Predicting Terms in IS-A Relations with Pre-trained Transformers

Irina Nikishina[1], **Polina Chernomorchenko**[2], **Anastasiia Demidova**[3], **Alexander Panchenko**[3,4], **and Chris Biemann**[1]
[1]Universität Hamburg, [2]HSE University,
[3]Skolkovo Institute of Science and Technology, [4]AIRI
irina.nikishina@uni-hamburg.de, pvchernomorchenko@edu.hse.ru,
{anastasiia.demidova,a.panchenko}@skol.tech, chris.biemann@uni-hamburg.de

Abstract

In this paper, we explore the ability of the generative transformers to predict objects in IS-A (hypo-hypernym) relations. We solve the task for both directions of the relations: we learn to predict hypernyms given the input word and hyponyms, given the input concept and its neighbourhood from the taxonomy. To the best of our knowledge, this is the first paper which provides a comprehensive analysis of transformer-based models for the task of hypernymy extraction. Apart from the standard finetuning of various generative models, we experiment with different input formats and prefixes, zero- and few-shot learning strategies, and generation parameters. Results show that higher performance on both subtasks can be achieved by generative transformers with no additional data (like definitions or lemma names). Such models have phenomenally high abilities at the task given a little training and proper prompts in comparison to specialized rule-based and statistical methods as well as encoder-based transformer models.

1 Introduction

Nowadays, pre-trained transofmers including ChatGPT[1] and other transformer-based models with instructions (Ouyang et al., 2022) demonstrate high performance on most NLP tasks (Chowdhery et al., 2022; OpenAI, 2023). However, it is not clear, how well they understand the inner structure of a language and could be applied to purely linguistic tasks, e.g. identification of semantic relations. Those tasks have always been important benchmarks for measuring linguistic capabilities of natural language processing approaches, including neural networks (Jawahar et al., 2019; Rogers et al., 2020). They demonstrate whether the models can comprehend language structure and semantic relations between words: synonymy (Wijesiriwardene et al., 2022), hypernymy (Ravichander et al., 2020), negation (Ettinger, 2020), etc.

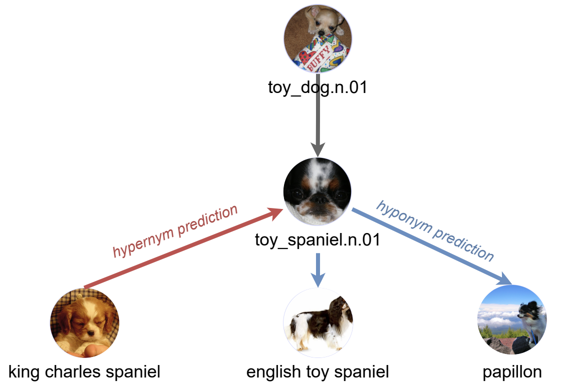


Figure 1: Two formulations of taxonomy enrichment task: attaching new candidates to the existing nodes (red) and generating new candidates at the specified place in the taxonomy (blue).

Applying modern transformers for identifying IS-A relations would allow not only to check the capacities of large language models (LLMs) in linguistics, but also to perform automatic extension of the lexical taxonomic structures with such kinds of relations to alleviate the manual annotation process. Taxonomies play a central role in evaluation tasks, e.g. word-in-context (Armendariz et al., 2020), and are also used for downstream tasks, e.g. (dynamic) entity typing (Del Corro et al., 2015), or for Knowledge Graph Questing Answering (Huang et al., 2019).

There exist several probing experiments for IS-A relation prediction with transformer-based encoder BERT (Devlin et al., 2019; Ettinger, 2020; Hanna and Mareček, 2021). However, to the best of our knowledge, there is no such work on generative transformer models: decoders, e.g. GPT-2 (Radford et al., 2019), or encoder-decoders, e.g. T5 (Raffel et al., 2020).

In this paper, we evaluate generative transformer architectures for two existing Taxonomy Enrichment task formulations (see Figure 1). First, we

[1] https://openai.com/blog/chatgpt

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 134–148
November 1–4, 2023. ©2023 Asian Federation of Natural Language Processing

test zero- and few-shot setups as well as finetuning on the SemEval-2018 Task 9 (Camacho-Collados et al., 2018) dataset on hypernym prediction for English language. We experiment with different prompt formats and the amount of information provided on each hypernym node. Then we evaluate the same models on the hyponym prediction dataset (Nikishina et al., 2022b). Finally, we perform error analysis on the manually selected nodes for hyponym prediction. Thus, we understand the capacities of models for predicting taxonomic relations in both directions.

The main contribution of our work is the first study on using generative transformers for IS-A relationship prediction. We test them in various setups, as it might not be clear, what is the best way to do that: via natural or artificial patterns, few-shot learning or proper fine-tuning, with decoder or encoder-decoder models. We show that prior work based on Hearst (1992) patterns with machine learning classifiers, and encoder-based transformers, are largely outperformed by generative transformers.

Moreover, our approach presents the way for linearization of the context subgraph for using them in LLMs, which can be applied to any task with lexical elements (e.g. synonyms, antonyms, part-of). The methodology may be also generalized to various types of relations available in Knowledge Graphs, such as "capital-of", allowing us to "mine" large pre-trained LMs for new relations.

We also make the code[2] and the models[3] available.

2 Related Work

In this section, we overview the task of Taxonomy Enrichment and previous approaches for predicting IS-A relations. We start with a short description of WordNet[4], which is the main data source for most taxonomy-related tasks.

WordNet (Miller, 1995) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. In this paper, we limit our experiments to nouns represented in 82,115 synsets and 117,798 lemmas. We do not consider other part-of-speech subsets as their hierarchical structure is mostly flat.

Taxonomy Enrichment (Jurgens and Pilehvar, 2016) and Hypernym Discovery (Camacho-Collados et al., 2018) are two main tasks for identifying hypernyms given the input words. The goal of the first task is to attach new words to the correct place in the taxonomy. The second task aims at discovering suitable hypernyms for the input term using a large corpus. Most approaches use static word vector representations like word2vec or fastText (Schlichtkrull and Martínez Alonso, 2016; Bernier-Colborne and Barrière, 2018).

There exist several recent papers on Taxonomy Enrichment that make use of word vector representations and/or large pre-trained language models. For instance, (Nikishina et al., 2022a) present an approach applying numerous of text and graph embeddings as well as their combinations; (Takeoka et al., 2021) solves the same problem, but for the low-resource scenario using BERT-based classifier, while Roller et al. (2018) revise Hearst Patterns for the task. Cho et al. (2020), view taxonomy enrichment as a sequence-to-sequence problem and the authors train an LSTM model on the WordNet data.

3 Hypernymy and Hyponymy Prediction

In this section, we introduce two tasks for predicting terms in IS-A relations. We provide formal descriptions for each task and describe the datasets we use for evaluation.

3.1 Hypernym Prediction Task

SemEval-2018 Task 9 is the acknowledged benchmark for Hypernym Discovery covering several languages and knowledge domains. Camacho-Collados et al. (2018) define the task as finding the appropriate hypernyms $\{h_1, ..., h_n\}$ for a target input term t. At the time of the competition phase, most of the participants of the shared task used Hearst patterns and static word embeddings (Bernier-Colborne and Barrière, 2018; Maldonado and Klubička, 2018; Qiu et al., 2018). A follow-up study (Hanna and Mareček, 2021) analyzes the performance of the BERT model and shows that the encoder-based transformers cannot outperform the state-of-the-art results. To the best of our knowledge, there are no recent studies on large generative transformers like T5 or GPT for this setup.

We use the SemEval-2018 Task 9 (Camacho-Collados et al., 2018) dataset for the hypernym prediction subtask. It consists of a source corpus and input terms with gold hypernyms extracted

[2] https://github.com/orgs/uhh-lt/hypernym-generation

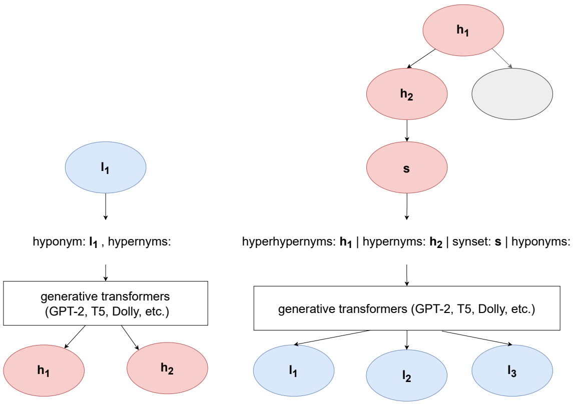
[3] https://huggingface.co/okamifawkes

[4] https://wordnet.princeton.edu

from WordNet (Miller, 1995), Wikidata[5], Multi-WiBi (Flati et al., 2016), and Yago (Rebele et al., 2016). In this paper, we focus on the English general domain dataset (subtask 1A), with 3,000 labelled terms (1,500 in both train and test).

3.2 Hyponym Prediction Setup

Hyponym prediction (generation) is a less studied lexical semantic task. To the best of our knowledge, there exists only one paper where the authors predict hyponyms at the specific place of the taxonomy (Nikishina et al., 2022b). To solve the task, they combine graph representations with the pretrained BERT model. In this paper, we compare their approach to generative transformers in Section 4.2.



h1
h2
s
l1
hyponym: l1 , hypernyms:
hyperhypernyms: h1 | hypernyms: h2 | synset: s | hyponyms:
generative transformers (GPT-2, T5, Dolly, etc.)
generative transformers (GPT-2, T5, Dolly, etc.)
h1
h2
l1
l2
l3

(a) Hypernym prediction (b) Hyponym prediction

Figure 2: Pipelines for predicting IS-A relations for both directions. For hypernym prediction we use the input word in the specific prompt structure, while for the hyponym prediction we linearize the graph structure to feed it to the model as textual input.

The task of Hyponym Prediction is formulated as following. Given subgraph $S = (V, E)$ from a taxonomy T, and a leaf node $v_{leaf} \in V$, the goal is to predict new leaves $l_1, ..., l_n$, which relate to v_{leaf} as hyponyms. Edges E denote the closest (hop$=2$) IS-A relations between v_{leaf}: its hypernyms and hyperhypernyms. In this task formulation, we provide local context, as this information can be used for disambiguation of the synset that can have different meanings, e.g. *"table"* or *"rock"*. Each synset represents only one meaning: if we anticipated children of all meanings at the same time, we would not be able to differentiate, which hyponyms belong to which meanings.

We experiment with following setups: zero-shot, few-shot and model finetuning. We describe the input patterns for zero- and few-shot learning as well as parameters used at the generation step. In this paper, we limit the experiments to the leaf nodes only, as masking and deleting nodes in the middle of the graph needs additional careful study of the graph in order to avoid data leakage. That is why we leave other cases for further experiments.

For this task, we use dataset from Nikishina et al. (2022b). The authors randomly select 1,000 nodes out of 15,646 nodes which children are leaves, i.e., the children do not have hyponyms of their own. They also take into consideration the distance length from the root to the leaf, which should be more than 5 hops. This allows them to exclude the case of predicting very abstract or broad concepts. For each "parental" hypernym all its hyponyms (leaves) were replaced by a single "masked" node and all of the hyponyms are considered as the true answer. All in all, there are 4,376 masked leaves to predict for 1000 synsets.

However, the automatic way of the construction of the CHSP dataset results in occurrence of narrow-field words in the dataset (e.g. "expurgation", "butcherbird", "dot matrix printer"), which hyponyms are too specific for the model to predict. On the other hand, the structure of the WordNet might differ from the pragmatic usage of words in the discourse, so the model would have a different encoded linguistic structure. For instance, according to WordNet, "rottweiler.n.01" is a "shepherd_dog.n.01", which is a "working_dog.n.01" which is a "dog.n.01". At the same time, it is common knowledge that "rottweiler is a dog", therefore, we might expect that models predict "dog" which would not be considered as the correct answer.

To analyse the models performance on common knowledge data, we manually select 22 nodes with hyponyms from the general domain in the English WordNet. Our main guidelines are: (i) to avoid too abstract or general concepts; (ii) to selects ones with numerous instances, subtypes or subclasses, (iii) all the descendants from any level of the chosen node are considered as correct. We are also aware that this dataset is more subjective, however, it is created for additional analysis and not to replace the original dataset. As a result, the selected synsets on average are 7 hops away from the root node and the longest path to a leaf is on average 3. The full list of concepts can be found in Appendix A in Table 7. In this paper, we limit the experiments to the leaf nodes only, as masking and deleting nodes

[5] https://www.wikidata.org

Model	Hypernym Prediction		Hyponym Prediction	
	Average	Best	Average	Best
Zero-shot				
GPT-2-large	0.0351±0.0345	0.0708	0.2677±0.016	0.5280
OPT-1.3b	0.0568±0.0563	0.1140	**0.3081±0.013**	**0.5310**
T5-large	**0.0602±0.0591**	**0.1215**	0.1168±0.001	0.1690
Few-shot				
GPT-2-large	**0.0610±0.0595**	**0.1234**	0.4557±0.005	0.5940
OPT-1.3b	0.0585±0.0577	0.1177	**0.4623±0.005**	**0.6030**
T5-large	0.0058±0.0013	0.0161	0.1144±0.001	0.1920

Table 1: MRR scores for pre-trained models on zero-shot and few-shot setups for predicting hypernyms and hyponyms. *Average* is average MRR on all 26 Hearst patterns, *Best* denotes the best MRR score.

in the middle of the graph needs additional careful study of the graph in order to avoid data leakage.

4 Methodology

The current section presents methodology for zero- and few-shot experiments and the fine-tuning process. It also discusses possible input formats as well as models used in each setup. Figure 2 depicts the strategy for both tasks. The main idea for the hyponym prediction is to transform the input subgraph to the linear form so it can be processed by large language models. For the hypernym prediction task, we do not have a subgraph as input, that is why we provide the input word only. We use the transformers library (Wolf et al., 2020) for all experiments as well as pre-trained models from the same source.

Considering potential data leakage, one may raise a concern about the WordNet being exposed in the LLMs during their pre-training phase. To the best of our knowledge, pre-trained transformers were not trained on the IS-A relationship tasks and WordNet, however, it is quite clear that the model has already seen most of the words from WordNet, which we assume might help to understand the meaning of the input word while predicting hypernyms or retain relevant words from the memory when predicting hyponyms.

4.1 Zero- and Few-shot Learning

To conduct zero-shot experiments with pre-trained generative transformers, we utilize numerous Hearst patterns (14 for hypernym prediction and 26 patterns for hyponym prediction) that naturally precede the generation of hyponyms from (Hanna and Mareček, 2021; Hearst, 1992). We evaluate both decoder (*GPT* (Radford et al., 2019), *OPT* (Zhang et al., 2022)), and encoder-decoder (*T5* (Raffel et al., 2020)) architectures. We use top-k sampling ($k = 20$), and restrict the generation length to up to 10 new tokens in addition to the input pattern. The list of the patterns used can be seen in Tables 8 and 9 (as well as the results for each pattern). Additionally, we sample the results multiple times and sort the answers according to their frequency. As a postprocessing step, we lemmatize the output and keep only nouns.

The next step of our experiments is few-shot learning: we design longer prompts to provide the models with more examples. Each of the extended input contains three patterns completed with the correct answers; examples are separated by a line break. Then the prompt is concatenated with a pattern for a test hypernym node.

4.2 Fine-tuning Model

Since our main goal is to evaluate the ability of generative models to acquire semantic relationships, we utilize commonly known models such as GPT2 and T5. We finetune them on different versions of the input data: default and processed, described below. We also present the results for bigger recent architectures like Flan-T5 (Chung et al., 2022), GPT-J (Wang and Komatsuzaki, 2021) and Dolly (Conover et al., 2023) to define further result boundaries for LLMs.

To conduct several experiments with model fine-tuning, we do not use patterns from the previous step. From the linguistic perspective, by forcing model to predict hyponyms or hypernyms embodied in a certain context (e.g. *"My favourite [PARENT] is [CHILD]."*), we do not teach it to solve a new task. Instead, we make it guess which specific words should be used. From the taxonomy perspective, such patterns do not allow to insert additional information from the taxonomy for the hyponymy

Method	**MAP**	**MRR**	**Pr@1**	**Pr@2**	**Pr@5**	**Pr@10**
word2vec k-NN, $k = 20$	0.0050	0.0210	0.0010	0.0070	0.0109	0.0109
fastText k-NN, $k = 20$	0.0600	0.0190	0.0000	0.0030	0.0090	0.0090
GloVe k-NN, $k = 20$	0.0100	0.0370	0.0000	0.0085	0.0212	0.0208
WebIsaDB (top-20) (Seitner et al., 2016)	0.0222	0.0609	0.0639	0.0460	0.0435	0.0354
TAXIDB (top-20) (Panchenko et al., 2016)	0.0129	0.0398	0.0425	0.0300	0.0275	0.0198
CHSP (Nikishina et al., 2022b)	-	0.0215	0.0228	0.0160	-	0.0074
GPT-2	0.0390	0.1720	0.1100	0.1040	0.0910	0.0740
GPT-2 (input & output proc.)	0.0620	0.2320	0.1950	0.1650	0.1190	0.0910
T5	0.0480	0.1889	0.1230	0.1150	0.0962	0.0764
T5 (input & output proc.)	0.0500	0.1710	0.1130	0.1070	0.0890	0.0690
Flan-T5	0.0330	0.1330	0.0870	0.0750	0.0630	0.0510
Flan-T5 (input & output proc.)	0.0390	0.1410	0.0940	0.0860	0.0710	0.0570
GPT-J 8-bit LoRa	0.1080	0.3230	0.2250	0.1970	0.1620	**0.1230**
Dolly 8-bit LoRa	**0.1110**	**0.3240**	**0.2260**	**0.2020**	**0.1640**	0.1220

Table 2: Results for Hyponym Prediction after fine-tuning on CHSP dataset (Nikishina et al., 2022b).

prediction task.

We design the input formats as (1) for hypernym prediction and (2) for hyponym prediction.

(1) hyponym: l_i, hypernyms: h_{1-k}.

Here we have only an input word l_i, with no additional context or definition and expect the model to predict a list of possible hypernyms h_{1-k}.

(2) hyperhypernyms: h_{1-n} | hypernyms: h_{1-m} | synset: s | hyponyms: l_{1-k}.

In this example, h_{1-n} and h_{1-m} refer to the synset name lists of a certain level, s denotes the target "parental" synset name, and l_{1-k} refers to the output list of lemmas collected from all the descendants of the "parental" synset.

The input & output processed version of the dataset is created with respect to the fact that the output list of correct predictions (both hypernyms and hyponyms) does not have a specific order. Therefore, we can add permutations to the output of the dataset. For each hypernym node, we select 10 random hyponyms and repeat this n times, where n is the total number of hyponyms. Thus, we build a larger dataset of 170,000 examples. As for the input extension, we add three additional examples before the input data. As can be later seen from Tables 2, 3, and 4, such data augmentation improves the results significantly for some models.

All models are trained for three epochs with a batch size of 16. The results are collected as follows: we generate 50 sequences, setting the maximum number of new tokens to 15 and the top-k sampling value to 20. Then we split the output n-grams with a comma (as all models correctly learn the expected output format) and sort the results by frequency. Examples of input and output data are presented in Table 10 in Appendix A.

4.3 Baselines

To compare with other approaches, we implement several baselines: Hearst (1992) patterns, Contextualized Hidden State Projection Method (Nikishina et al., 2022b), which is based on BERT, and k-Nearest Neighbours ($k = 20$) on three different embeddings: word2vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2017), and GloVe (Pennington et al., 2014). We also fine-tune recent large architectures like GPT-J (Wang and Komatsuzaki, 2021) and Dolly (Conover et al., 2023) to understand further capacities of large language models. However, because of the limitation of computational resources, we finetune Dolly and GPT-J using low-rank adaptation for language models and restrict the model to the 8-bit format. We also ask ChatGPT to list possible hyponyms, using its web interface[6] on the common knowledge dataset for hyponym prediction.

5 Experiments

In this section we describe the evaluations metrics, present the result for zero-shot, few-shot and finetuning experiments, and compare performance with different data inputs. We also perform the error analysis for both tasks and explain why certain words or phrases are easier or harder to predict.

[6] Version from 13.04.2023

Method	MAP	MRR	Pr@1	Pr@2	P@5	Pr@10
word2vec k-NN, $k = 20$	0.0400	0.1900	0.0454	0.0681	0.1090	0.1227
fastText k-NN, $k = 20$	0.0200	0.0700	0.0000	0.0227	0.0363	0.0500
GloVe k-NN, $k = 20$	0.0500	0.1900	0.0000	0.0681	0.1363	0.1727
WebIsaDB (top-20) (Seitner et al., 2016)	0.2837	0.5834	0.4500	0.5000	0.4300	0.3600
TAXIDB (top-20) (Panchenko et al., 2016)	0.2240	0.4917	0.4000	0.4250	0.3600	0.2800
CHSP (Nikishina et al., 2022b)	0.0818	0.3377	0.3182	0.2045	0.0818	0.0500
GPT-2-large	0.1270	0.6460	0.5000	0.5000	0.4730	0.4500
GPT-2-large (input & output proc.)	0.3030	0.7320	0.6360	0.6820	0.6450	0.5820
T5-large	0.1460	0.8480	0.7730	0.6590	0.6360	0.5360
T5-large (input & output proc.)	0.1970	0.8200	0.6820	0.7050	0.6180	0.5230
Flan-T5-large	0.1170	0.7220	0.5910	0.6360	0.4910	0.4000
Flan-T5-large (i&o)	0.1990	0.7720	0.6820	0.6590	0.5820	0.5450
GPT-J 8-bit LoRa	0.1780	0.9120	**0.8640**	0.7500	0.6360	0.5500
Dolly 8-bit LoRa	0.1830	**0.9320**	**0.8640**	**0.8410**	**0.7000**	**0.5680**
ChatGPT	**0.3424**	0.7150	0.6363	0.6363	0.6000	0.5091

Table 3: Fine-tuning results for the small common knowledge dataset for Hyponym Prediction.

Method	MAP	MRR	Pr@5
Bernier-Colborne and Barrière (2018)	19.78	36.10	19.03
MSCG-SANITY	11.83	24.79	11.60
Hanna and Mareček (2021)	20.17	12.65	10.49
GPT-2-large	13.89	37.56	11.95
GPT-2-large (input & output proc.)	19.70	35.14	19.18
T5-large	**25.68**	42.40	**23.03**
T5-large (input & output proc.)	18.67	31.19	17.66
Flan-T5	22.97	**45.22**	21.74
Flan-T5 (input & output proc.)	23.07	42.03	22.19
GPT-J-6B 8-bit	19.93	38.75	18.45
Dolly LoRa 8-bit	18.54	37.89	17.21

Table 4: Results for Hypernym Prediction on SemEval-2018 Task 9 subtask 1A: English (Camacho-Collados et al., 2018). Transformer-based models are compared against the top-2 participant results. Standard deviation for both *GPT-2-large* and *T5* models is not more than ±0.001 on 5 runs.

5.1 Evaluation Metrics

Generated candidates are compared against the true hypernyms or hyponyms from the taxonomy. We utilize several metrics for both tasks. First, we apply Precision@k (Pr@k) metric. Pr@k is the ratio of the correct answers measured at the fixed rank k. It allows to understand how many correct answers present in the top-k results. Another metric used for retrieval tasks is Mean Reciprocal Rank (MRR). This metric is more relaxed as it is takes into account the multiplicative inverse of the rank of the first correct answer, but does not reflect coverage. Therefore, we also use the Mean Average Precision (MAP) score which takes into account the total number of gold answers and their rank in the candidate list.

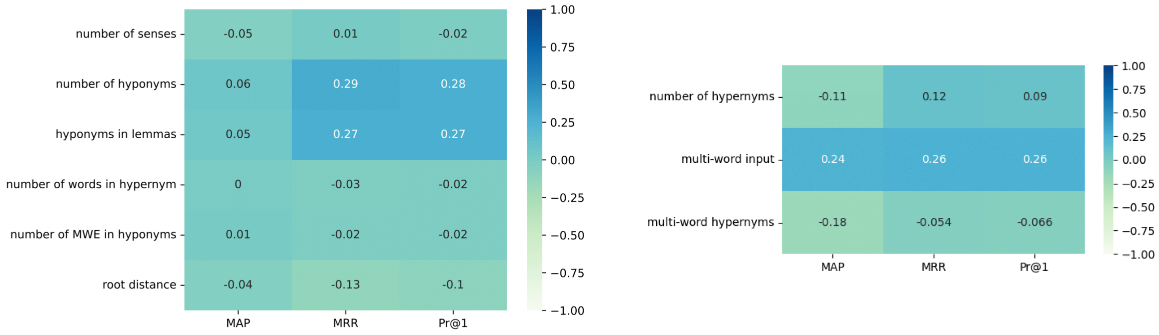
5.2 Discussion of the Results

Precision scores for all tested models in zero-shot setup are presented in Table 1. The second and the fourth columns demonstrate the average scores for all 26 patterns for hyponym prediction and 15 options for hypernym one, whereas the third and the fifth show the best results on both datasets. As one can see, no pattern in the zero-shot setting yields acceptable results for hypernymy prediction, while for the hyponym prediction, the best results are shown by *"other [PARENT] such as [CHILD]"*, which is has the highest average MRR score of 0.3.

As it can be seen from Table 1, the few-shot hyponym prediction scores are improved over the zero-shot results. For this task, the best score is reached with the prompt *"I know such types of [PARENT] as"*. It is quite unexpected, since it is not in the top-5 of the best prompts according to average precision scores for the zero-shot setup.

Considering the hypernym prediction few-shot setup, we achieve the best results with the pattern *"[CHILD] is a type of [PARENT]"*. However, the results are increased only with GPT-2. Other models, in their turn, do not produce significantly improved scores. From the T5 model, we obtain lower results over the zero-shot setup. Also, for OPT-1.3b model, almost identical results were received. Thus, as we show later, zero- and few-shot setups lag far behind the finetuning experiments. We explain the low scores for the results by task complexity and unawareness of the WordNet structure by the model.

The fine-tuning results for the Hypernym Prediction task are displayed in Table 4. Tables 2 and



(a) For hyponym prediction (number of senses, number of children, number of children in lemmas, number of words in synset, root distance)

(b) For hypernym prediction (number of hypernyms, multi-word input, multi-word hypernyms)

Figure 3: Pearson correlation scores for the data characteristics of the input and output against MAP, MRR and Pr@1 scores. The colour denotes the correlation strength: the darker the colour is, the stronger the correlation.

3 denote the results on CHSP and small common knowledge datasets, respectively. As we can see, the best performing models are different for two tasks. Interestingly, the results for GPT-J and Dolly on the Hypernym Prediction dataset are not much higher (or even lower) than smaller and older models like GPT-2 and T5. Moreover, they perform on par with the best approaches from SemEval-2018 Task 9. At the same time, both GPT-J and Dolly demonstrate the best performance on the Hyponym Prediction datasets. In comparison to the baselines and previous approaches with transformer encoders, we can see that generative transformers outperform them by a large margin on both datasets. As for ChatGPT, it demonstrates very high results on the small hyponym prediction dataset, however, we cannot compare it with other models directly, as it is used in the zero-shot setup and not finetuned for the task.

When comparing decoders and encoder-decoder architectures, we also see controversial results. On the CHSP dataset, decoders perform better, while the second-best results for the common knowledge dataset are achieved by T5 and Flan-T5. Moreover, those models also perform better for hypernym prediction, outperforming GPT-J and Dolly. By looking at the predictions generated by decoder and encoder-decoder models, we discover that *GPT*-based approaches predict fewer candidates, whereas the answers of *T5*-like models are longer. This explains the higher results for $Pr@1$ and $Pr@2$ of both *GPT*-based approaches and the lower score of *T5-large* on $Pr@10$. Examples are presented in Table 11 in Appendix A.

Another observation is that very high scores are achieved by the same models on the common knowledge dataset in comparison to the automatically constructed CSHP dataset. Even though we cannot draw any conclusion from their comparison, as the datasets are of different size, we still can see that common domain words are easy to predict, as they come numerous descendants and relaxed evaluation setup (all descendants are considered to be the correct answer).

5.3 Error Analysis

In order to understand the main difficulties of the models while solving both tasks, we perform both quantitative and manual error analysis.

As for hypernym prediction, we first calculate the scores for entities and concepts separately, as it was done in (Camacho-Collados et al., 2018). The results are displayed in Table 5. We can see that entities are easier to predict than concepts across all models. Camacho-Collados et al. (2018) argue that entities are specific instances of concepts, which are easier to comprehend compared to abstract ideas or categories. As we can see from the data, entities that contain frequently occurring hypernyms like “person” and “city” are easier to guess and to memorize for the models.

We also analyse which words are easier / more difficult to find hyponyms for, even though there is not such word types split in the dataset for the hyponym prediction task. In order to do that, we select top-10 input words according to the MAP and Pr@5 metrics for different models to check, whether there are any common linguistic features. Table 8 demonstrates the top-10 words for the best-performing models. From the obtained lists we can conclude that high MAP scores are achieved for synsets with a small number of hyponyms (often

Method	MAP	MRR	Pr@5
Entities			
Bernier-Colborne and Barrière (2018)	29.21	51.82	27.74
MSCG-SANITY	17.72	38.85	16.91
GPT-2-large	25.26	46.75	23.82
T5-large	**33.76**	62.35	**32.29**
Flan-T5	29.82	**62.75**	27.71
GPT-J-6B 8-bit	32.09	60.19	29.66
Dolly LoRa 8-bit	28.04	57.00	25.79
Concepts			
Bernier-Colborne and Barrière (2018)	16.08	30.04	15.41
MSCG-SANITY	09.36	18.90	09.38
GPT-2-large	15.61	28.65	15.12
T5-large	**20.71**	**38.77**	**20.07**
Flan-T5	19.68	37.10	18.73
GPT-J-6B 8-bit	14.83	29.76	13.74
Dolly LoRa 8-bit	14.55	29.87	13.61

Table 5: Results for Hypernym Prediction on SemEval-2018 Task 9 with Entity and Concept splits.

including words from the synset, for example, the only hyponym of the synset "mousse" is "chocolate mousse"), whereas high Pr@5 scores are achieved for more frequent and general concepts. Then we also calculate the average distance from the top-100 and bottom-100 words to the root word. According to our hypothesis, more general (common) words achieve larger scores than very specific ones. The results confirm that the distance to the root for the top-100 words is on average smaller than for bottom-100 (6.65 hops against 7.82 hops).

We also compute some statistical tests in order to understand the main features of input words or the predicted ones that influence the score. We consider the following features for hyponym prediction: input word ambiguity (in how many other senses this word appears in taxonomy), number of words in the input (is input multi-word expression or not), amount of multi-word expressions in hyponyms, number of hyponyms in nodes, number of hyponym in lemmas (one hyponym node can have different lemma names), distance from the root in hops. For hypernym prediction, we consider only the number of hypernyms to predict, multi-word input and multi-word hypernyms output. We cannot use the same features as for hyponym prediction, as the hypernym prediction data is not related to WordNet. Then we calculate the correlation between the listed features with the average Pr@1, MAP and MRR for all fine-tuned models. We display the result correlation matrices in Figure 3. It is calculated with *pandas* (McKinney et al., 2010); figures are generated using *matplotlib* (Hunter, 2007).

The obtained results indicate that most of our hypotheses have not been confirmed for hyponym prediction, see 3b. A weak positive correlation (0.29) is observed only for number of hyponyms and Pr@1 and number of hyponyms and MRR (0.29). We also see a very weak negative correlation of MRR score and root distance, which is -0.13. Other correlation scores do not exceed 0.1. As for hypernym prediction task, we received a weak positive correlation for multi-word input factor among all scores (0.24 and 0.26). Moreover, we observed a very weak negative effect of the number of hypernyms (-0.11) and multi-word hypernyms (-0.18) on the MAP score. Also, attribute the number of hypernyms has a very weak positive influence on the MRR and Pr@1, which are around 0.1. No other correlation scores surpass 0.1.

As for hypernym prediction correlation scores, we can see a weak correlation between the number of input words and MAP, MRR and Pr@1 metrics, see 3b. We assume that having more words in the predicted hypernym can provide additional examples for the training process, potentially leading to more accurate predictions. Additionally, multi-word input can provide more contextual information and connections between different parts of a phrase, which can help to predict hypernyms.

6 Conclusion

In this paper, we explore the practical utility of generative transformer-based models for hyponym and hypernym prediction. We show that promising results can be obtained in both tasks without any additional data, like word definitions or corpora with input term occurrences. Medium-size decoders, and encoder-decoder models yield amazing results after a few epochs of fine-tuning, outperforming other previous baselines by a large margin.

We notice that the fine-tuned sequence-to-sequence outputs contain more plausible candidates, whereas decoders generate fewer candidates with higher quality. Moreover, we cannot conclude which model is the best for both tasks, as the result differ across different datasets. Error analysis of the results shows that there are no specific features like number of senses of the input word in WordNet or number of words to predict that are difficult for the model. Therefore, we assume that generative transformer models demonstrate decent knowledge of IS-A relationships and could be further used for the taxonomy enrichment applications.

Top-10 words	True hyponyms	Predicted hyponyms
mousse	chocolate mousse	chocolate mousse, lemon mousse, coffee mousse
bull	bullock	bulldog, stud, bullock
eclipse	partial eclipse, lunar eclipse, solar eclipse	solar eclipse, lunar eclipse, transit
trace	footprint	footprint, trail, track
wick	candlewick	candlewick, taper, wax wick
learning disorder	dyscalculia, dyslexia, dysgraphia	dyslexia, dysgraphia, dyscalculia
nitrite	sodium nitrite	sodium nitrite, nitroglycerin, nitrocellulose
reproductive system	male reproductive system, female reproductive system	male reproductive system, sexual system, female reproductive system
retinopathy	diabetic retinopathy	diabetic retinopathy, macular degeneration, age-related macular degeneration
eclair	chocolate eclair	mille-feuille, chocolate eclair, vanilla eclair

Table 6: Top-10 synsets with the true hyponyms (at most 3) and predictions (at most 3) according to MAP score.

As the outcome of our research, we also notice that large generative models can not only predict next words (do language modelling), conditioned to some input, but also capture relations between words. Our research further confirms the utility of methodology where information extraction is not directly done on text corpora but on an LLM as a proxy. This new paradigm is useful for several reasons: there is no need to access the original corpus which may be huge and inaccessible, it also allows for certain generalisations required for mining of relations of rare lexical items, and it does not require an explicitly encoded lexical database like WordNet/BabelNet. Our work shows that the very same information may be more compactly stored in the weight of a neural network and explicitly retrieved if needed. This may provide additional computational gain as storage and accessing of large resources like BabelNet featuring millions of nodes and hundreds of millions of relations (hypernyms) between them may be not practical.

As future work, we plan to work with other subtasks for adding new words into taxonomies: insertion of additional nodes in the middle of the graph, crossing two nodes, moving nodes, etc. We also want to solve the tasks for multiple languages using multilingual models and adapters.

Limitations

We find the main limitation of our work as follows:

- We expect that it is possible to further push quality reported in our work if larger versions of large pre-trained transformers are used, such as T5-3b and T5-11b, as it was the case for multiple other tasks. However, the general trend shall be clear from our experiments.
- We did not test multilingual setting of our approach, which is possible if multilingual version of sequence-to-sequence models are used, such as mT5 or mBERT. 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 how similar models and not perform and exhaustive search of all models.

Ethics Statement

We use in our work large neural models, such as T5, 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 straighforward to apply our techniques on other pre-trained models which 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).

References

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020. SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Gabriel Bernier-Colborne and Caroline Barrière. 2018. CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 725–731, New Orleans, Louisiana. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Zaharia Matei, and Reynold Xin. 2023. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM. `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

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*, pages 868–878, Lisbon, Portugal. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Tiziano Flati, Daniele Vannella, Tommaso Pasini, and Roberto Navigli. 2016. MultiWiBi: The multilingual Wikipedia bitaxonomy project. *Artificial Intelligence*, 241:66–102.

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.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *14th International Conference on Computational Linguistics, COLING*, pages 539–545, Nantes, France.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 105–113, New York, NY, USA. Association for Computing Machinery.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

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.

Alfredo Maldonado and Filip Klubička. 2018. ADAPT at SemEval-2018 task 9: Skip-gram word embeddings for unsupervised hypernym discovery in specialised corpora. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 924–927, New Orleans, Louisiana. Association for Computational Linguistics.

Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

George A. Miller. 1995. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

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.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. TAXI at SemEval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, California. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Wei Qiu, Mosha Chen, Linlin Li, and Luo Si. 2018. NLP_HZ at SemEval-2018 task 9: a nearest neighbor approach. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 909–913, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A multilingual knowledge base from Wikipedia, WordNet, and Geonames. In *15th International Semantic Web Conference ISWC Proceedings, Part II*, volume 9982 of *Lecture Notes in Computer Science*, pages 177–185, Kobe, Japan. The Semantic Web.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Michael Schlichtkrull and Héctor Martínez Alonso. 2016. MSejrKu at SemEval-2016 task 14: Taxonomy enrichment by evidence ranking. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1337–1341, San Diego, California. Association for Computational Linguistics.

Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A large DataBase of hypernymy relations extracted from the web. In

Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 360–367, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Thilini Wijesiriwardene, Vinh Nguyen, Goonmeet Bajaj, Hong Yung Yip, Vishesh Javangula, Yuqing Mao, Kin Wah Fung, Srinivasan Parthasarathy, Amit P. Sheth, and Olivier Bodenreider. 2022. UBERT: A novel language model for synonymy prediction at scale in the UMLS metathesaurus. *CoRR*, abs/2204.12716.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *CoRR*, abs/2205.01068.

A Appendix

Synset	Lemmas
coin.n.01	coin
chromatic color.n.01	chromatic color, chromatic colour, spectral color, spectral colour
coat.n.01	coat
jewelry.n.01	jewelry, jewellery
furniture.n.01	furniture, piece of furniture, article of furniture
pasta.n.02	pasta, alimentary paste
cheese.n.01	cheese
room.n.01	room
meat.n.01	meat
wine.n.01	wine, vino
dinosaur.n.01	dinosaur
candy.n.01	candy, confect
beverage.n.01	beverage, drink, drinkable, potable
cloak.n.02	cloak
doll.n.01	doll, dolly
pie.n.01	pie
drum.n.01	drum, membranophone, tympan
makeup.n.01	makeup, make-up, war paint
movie.n.01	movie, film, picture, moving_picture, moving-picture_show, motion_picture, motion-picture_show, picture_show, pic, flick
child's game.n.01	child's game
trouser.n.01	trouser, pant
guitar.n.01	guitar

Table 7: Manually selected test synsets for Hyponym Prediction.

Pattern	GPT-2-large	OPT-1.3b	T5-large	Average
Other [PARENT] such as	0.528	0.531	0.114	0.391
There are a lot of [PARENT] such as	0.454	0.457	0.208	0.373
There are a lot of [PARENT] here such as	0.455	0.459	0.163	0.359
There were a lot of [PARENT] such as	0.469	0.434	0.155	0.353
There were a lot of [PARENT] here such as	0.452	0.438	0.166	0.352
I know such types of [PARENT] as	0.416	0.451	0.118	0.328
[PARENT] such as	0.411	0.396	0.115	0.307
which includes various [PARENT] such as	0.323	0.430	0.146	0.300
I know such kinds of [PARENT] as	0.283	0.456	0.104	0.281
My favorite [PARENT] is	0.326	0.415	0.100	0.280
which includes various [PARENT] like	0.282	0.329	0.169	0.260
[PARENT] e.g.	0.296	0.346	0.118	0.253
Other [PARENT] especially	0.325	0.309	0.077	0.237
My favorite [PARENT] is either	0.246	0.320	0.118	0.228
I know many types of [PARENT] for example	0.228	0.266	0.170	0.221
[PARENT] including	0.202	0.318	0.059	0.193
[PARENT] namely	0.208	0.311	0.051	0.190
I know many kinds of [PARENT] for example	0.181	0.227	0.153	0.187
which includes various [PARENT] for example	0.141	0.28	0.139	0.187
Other [PARENT] for example	0.181	0.193	0.103	0.159
[PARENT] like	0.140	0.192	0.095	0.142
There are a lot of [PARENT] here for example	0.157	0.140	0.126	0.141
There are a lot of [PARENT] for example	0.091	0.130	0.083	0.101
[PARENT] especially	0.103	0.068	0.047	0.073
[PARENT] for example	0.034	0.060	0.065	0.053
[PARENT] for instance	0.027	0.054	0.075	0.052

Table 8: MRR scores for zero-shot hyponyms generation on small common knowledge dataset.

Pattern	GPT-2-large	OPT-1.3b	T5-large	Average
[CHILD] is a type of [PARENT]	0.0615	0.1093	0.1215	0.0974
[CHILD] which is a kind of [PARENT]	0.0379	0.0785	0.0720	0.0628
[CHILD] refers to [PARENT]	0.0653	0.1140	0.0040	0.0611
[CHILD] is child word for [PARENT]	0.0708	0.0735	0.0125	0.0523
[CHILD] is my favorite [PARENT]	0.0407	0.0458	0.0257	0.0374
[CHILD] is a [PARENT]	0.0208	0.0198	0.0555	0.0320
[CHILD] a special case of [PARENT]	0.0385	0.0249	0.0088	0.0241
[CHILD] which is labeled as [PARENT]	0.0209	0.0293	0.0087	0.0196
[CHILD] as [PARENT]	0.0157	0.0259	0.0079	0.0165
[CHILD] belongs to class of [PARENT]	0.0057	0.0040	0.0281	0.0126
[CHILD] belongs to [PARENT]	0.0135	0.0183	0.0039	0.0119
[CHILD] is a member of [PARENT]	0.0139	0.0196	0.0011	0.0115
[CHILD] and other [PARENT]	0.0061	0.0082	0.0138	0.0094
[CHILD] among other [PARENT]	0.0027	0.0009	0.0069	0.0035
[CHILD] is one of [PARENT]	0.0006	0.0005	0.0031	0.0014

Table 9: MRR scores for zero-shot hypernyms generation on SemEval 2018 Task-9 dataset.

Model		Example
FLAN-T5	input	hyponym: buckler, hypernyms:
	output	hat, headgear, earmuff, protective covering, clothing, body armor, protection, clothes
GPT-2 (Radford et al., 2019) (input&output extended)	input	hyponym: inquisitor , hypernyms: judge, jurisconsult, worker, person hyponym: maliciousness , hypernyms: malevolence, distaste, hatred, hate, malignity hyponym: buckler , hypernyms: body armor hyponym: spelunker , hypernyms:
	output	person, explorer, adventurer, traveler, spacer
T5-large (Raffel et al., 2020)	input	hyponym: maliciousness, hypernyms:
	output	malice, evil, malicious intention, wrongful conduct, wrongdoing

Table 10: Examples of models outputs after fine-tuning for the Hypernym Prediction task.

Model		Example
GPT-2 (Radford et al., 2019)	input	hyperhypernyms: toiletry.n.01 | hypernyms: cosmetic.n.01 | synset: makeup.n.01 | hyponyms:
	output	eyeliner, lip, lip balm
GPT-2 (Radford et al., 2019) (input&output extended)	input	hyperhypernyms: garment.n.01 | hypernyms: overgarment.n.01 | synset: cloak.n.02 | hyponyms: burnous, burnoose, burnouse, caftan, kaftan, cape hyperhypernyms: garment.n.01 | hypernyms: overgarment.n.01 | synset: coat.n.01 | hyponyms: box coat, coatee, cutaway, duffel coat, duffle coat, frock coat hyperhypernyms: clothing.n.01 | hypernyms: garment.n.01 | synset: trouser.n.01 | hyponyms:
	output	slacks, chino, straitjacket, jean, trews, short trousers, breeches, flannel, churidars, pantaloon,
T5-large (Raffel et al., 2020)	input	hyperhypernyms: baked goods.n.01 | hypernyms: pastry.n.02 | synset: pie.n.01 | hyponyms:
	output	shepherd's pie, potpie, spud, apple pie, apple pies, baklava, compote, cobbler, strudel, crab cake, pies-a-la-carte, meatloaf, fish pie, trifle, meatpox, pork pie, fruit pie, cheesecake, sausage pie, meatball, applesauce pie, tartlet, steak pie, walnut pie, crabcake, cherry pie, steamer, Victoria pie, crab pie, clam pie, cow pie, hearth pie, sour pie, roulade Pie, scone, roulade pie, Scottish pie, roulade, pies-a-la-mousse, pork-and-veal pie, pork-in-a-brick oven, pork-and-vegetable pie, pie fillet, pork-pie, macaroni pie, bacon-wrapped pie

Table 11: Examples of models outputs after fine-tuning for the Hyponym Prediction task.

Top-10 words	**True hyponyms**	**Predicted hyponyms**
vegetable	pinto bean, artichoke, globe artichoke	beet, broccoli, cabbage
whale	vaquita, Phocoena sinus, right whale	beaked whale, baleen whale, blue whale
military unit	division, naval division, Praetorian Guard	company, airborne unit, artillery unit
court game	tennis, lawn tennis, jai alai	volleyball, basketball, tennis
dwelling	bed and breakfast, bed-and-breakfast, shooting lodge	cabin, farmhouse, chalet
beverage	Burton, Saint Emilion, red wine	ale, beer, alcohol
wheeled vehicle	minicar, lorry, camion	carriage, cart, car
bread	limpa, baking-powder biscuit, simnel	bun, bap, brioche
natural science	statics, cytology, urology	chemistry, biology, botany
citrus	grapefruit, citrange, key lime	orange, grapefruit, lemon

Table 12: Top-10 synsets with the true hyponyms (at most 3) and predictions (at most 3) according to the Pr@5 metric.