2026

Probing Ontological Knowledge in Masked Language Models:

A Comparative Study of BERT and RoBERTa using PRONTO and LMAOSI Approaches

– Cavallini Francesco

BACKGROUND:

Masked Language Models (MLMs) have demonstrated remarkable performance across various NLP tasks, yet their capacity to encode structured ontological knowledge remains under-explored. Understanding whether and how MLMs capture hierarchical is-a relationships is crucial for knowledge-intensive applications.

Our queststion:

"To what extent can LMs infer conceptual knowledge modelled by an ontology?"

OBJECTIVE:

This study investigates the extent to which BERT and RoBERTa encode ontological subsumption relations by **replicating and comparing two state-of-the-art probing methodologies: LMAOSI** (Language Model Analysis for Ontology Subsumption Inference) **and PRONTO** (Prompt-Based Detection of Semantic Containment Patterns).

Our queststion:

“To what extent can LMs infer conceptual knowledge modelled by an ontology?”

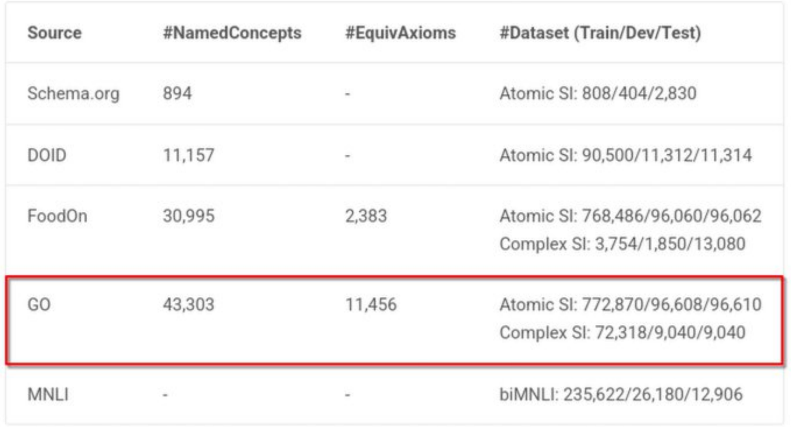
01

THE DATASET

DATASET: OntoLAMA / GO-Atomic-SI

We use one of the datasets suggested **from LMAOSI's paper** for following reasons:

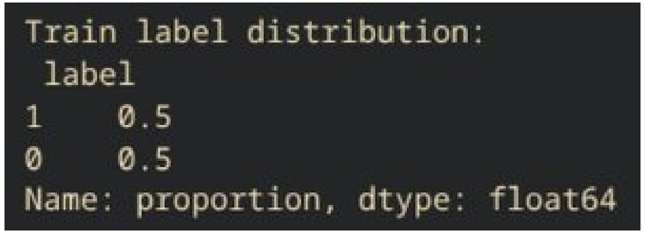
1. **OntoLAMA** is a set of language model (LM) probing datasets and a prompt-based probing method for ontology subsumption inference or ontology completion. The work follows the "LMs-as-KBs" literature but focuses on conceptualised knowledge extracted from formalised KBs such as the OWL ontologies.
   In particular, **GO-Atomic-SI is the one with the most named concepts** in it, therefore we consider it as the most complete:



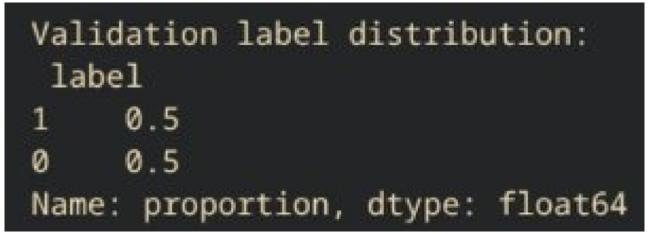
Source	#NamedConcepts	#EquivAxioms	#Dataset (Train/Dev/Test)
Schema.org	894	-	Atomic SI: 808/404/2,830
DOID	11,157	-	Atomic SI: 90,500/11,312/11,314
FoodOn	30,995	2,383	Atomic SI: 768,486/96,060/96,062 Complex SI: 3,754/1,850/13,080
GO	43,303	11,456	Atomic SI: 772,870/96,608/96,610 Complex SI: 72,318/9,040/9,040
MNLI	-	-	biMNLI: 235,622/26,180/12,906

2. Having a dataset with already marked positive and negatives exaples (of concept and sub-concept) would certainly **help replicate LMAOSI's experiment** but also **facilitate dataset manipulation done in PRONTO's paper.**

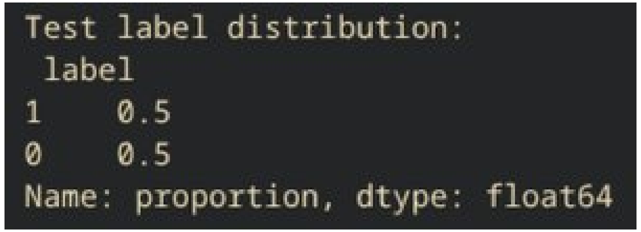
Labels distribution:



```
Train label distribution:
 label
1    0.5
0    0.5
Name: proportion, dtype: float64
```



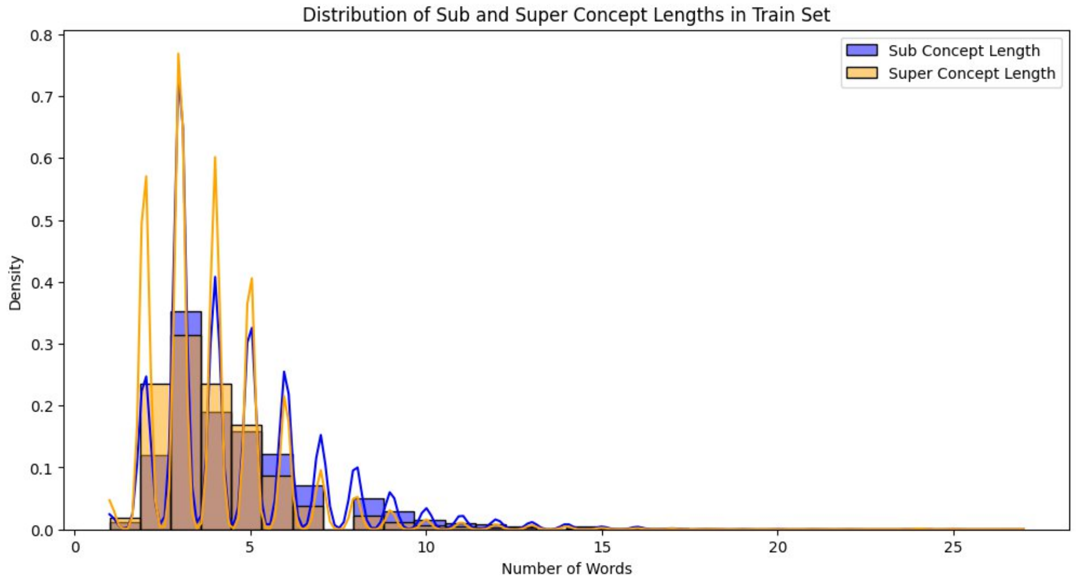
```
Validation label distribution:
 label
1    0.5
0    0.5
Name: proportion, dtype: float64
```



```
Test label distribution:
 label
1    0.5
0    0.5
Name: proportion, dtype: float64
```

We have perfectly even distribution of labels (is-a and not-a)

Token lenghts:



Here we can clearly see that most of tokens are made of 5 or less words (separated by space), wich is a lot, but considering we are using a dataset full of medical terms it seems correct

DATASET MANIPULATION: Creating Prompts

Since we are trying to replicate results from both papers, and those 2 papers use a different prompt style, instead of using multiple prompts from a single paper we decide to use **the prompt which gives better results from each paper**, having a total of 2 prompts.

Where:

- <C> is sub-concept
- <D> is super-concept

PRONTO'S Prompt:

<C> is a <MASK> of <D>

LAMOSI'S Prompt:

It is a <C>? [MASK] It is a <D>

02

BASELINE RESULTS

prompt → MLM (frozen) → prediction → manual verbalizer → label

First probing experiment with manual verbalizer.

We define for both prompts some words that could exemple a positive label and a negative label:

	Positives	**Negatives**
LMAOSI'S prompt (from paper)	"yes", "right"	"no", "wrong"
PRONTO's prompt	"kind", "type", "category", "class", "group", "form", "sort", "example", "instance", "subclass", "species", "variety", "family", "genus"	"part", "member", "component", "piece", "element", "section", "fragment", "portion", "aspect", "feature", "attribute", "property", "role"

0-Shot verbalizers

As said before: by following LMAOSI’s paper first experiment we implement some 0-shot verbalizers. Wich means keeping the whole model (Bert / RoBerta) for prediction of <mask> token.

	BERT	**RoBERTa**
LMAOSI’s paper results	(unvavailable)	49.0%
PRONTO’s prompt experiment	65.0%	61.16% (+12%)
LMAOSI’s prompt experiment	46.96%	51.56% (+2%)

As we can see the results using **PRONTO’s prompt always exceeds the ones obtained in the LMAOSI's paper with 0-shot.** The fact that this prompt is slightly different, suggests to us that it easier for the model to predict the right word.

This suggests to us that **prompt engeneering might be extremely important in this studies**. From here we will try and validate said hypotesis with following experiments.

03

K-SHOT VERBALIZERS

prompt → MLM (frozen) → logits→ trainable classifier trained on total of k random picked samples → label

Probing experiment with **k-verbalizers** (trained models as verbalizers with total of k training samples)

The K-shot setting refers to training a lightweight probe on top of frozen language models using **K labeled examples per class**, enabling the evaluation of how quickly ontological knowledge can be elicited from pre-trained representations.

This is the classical way to train the probe model.

K-Shot verbalizers results

	K = 4		K = 32		K = 128	
	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
LMAOSI’s paper results	/	65.20%	/	84.60%	/	89.0%
PRONTO’s prompt experiment	69.80%	64.50% (-1%)	77.43%	75.13% (-9%)	84.73%	82.63% (-7%)
LMAOSI’s prompt experiment	64.13%	62.80% (-3%)	72.63%	75.53% (-9%)	82.63%	82.46% (-7%)

As we can see, for this experiment, we have a bit of a gap (of around 8%) accuracy between results from original paper (in 32 and 128 shot) and result we obtained in our experiment replication.

Yet, if we look closer the results using **PRONTO’s prompt still exceeds the ones obtained with LMAOSI's Promt**. If we evaluate difference between this prompt (**especially with BERT**) and LMAOSI’s prompt we can **narrow the difference between results from original paper and our results**, once again to sustain hypotesis that prompting correctly matters.

Yet, we can also see that difference between promtpt1 and prompt2 become much smaller the more training the verbalized is subjected to.

PRONTO’S VERBALIZERS

prompt → MLM (frozen) → logits / hidden states → trainable classifier → label

Probing experiment that aims to better performance from regular verbalizers, as seen before, by tweaking the verbalizers models components.

For this experiment we’ll implement a total of 4 verbalizers each independet from one-another

DISCLAIMER: modifying the dataset

To evaluate this experiment PRONTO’s paper uses a completely different dataset of only positives, then builds a dataset with Reverse, Soft and Hard negatives. To fully replicate the experiment we also build a similiar dataset.

Given we are using OntoLAMA (atomic-si/go), we don't have a rich graph like DBpedia, so the realistic and defensible version is negatives to implement. Differenttly from PRONTO's paper by using OntoLams's dataset we already have a dataset that has soft negatives (like the one used int the PRONTO's paper) but we'll need a dataset with following structure:

- **Reverse negatives** If (C, D) is positive → (D, C) is negative
  We introduce reverse negatives to explicitly test the non-commutative nature of semantic containment.

- **Soft negatives** If (C, D) is positive → (C, D')
  Additional soft negatives were not introduced, as OntoLAMA already contains randomly sampled negative pairs and naive soft negative generation would result in trivial examples that do not meaningfully stress containment reasoning.

- **Hard negatives**
  We'll build hard negatives following PRONTO logic using a containment graph reconstructed from OntoLAMA positives. Hard negatives are "cousin" concepts. They share common ancestry with <C> but are NOT directly related by subsumption to <D>. They are "hard" negatives because they are semantically close!

THE VERBALIZERS:

PRONTO-VF

A verbalizer-free baseline approach, where the **hidden state of the [MASK]** token is fed into two fully connected layers with a final sigmoid activation:

$$p(y|x) = \sigma(W_2^T \cdot Tanh(W_1^T \cdot LayerNorm(\genfrac{}{}{0pt}{}{\textit{Hidden}}{\textit{state}}{}_{\mathrm{MLM}}))).$$

PRONTO-WS

A direct-mapping approach where **logits are re-weighted before the Softmax** like this:

$$\mathrm{softmax}(x, w) = \left(\frac{w_1 \exp(x_1)}{\sum_{i=1}^{n} w_i \exp(x_i)}, \ldots, \frac{w_n \exp(x_n)}{\sum_{i=1}^{n} w_i \exp(x_i)} \right),$$

where **w are parameters learned jointly in the optimization process** and constrained in the [0, 1] range.
Then the final label probability is obtained by

$$p(y|x) = \sum_{j=1}^{V_{dim}} \lambda_j p(\texttt{[MASK]} = v_j|x) = \sum_{j=1}^{V_{dim}} \sigma(\beta_j) p(\texttt{[MASK]} = v_j|x),$$

PRONTO-LIN

A naive linear direct-mapping approach:

$$p(y|x) = \sum_{j=1}^{V_{dim}} \lambda_j p(\texttt{[MASK]} = v_j|x) = \sum_{j=1}^{V_{dim}} \sigma(\beta_j) p(\texttt{[MASK]} = v_j|x),$$

where **λj ∈ [0, 1] is a weighting factor that modulates the contribution of each token vj in the vocabulary** to the probability of predicting y given x. The λj weights can be learned through an optimization process aiming to minimize a specified loss function

PRONTO-MAV

A verbalizer-free baseline approach, where the **logits of the [MASK]** token is fed into two fully connected layers with a final sigmoid activation:

$$p(y|x) = \sigma(W_2^T \cdot Tanh(W_1^T \cdot LayerNorm(\mathbf{logits}_{\mathbf{MLM}}))).$$

PRONTO’S verbalizers results: comparaison with “unseen instances”

	VF		LIN		WS		MAV	
	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
PRONTO’s paper results	/	79.09%	/	58.59%	/	74.69%	/	81.97%
PRONTO’s prompt experiment	80.85%	80.35% (no diff)	47.37%	55.45% (-3%)	56.96%	61.45% (-10%)	81.21%	81.01% (no diff)
LMAOSI’s prompt experiment	80.90%	75.65% (-3%)	55.20%	53.58% (-5%)	47.97%	62.22% (-9%)	79.69%	76.36% (-5%)

As we can see, for this experiment, most verbalizers from our experiment exactly replicate results from PRONTO’s paper having minimal difference. Only one that has quite differences in accuracy is WS verbalizer (wich could easily be justified by the fact that that’s the more calculation heavy verbalizer and we have a totally different dataset).

Yet, situation is exactly as in the original paper: the he Mapping-Free **Automatic Verbalizer (MAV-VF) consistently outperforms those based on direct mapping (LIN and WS)**. This suggests token probabilities likely contain complex relationships that direct mapping approaches might miss. The MAV strategy seems to capture these more effectively.
Also we can still see LMAOSI’s prompt always underperforming PRONTO’s prompt, further validating our theory.

Since MAV verbalizer is trained on a more difficult dataset thatn the one of k-shots we can confidently say that this verbalizer is the most performing one we came across even if it reaches slightly lesser results than 128-shot.

04

CONCLUSIONS:

Website containing all evaluations and reasoning about runned experiment

Even throug some experiments got a few percentage of accuracy below expected results from papers, all experiments performed in a similar way of what we'd expect. Also since we used different datasets for some of this experiments we are completely satisfied by the results.

More evaluations of said results can be found here:
https://vr3ed.github.io/data-semantics-project/