-tuned using fixed hyperparameters. Hyperparameter tuning could potentially lead to improved results.

The first category of our systems consists of those where BERT was trained sequentially with 1, 2, 3, ... up to 8 different prompts, with one epoch of training for each prompt. Since the terms *ancestor* and *descendant* used in prompts 7 and 8 differ somewhat from those used in the other prompts, two more systems were trained separately: one on the set of prompts 1 to 4 and 7, and another on prompts 1 to 4, 7, and 8. For the testing phase of all the aforementioned models, 9 prompts were used.

By analyzing the results and performance of the systems, and considering the functioning and structure of BERT, we hypothesized that using one set out of superclass statements and subclass statements could help improve the results. For this purpose, in the next category of systems, BERT was trained only on superclass statements, as follows: once with prompt 1, once with prompts 1 and 3, once with prompts 1, 3, and 5, etc. Each prompt is being trained for one epoch. For the testing phase of these models, the same 5 prompts are used.

Table 1 presents the results of the different systems on the validation dataset. As mentioned, in each system, the predicted parent in some instances was randomly selected from among the candidate parents. The last column of this table indicates the percentage of parents that were *not randomly* selected. The metric values are reported in percentage. These values are rounded to one decimal place in all columns except the last one, where they are reported without decimal places. Additionally, weighted averages were used in calculating precision, recall, and F1-score. In this table, the best result in each column is bold and underlined, and the second and third best results in each column are bold.

4.2 Experiments on the LLaMA-Based Systems

The initial results using the prompts on the evaluation dataset, which was submitted for the competition, are presented in Table 2. The values are rounded to one decimal place in all columns.

In both prompts, the classification problem and the is-a relationship were defined precisely:

- "The problem under consideration is classification. X is a subclass of Y, meaning that X shares common features and properties with other members of class Y."
- "If we say "X is a Y," it means that X is a specific instance of Y and inherits all the features and behaviors of Y."

[1] https://github.com/s-m-hashemi/llms4ol-2024-challenge

Table 1. *Evaluation results of different BERT-based systems on validation dataset.*
**SC: Superclass Statements*

No	**Model**	**Precision**	**Recall**	**F1-Score**	**% of Non-Randoms**
1	Prompt 1	6.6	18.4	8.4	71
2	Prompts 1, 2	56.8	22.4	21.9	**95**
3	Prompts 1-3	44.7	40.8	38.8	**92**
4	Prompts 1-4	54.2	25.0	23.5	**95**
5	Prompts 1-5	53.8	44.7	44.3	49
6	Prompts 1-6	45.2	34.2	34.4	66
7	Prompts 1-7	**67.5**	**61.8**	**63.0**	54
8	Prompts 1-8	37.0	25.0	23.1	53
9	Prompts 1-4, 7	63.0	0.5	**52.9**	62
10	Prompts 1-4, 7, 8	**64.7**	18.4	12.0	**83**
11	SC* Prompt 1	8.9	19.7	10.1	71
12	SC Prompts 1, 2	7.2	21.1	10.3	70
13	SC Prompts 1-3	50.5	**51.3**	45.6	32
14	SC Prompts 1-4	42.2	50.0	45.0	29
15	SC Prompts 1-5	**66.2**	**59.2**	**60.8**	50

Table 2. *Evaluation results of LLaMA 3 70B tested using different prompts on validation dataset.*

Prompt / Eval metrics	**Precision**			**Recall**			**F1-Score**		
Extra function	**None**	**Sub**	**Levn**	**None**	**Sub**	**Levn**	**None**	**Sub**	**Levn**
Class concept	64.9	64.8	57.1	51.3	51.3	51.3	54.6	54.6	51.9
Is-a (individual query)	21.4	72.6	47.2	7.8	64.4	21	10.1	62.9	18.7
Is-a (batch query)	0	68.4	16.8	0	39.4	13.1	0	46.4	8.2

An example of the prompts is as follows:

- "If we say 'X is a Y,' it means that X is a specific instance of Y and inherits all the features and behaviors of Y. Given an instance as 'X,' select the most appropriate 'Y' from (city, village) or (country, state, region) or (forest, heath) or (mountain, hill, rock) or (parks, area) or (road, railroad) or (spot, building, farm) or (stream, lake) or (undersea)."

In order to improve the results, several iterations of modifying the prompt definitions were undertaken, leading to improved outcomes, which are detailed below.

In the initial prompts, a sample was included to clarify the definition. For example, in the classification prompt, it was stated: "wadi mouth" is considered a subclass of "parks, area." We observed that the model tended to favor the mentioned class. Based on this observation, this example was removed from the new prompts. Furthermore, the class names, due to their specific structure and the presence of commas between them, needed to be more precisely distinguished. Therefore, each class title was enclosed in a pair of parentheses, and the term "or" was used between them.

In the initial prompt, it was written: "In lexical networks, a concept known as a triplet is discussed. This triplet is formed between two words and a relationship between them." However, in the improved prompt, the definition was changed to: "If we say 'X is a Y,' it means that X is a specific instance of Y and inherits all the features and behaviors of Y." The results with the evaluation data using the modified prompts are presented in Table 3. The values are rounded to one decimal place in all columns.

Table 3. *Evaluation results of LLaMA 3 70B tested using improved prompts on validation dataset.*

Prompt / Eval metrics	Precision			Recall			F1-Score		
Extra function	**None**	**Sub**	**Levn**	**None**	**Sub**	**Levn**	**None**	**Sub**	**Levn**
Class concept	75.8	75.7	73.4	64.4	71	67.1	67.7	72	67.7
Is-a	76.2	76	73.8	65.7	72.3	68.4	69.1	73.1	68.9

By examining the results of batch and individual submissions, it was found that the results in the batch mode were weaker, so in this phase, batch question evaluations were not conducted.

4.3 Results of the Systems on the Test Dataset

The results of the BERT-based and LLaMA-based systems on the final test dataset are presented in Table 4. This table includes the results of the three best BERT-based and two best LLaMA-based systems, both with the best F1-scores on the validation dataset. In each column, the best result is bold and underlined, and the second-best result is bold. The two best LLaMA-based systems are those mentioned in Table 3 with substring function applied.

However, since the results for the LLaMA-based models in this table are based on a new prompt that was tested after the competition, the best result during the competition was achieved by the BERT-based models.

Table 4. *Evaluation results of best systems on test dataset.*

No	Model	Precision	Recall	F1-Score	% of Non-Randoms
1	Prompts 1-7	67.2	62.7	**62.8**	**56**
2	SC Prompts 1-5	**78.1**	56.4	62.5	47
3	Prompts 1-4, 7	64.6	47.1	51.4	**70**
4	Prompt with class concept	**69.4**	**67.6**	**66.5**	-
5	Prompt with is-a concept	68.0	**63.2**	62.3	-

5 Results Analysis

5.1 Analysis of BERT-Based Systems Results

The BERT model, when exposed to various prompts, can learn to focus on the relationship between the two target words rather than other words in the sentence. This ability generally leads to improved results as the number of training prompts increases. However, when subclass statement prompts are introduced, except for the second prompt, performance decreases, as shown in Table 1. Consequently, the improvement trend continues until prompt 7, but with the addition of prompt 8, the results significantly deteriorate. It seems that the sharp decline in results with prompt 8 is due to the different words used in prompts 7 and 8. This pattern is also observed when comparing models 9 and 10, where the inclusion of prompt 8 leads to a noticeable drop in various metrics.

During the system design phase, it was hypothesized that BERT might perform better if it consistently sees the (parent, child) pairs in sentences in a fixed order. Therefore, in systems 12 through 16 in Table 1, only prompts in which the parent comes first and the child second, referred to as superclass statement prompts, were used. Although these systems do not perform well with a small number of prompts, as the number of prompts increases to three, the results improve significantly, reaching their best with five prompts.

Looking at the last column in Table 1, it is observed that the models with the lowest F1-scores produce the least random results. However, our best models generate 50 to 60 percent of their results randomly. The reason for this is that while the initial models are more confident in their generated answers, the quality of those answers is not sufficient. On the other hand, in many cases, this random selection is made from between two or three parent candidates, which contributes to the better performance of the final systems. Nevertheless, efforts to reduce the percentage of randomly-generated answers could be a focus for future stages.

Examining the results of the systems on the test dataset, as shown in Table 4, also shows that these results are fairly close to the validation dataset results, and the pattern of results across different systems is consistent. This consistency suggests that BERT has been able to generalize significantly even with a relatively small dataset. The best result comes from the system trained with the first seven prompts, achieving an F1-score of 62.8 percent. Close behind is the system trained with five superclass statement prompts, scoring 62.5 percent.

5.2 Analysis of LLaMA-Based Results

The following points are noteworthy after reviewing the results obtained from structural changes in the prompts:

- In the initial prompt, where concepts were defined broadly, the class concept performed better than the is-a relationship, which was not well understood by the model.
- Using variable names in the prompt definitions and introducing how each concept fits into the prompt helped in understanding the relationships.
- Since the answers are drawn from a set of nine options, including an example in the prompt tends to bias the model toward that answer.
- Contrary to the initial prompt, where classification results were better than the is-a relationship, after modifications in the prompt definitions, the is-a relationship shows better results.
- After structural changes in the prompts, the results for both classification and is-a methods are very close.
- Applying the substring function on the results derived from the improved prompts increases the F1-score on the evaluation dataset to 73.1, which is notable.
- The results from the improved prompts (which were not used in the competition) on the test data show an F1-score of 66.5.

6 Conclusion and Future Work

As observed, BERT, when properly fine-tuned, can yield outstanding results in the taxonomy discovery task in the competition. However, spending more time and experimenting with different combinations of prompts could significantly improve these results. It was also seen that by using appropriate prompts for the LLaMA 3 70B model and adding an auxiliary function like substring, even better results can be achieved. Although this model produces better results than BERT-based systems, the small gap indicates that BERT, when fine-tuned properly, is well-suited for this task. Our results show a significant improvement over the best results reported by BabaeiGiglou et al.

[1] in Task B on the GeoNames dataset, in which the task was simplified as a binary classification problem. This suggests that the methods examined in this paper perform very well in taxonomy discovery.

Future work could explore the following ideas for extending this work.

- For the BERT-based systems:
  - Adding more prompts to the set of prompts.
  - Increasing the number of epochs per prompt while training the model.
  - Using a set of subclass statements instead of superclass statements.
  - Not generating inverted records in the negative dataset.
  - Utilizing other encoder-based language models.
- For the LLaMA-based systems:
  - Using Few-Shot Learning in the prompts and examining its impact on results.
  - Applying the same prompts used in this study to GPT-4 and comparing the results with LLaMA 3 70B.
  - Comparing the results of LLaMA 3 8B and LLaMA 7B.
  - Using knowledge graphs to analyze the relationship between the model’s response and the correct answer.

Authors Contributions

Seyed Mohammad Hossein Hashemi: Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing - Original Draft, Project administration.
Mostafa Karimi Manesh: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft.
Mehrnoush Shamsfard: Writing - Review & Editing, Supervision.

Competing Interests

The authors declare that they have no competing interests.

References

[1] H. Babaei Giglou, J. D’Souza, and S. Auer, “Llms4ol: Large language models for ontology learning,” in *The Semantic Web – ISWC 2023*, T. R. Payne, V. Presutti, G. Qi, *et al.*, Eds., Cham: Springer Nature Switzerland, 2023, pp. 408–427, ISBN: 978-3-031-47240-4.

[2] H. Babaei Giglou, J. D’Souza, and S. Auer, “Llms4ol 2024 overview: The 1st large language models for ontology learning challenge,” *Open Conference Proceedings*, vol. 4, Oct. 2024.

[3] H. Babaei Giglou, J. D’Souza, S. Sadruddin, and S. Auer, “Llms4ol 2024 datasets: Toward ontology learning with large language models,” *Open Conference Proceedings*, vol. 4, Oct. 2024.

[4] “Geonames.” (n.d.), [Online]. Available: https://www.geonames.org/ (visited on 08/05/2024).

[5] V. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Proceedings of the Soviet physics doklady*, 1966.