Semantic Containment in MLMs: A Prompt-Based Approach[⋆]

Discussion Paper

Vito Walter Anelli[1,*], Alessandro De Bellis[1,*], Tommaso Di Noia[1] and Eugenio Di Sciascio[1]

[1] *Politecnico di Bari, Via Orabona 4, Bari, 70125, Italy*

Abstract

This research explores whether Masked Language Models (MLMs) can understand semantic containment relations, such as sub-class and instance-of relationships, which are crucial for Semantic Web applications. The study introduces PRONTO, a novel approach that leverages MLM predictions to discover semantic containment relations in unstructured text by translating the model's internal predictions into classification labels. The effectiveness, reliability, and interpretability of PRONTO are assessed through a comprehensive probing procedure. The findings demonstrate that MLMs can capture semantic containment relationships, which has significant implications for ontology construction and aligning text data with ontologies. For the sake of reproducibility, we make our code, datasets, and evaluation tools available at https://github.com/sisinflab/PRONTO.

Keywords

Masked Language Models, Prompt Learning, Ontologies

1. Introduction

Pre-trained Language Models (PLMs) have become essential in Natural Language Processing (NLP) due to their ability to capture complex language patterns through extensive training on large text datasets. Studies show PLMs effectively capture factual [2] and ontological [3] knowledge from this pre-training [4]. For example, when given a prompt like "Paris is a [MASK]," a PLM is more likely to predict "capital." This suggests PLMs possess knowledge modeling capabilities beyond simple word co-occurrence [5]. However, this inherent knowledge is rarely used in applications; instead, other types of structured knowledge are employed [6, 7, 8], as these models are often fine-tuned to achieve competitive levels of performance in downstream tasks. This research aims to understand if bidirectional PLMs inherently recognize ontological containment, which includes *subclass* and *instance of* relationships. Ontological containment

SEBD 2025: 33rd Symposium on Advanced Database System, June 16-19, 2025 - Ischia, Italy

[⋆] Extended version [1] published in the 23rd International Semantic Web Conference

[*] Corresponding author.

vitowalter.anelli@poliba.it (V. Anelli); a.debellis6@phd.poliba.it (A. De Bellis); tommaso.dinoia@poliba.it (T. Di Noia); eugenio.disciascio@poliba.it (E. Di Sciascio)

https://sisinflab.poliba.it/anelli/ (V. Anelli); https://sisinflab.poliba.it/dinoia/ (T. Di Noia); https://sisinflab.poliba.it/disciascio/ (E. Di Sciascio)

0000-0002-5567-4307 (V. Anelli); 0000-0002-1220-9878 (A. De Bellis); 0000-0002-0939-5462 (T. Di Noia); 0000-0002-5484-9945 (E. Di Sciascio)

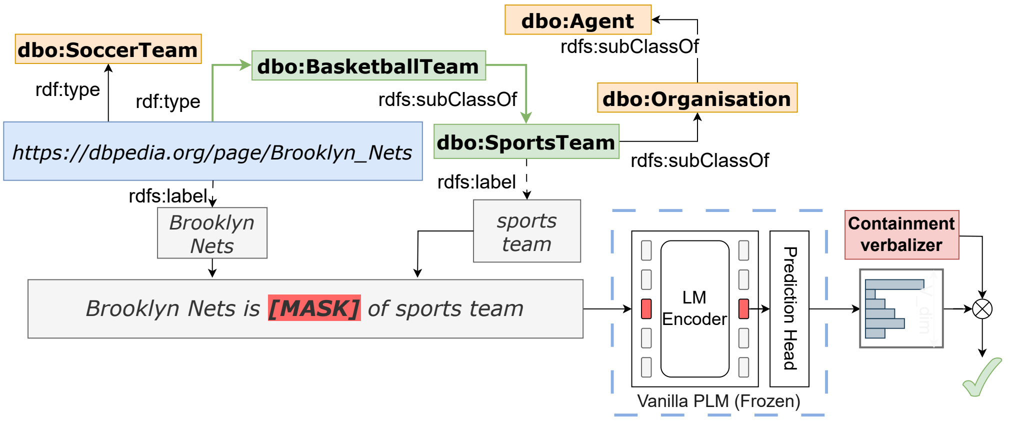

Figure 1: PRONTO Schematization: given a pair representing a containment in a reference taxonomy, representing a semantic containment relationship ("is-a"), PRONTO predicts the plausibility of the pair with a learned verbalizer.

reflects a hierarchical "is a" relationship between entities. The study explores whether PLMs can identify semantic containment when two entities are present in a prompt (e.g., "Paris [MASK] city"), to determine if PLMs are zero-shot semantic containment learners. We propose PRONTO, a novel procedure aimed at the extraction of semantic containment relations from bidirectional PLMs based on the examination of their masked language modeling prediction head. Our key contributions can be summarized as follows:

- We propose a general procedure to probe semantic containment knowledge from MLMs by means of automatically learned verbalizers, i.e. mappings between a MLM prediction head and a label.
- Through extensive analysis, we reveal how vanilla (i.e., not fine-tuned) MLMs exhibit an inner awareness with respect to semantic containment.

To the best of our knowledge, this is the first attempt to use the knowledge stored in PLMs to detect ontological containment through relation prediction with automatically extracted verbalizers. Finally, we present practical applications in zero-shot entity typing.

2. Methodology

In this section, we formally introduce our **containment prediction** task that we schematize in Figure 1. Let $C = \{c_1, c_2, \ldots, c_n\}$ represent the set of classes in a reference ontology O. Each class c_i is a node within the ontology graph. Let E_C be the set of edges representing the *subclass* relations among these classes, where each edge $(c_i, c_j) \in E_C$ denotes that class c_i is a subclass of class c_j. Let $I = \{i_1, i_2, \ldots, i_m\}$ denote the set of instances of classes in C, and E_I be the set of edges denoting the *instance of* relation, where each edge $(i_k, c_i) \in E_I$ indicates that instance i_k is of type c_i, linking instances to their respective classes. We define the **semantic containment graph** G as the union of the two sets of edges E_C and E_I, combined with their respective node sets C and I. Formally, $G = \langle C \cup I, E_C \cup E_I \rangle$. For any two nodes $v_i, v_j \in G$, we aim to determine whether there exists a path from v_i to v_j that signifies an "is-a" relationship within the ontology O. This relationship is characterized by a sequence of edges each representing either a direct subclass relation between classes or an instance belonging to a class, thereby forming a chain of semantic containment. Formally, we aim to learn a model $M_\theta : (v_i, v_j) \mapsto \hat{y}$ with $\hat{y}$ being 1 if there exists a path from v_i to v_j in G and 0 otherwise. The function M_θ is parameterized by the parameters θ derived from a vanilla PLM (e.g., BERT). The aim of M_θ is to learn the mapping $M_\theta : (v_i, v_j) \mapsto \hat{y}$, where $\hat{y}$ represents the predicted probability that a containment relationship exists between the concepts v_i and v_j. Let us define a function P that constructs a prompt for a pre-trained MLM, given two nodes v_i and v_j. The function obtains the verbalized forms of v_i and v_j through $V(\cdot)$ and inserts a mask token `[MASK]` between them to form the prompt. Formally, the prompt construction can be represented as

$$P(v_i, v_j) = V(v_i) \oplus \text{"}\texttt{[MASK] }\text{"} \oplus V(v_j), \tag{1}$$

where V(v_i) and V(v_j) are two natural language representations for the nodes v_i and v_j, respectively. The symbol $\oplus$ stands for string concatenation, and V(v_j) is the `rdfs:label` associated to v_j.

Automatic Extraction of a Containment Verbalizer. Given the prompt $P(v_i, v_j)$ as input to a bidirectional PLM capable of mask-filling, the output of its MLM prediction head consists of the predicted probability distribution over possible tokens that could replace the `[MASK]` (Fig. 1). We propose investigating whether these predicted probabilities can help determine the existence of a containment relationship between v_i and v_j. Given a PLM capable of mask-filling trained on a vocabulary of size V_{dim} and a prompt function $P(\cdot, \cdot)$, we aim to create a mapping between the prediction head output and a discrete label y. Prior work formulate the concept of verbalizer [9] as a discrete mapping between a subset of tokens $v_y = \{v_{y1}, ..., v_{ym}\}$ and a label y. Formally:

$$p(y|x) = \frac{1}{m} \sum_{j=1}^{m} p(\texttt{[MASK]} = v_{yj}|x), \tag{2}$$

with m being the number of tokens in v_y and x being the prompt. The construction of v_y is often done manually: for instance, if y="*city*", a reasonable although simplistic verbalizer construction could be $v_y = \{city, town\}$. In this work, we formulate the construction of a verbalizer v_y as a search problem over the whole vocabulary. This enables our verbalizer to fully exploit the expressiveness of such a large vocabulary and possibly capture associations between labels and tokens that could be not easily identifiable even for domain experts. We want to design a verbalizer as a direct mapping function between the PLM prediction head and a label. An implementation of such a verbalizer is the following:

$$p(y|x) = \sum_{j=1}^{V_{dim}} \lambda_j p(\texttt{[MASK]} = v_j|x) = \sum_{j=1}^{V_{dim}} \sigma(\beta_j) p(\texttt{[MASK]} = v_j|x), \tag{3}$$

where $\lambda_j \in [0, 1]$ is a weighting factor that modulates the contribution of each token v_j in the vocabulary to the probability of predicting y given x. The λ_j weights can be learned through an optimization process aiming to minimize a specified loss function. In fact, we learn the β_j parameters jointly in our optimization procedure, constraining them in a range [0, 1] by means of a sigmoid. Ideally, we want the verbalizer to satisfy two useful properties:

P1 *Noise Resilience*: Since we are dealing with large vocabularies, the significant tokens' marginal probabilities in a PLM prediction head tend to be diluted by the presence of many less relevant tokens. This dilution is linked to the softmax function's property of distributing probabilities across all logits, diminishing the impact of pivotal tokens as the vocabulary size expands.

P2 *Sparsity*: We aim to enforce a sparsity constraint on the λ_j weights to promote interpretability. This constraint facilitates the identification of the most influential tokens minimizing the influence of less relevant ones. In fact, a PLM vocabulary is highly populated even for smaller models (30000+ tokens). Therefore, sparsity can aid interpretability for humans, which can only realistically focus on a smaller set of informative tokens simultaneously.

To satisfy **P1**, MLM prediction head logits pass through a weighted-softmax [10]:

$$\text{softmax}(x, w) = \left(\frac{w_1 \exp(x_1)}{\sum_{i=1}^{n} w_i \exp(x_i)}, \ldots, \frac{w_n \exp(x_n)}{\sum_{i=1}^{n} w_i \exp(x_i)} \right), \tag{4}$$

where w are parameters learned jointly in the optimization process and constrained in the [0, 1] range. To satisfy **P2**, we impose an L1 regularization term over the learned weights λ_j in our loss function. L1 regularization is known to promote sparsity over other alternative regularization strategies, as well as improving generalization. To investigate the potential benefits of non-linearity within our verbalization strategies, we draw inspiration from MAV (Mapping-Free Automatic Verbalizer) [11], in which the authors formulate a mapping-free verbalizer as a non-linear projection of a MLM prediction head in a latent vocabulary space. In our own adaptation, we substitute the inner *Tanh* activation function with *LayerNorm* for numerical stability. This is motivated by the observation that MLM logits can vary in unnormalized ranges, and the *Tanh* function suppresses information associated with high activations:

$$p(y|x) = \sigma(W_2^T \cdot Tanh(W_1^T \cdot LayerNorm(\mathbf{logits_{MLM}}))). \tag{5}$$

In summary, we experiment with different verbalization strategies:

- **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, as in Equation (5);
- **PRONTO-LIN**: a naive linear *direct-mapping* approach, based on Equation (3);
- **PRONTO-WS**: a *direct-mapping* approach where logits are re-weighted before the Softmax as in Equation (4), and the final label probability is obtained as in Equation (3);
- **PRONTO-MAV**: a *mapping-free* approach where logits are fed into two fully connected layers as in Equation (5).

The direct-mapping verbalizers (PRONTO-WS, PRONTO-LIN) are inherently interpretable, since each λ_j can measure the contribution of the j-th token for the final label prediction. On the other side, PRONTO-MAV and PRONTO-VF can give an indication on more subtle patterns in the prediction heads that can only be acquired by means of non-linearities.

Data Preparation. Given a semantic containment graph $G = \langle C \cup I, E_C \cup E_I \rangle$, we denote Π^+ as the set of all the pairs of nodes (s, o) that can be found along a path of G. In other

Table 1
Visualization of the different hard/soft prompts used in this study. [s#] denotes a soft token.

Template Type	**Template ID**	**Prompt Example**
Hard Templates	h_1	Paris is [MASK] of capital
	h_2	Paris is a [MASK] of capital
	h_3	Paris [MASK] capital
	h_4	Paris is [MASK] capital
	h_5	Paris [MASK] of capital
Soft Templates	s_1	Paris is [s1][MASK][s2] of capital
	s_2	Paris [s1][MASK][s2] capital

words, we compute the transitive closure of each node in G.Since G does not contain negative information, this leaves an important decision: how to extract useful negative pairs. This decision is crucial since it impacts both the efficacy and generalizability of our learned verbalizers and the reliability of our evaluation. Intuitively, we want our model to be capable of distinguishing between semantically similar classes, although disjoint ones (e.g., *"city"/"region"*). However, we want it to be also able to distinguish among completely unrelated classes (e.g. *"city"/"person"*). Furthermore, we want it to correctly model a semantic containment relationship that is non-commutative, instead of just discriminating based on word similarity. We devise three strategies to build the set Π^- of negative samples:

- **Reverse negatives**: given a positive pair (s, o) we obtain a negative pair by inverting subject and object (o, s);
- **Soft negatives**: given a positive pair (s, o), we replace o with a random class sampled based on the class distribution in the data;
- **Hard negatives**: given a positive pair $(s_i, o_i) \in \Pi^+$, we build the two sets $\pi^+(s_i, o_i) = \{o_j \mid (s_i, o_j) \in \Pi^+\}$ and $\hat{\pi}^+(s_i, o_i) = \{\hat{o_j} \mid (\hat{o_j}, o_j) \in \Pi^+$ and $\hat{o_j} \notin \pi^+(s_i, o_i)$ and $o_j \in \pi^+(s_i, o_i)\}$. While $\pi^+(s_i, o_i)$ represents the set of nodes along a path starting from s_i in the original graph G, namely all the nodes in a semantic containment relation with s_i, the set $\hat{\pi}^+(s_i, o_i)$ contains the nodes on the paths arriving in $\pi^+(s_i, o_i)$. These nodes are not in a semantic containment relation with s_i but are semantically "close" to it. Given a node s_i, the hard negatives are then built as $(s_i, \hat{o}_j)$ with $\hat{o}_j \in \hat{\pi}^+(s_i, o_i)$.

Prompt Construction. Prior work has demonstrated the sensitivity of PLM outputs to prompt selection [12]. In order to provide a more extensive analysis, we choose to experiment over different prompt templates. We report our prompt choices in Table 1. We design various hard templates to capture various linguistic manifestations of the containment relationship. Regardless of the prompt, both subject and object follow the same verbalization strategy, i.e., the `rdfs:label` literal value. In addition to manually designed prompts, we explore the integration of soft tokens [13], i.e. word vectors jointly fine-tuned during the optimization process.

3. Experiments

This section outlines the experimental setup to probe the ability of PLMs to understand ontological containment relationships. We specifically focus on evaluating the inherent capacity of vanilla pre-trained MLMs prediction heads to recognize the hierarchical relation between instances and classes. The experiments are structured around three core research questions:

RQ1: Do Masked Language Models (MLMs) capture semantic containment?

RQ2: How does contextual information influence MLM in semantic containment prediction tasks?

RQ3: Can MLMs generalize their semantic containment reasoning abilities to new data and tasks?

Dataset. We base our study on the dataset introduced by Wu et al. [3], a reputable dataset from recent literature on probing. This dataset is based on a restriction of DBPedia, containing 783 classes and up to 20 instances per class, with 8753 unique instances. The restriction is necessary because using the entire DBPedia is impractical due to resource limitations. Moreover, multi-hop link extraction scales exponentially as $(\text{entities} \times \text{branching\_factor})^{\text{hops}}$. To extract positive and negative pairs, we follow the procedure described in section 2. We construct the set of negative pairs Π^- as follows: for each pair in Π^+, we sample two hard, one soft and one reverse negative. From the union of negative samples and positive samples $\Pi = \Pi^+ \cup \Pi^-$, we extract training and evaluation splits with holdout. We find that the obtained evaluation split contains a significative amount of soft and reverse negatives, that could potentially inflate performances. For this reason, we extract a more challenging evaluation dataset, that we refer to as Eval (hard), removing all the soft and reverse negatives from the original evaluation split. We use the Eval (hard) dataset as evaluation dataset in all our experiments.

Probed PLMs. It is worth noticing that the proposed probing procedure is versatile and can be readily applied to any bidirectional PLM with mask-filling capabilities. For this investigation, we focus on two prominent encoder-only PLMs,[1] BERT [14] and RoBERTa [15], that leverage a masked language modeling objective during their pre-training stage.

3.1. Semantic Containment Understanding in PLMs (RQ1)

To evaluate the effectiveness of the probed PLMs in identifying semantic containment relationships, we analyze the performance of various combinations of verbalization strategies, templates, and PLMs (the interested reader may take a look to Section 2 for further details). We report the results in Table 2, presenting accuracy, precision, recall, and F1-score for each combination. A decision threshold of 0.5 was used for all models.

PLM Comparison. The analysis reveals several interesting trends. The first finding is that the *verbalization strategy matters*. The Mapping-Free Automatic Verbalizer (MAV) 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. The RoBERTa-Large model generally achieves better and more consistent results, particularly with direct-mapping verbalizations (LIN and WS). For the MAV verbalizer, RoBERTa-Base outperforms the larger model with specific template choices (h_4, h_2, s_1, and s_2). This suggests that prompt design plays a crucial role in performance, even for larger models. There is *no clear correlation between model size and the*

[1] For all the adopted PLMs, we employ the pre-trained checkpoints available at https://huggingface.co/.

Table 2
Results over all combinations of Verbalizer-Template and different PLMs on the DBPedia evaluation (hard) dataset. All negative samples in this evaluation split are hard negatives, constructed as detailed in section 2. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In **bold**, we report the best results for each column.

		RoBERTa-L				RoBERTa-B				BERT-B			
Verbalizer	TID	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
PRONTO-VF	h_1	80.22	95.28	74.37	83.54	75.99	95.57	67.55	79.15	77.09	94.73	69.94	80.47
	h_2	79.98	95.10	74.16	83.34	79.23	96.28	72.00	82.39	77.93	95.69	70.48	81.17
	h_3	78.48	95.36	71.61	81.79	74.47	95.60	65.18	77.51	77.75	95.43	70.40	81.03
	h_4	80.63	96.40	74.06	83.77	78.32	96.32	70.58	82.46	78.16	95.46	71.01	81.44
	h_5	75.94	95.37	67.64	79.14	74.34	95.47	65.07	77.39	76.91	94.02	69.51	80.25
	s_1	76.01	94.54	68.40	79.37	80.07	95.11	74.30	83.42	75.76	95.07	67.58	79.00
	s_2	80.99	95.59	75.30	84.24	80.51	96.14	74.10	83.69	80.38	95.20	74.69	83.71
PRONTO-LIN	h_1	56.32	88.78	40.38	55.51	52.71	89.16	34.08	49.31	46.37	89.04	23.42	37.09
	h_2	60.00	84.05	50.27	62.91	58.36	87.68	44.56	59.09	58.90	84.71	47.72	61.05
	h_3	41.99	81.69	18.10	29.64	46.04	80.22	26.62	39.97	46.98	75.99	31.36	44.39
	h_4	52.48	85.32	35.75	50.39	51.66	86.74	33.49	48.32	42.34	82.60	18.46	30.17
	h_5	49.09	89.34	27.90	42.53	51.45	91.00	31.15	46.41	51.83	89.54	32.41	47.59
	s_1	83.47	84.82	79.88	86.71	79.38	**96.97**	71.69	82.44	82.81	95.08	78.60	86.06
	s_2	80.42	92.93	76.83	84.12	79.57	94.98	73.63	82.95	78.75	94.59	72.67	82.19
PRONTO-WS	h_1	76.22	91.58	71.32	80.19	75.96	90.40	72.02	80.17	70.98	89.63	64.46	74.99
	h_2	73.91	91.56	67.58	77.76	70.47	93.89	60.16	73.33	72.40	90.25	66.27	76.42
	h_3	70.30	89.89	63.08	74.14	67.49	86.13	61.78	71.95	70.50	88.31	64.88	74.80
	h_4	76.05	90.57	72.01	80.23	68.96	87.59	62.92	73.23	71.24	87.46	66.99	75.87
	h_5	72.59	91.95	65.08	76.22	70.44	89.46	63.71	74.42	67.37	89.95	58.15	70.64
	s_1	80.59	93.48	76.57	84.19	81.04	94.21	76.63	84.51	80.89	91.91	78.61	84.74
	s_2	**84.60**	95.09	81.38	**87.70**	82.50	95.37	77.84	85.72	83.81	92.61	**82.61**	87.33
PRONTO-MAV	h_1	84.51	94.50	**81.80**	**87.70**	81.01	96.41	74.64	84.14	83.82	96.79	78.64	86.78
	h_2	81.67	93.97	77.84	85.15	82.77	**96.97**	76.87	85.76	84.32	96.83	79.37	87.24
	h_3	84.00	95.65	79.92	87.08	79.70	93.72	74.94	83.28	83.54	96.09	78.82	86.60
	h_4	84.38	**96.72**	79.55	87.30	**85.06**	95.63	**81.60**	**88.06**	**84.71**	**96.98**	79.83	**87.57**
	h_5	82.31	95.66	77.30	85.51	81.32	96.08	75.40	84.49	83.06	92.16	81.87	86.71
	s_1	79.20	95.51	72.59	82.49	83.09	95.84	78.35	86.22	83.42	95.61	79.05	86.55
	s_2	82.59	95.00	78.33	85.87	84.01	95.71	79.89	87.09	83.94	96.78	78.83	86.89

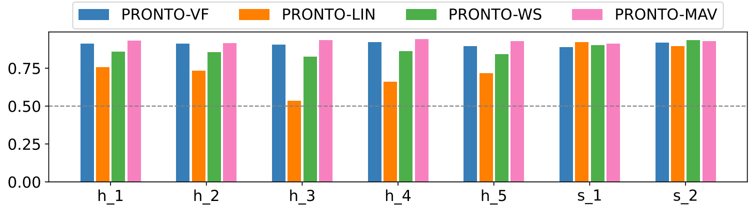

Figure 2: Comparison of AUC scores for different verbalization strategies on the DBPedia Eval (hard) dataset. The reference model is RoBERTa-Large.

PLMs' discriminative ability to distinguish containment relationships. MAV verbalizers show similar performance across model sizes while direct-mapping variants tend to improve with larger models. We hypothesize that smaller PLMs may exhibit more nuanced activation patterns for containment, requiring a non-linear verbalizer like MAV to capture them. This result is in line with previous works that reached conflicting conclusions on this matter: indeed, Petroni et al. [4] showed overall better results for larger PLMs in ontological memorization capabilities, while a more recent study [3] proved that model size does not have reasonable impact on stored ontological knowledge. The analysis suggests that *vocabulary size might not be the primary* *factor influencing performance in this task*. Interestingly, BERT-Base, with a smaller vocabulary compared to RoBERTa-Base (approximately $20,000$ fewer tokens), outperforms RoBERTa-Base for PRONTO-WS and PRONTO-MAV verbalizations across most prompts. This indicates that other factors, potentially the specific tokenization strategies or the training data used for each model, may play a more significant role in capturing semantic relationships.

Figure 2 shows the Area Under the ROC Curve (AUC) scores for all verbalizer-prompt combinations using the RoBERTa-Large PLM. These scores reflect the model's ability to distinguish between positive and negative containment pairs. While overall performance varies with prompt choice for the same verbalizer, the results indicate some general trends. While PRONTO-LIN achieves the lowest accuracy and F1 scores for hard prompts, it exhibits good AUC scores, particularly for the h_1 and h_2 prompts. This suggests that PRONTO-LIN might benefit from optimizing the decision threshold used to classify positive and negative pairs. A potential explanation is in its underlying architecture. Indeed, PRONTO-LIN computes the label probability as a linear sum of individual token probabilities. These token probabilities can be noisy and potentially influenced by irrelevant factors, especially as vocabulary size increases. However, *adjusting the decision threshold could help mitigate the impact of this noise and potentially improve PRONTO-LIN's performance.* The interested reader may find an additional comparison with GPT-3.5 turbo in the extended version of this paper.

Additional analyses. In the extended version of this paper, the interested reader may find the experiments regarding the sensitivity to the relative positioning of instances and classes to determine if the models' predictions were based on memorizing word co-occurrences rather than understanding the underlying meaning of containment relationships. The evaluation set consisted of positive and "reverse negative" examples, with the less specific concept appearing on the right-hand side of the prompt, and the verbalizers performed better on the reverse negative set. This suggests that the verbalizers could distinguish between the relative specificities of concepts, with models sensitive to the order in which concepts are presented. Moreover, we have performed an analysis of the PRONTO-WS verbalizer that revealed both interpretable and less intuitive top tokens, suggesting the model captures nuanced patterns beyond human comprehension. These findings support the idea that containment relationships are intricate and that the model uses a wide range of cues within the vocabulary, highlighting the need to explore the full vocabulary for developing effective verbalizers.

3.2. Enhancing Verbalizers with Knowledge Graph Descriptions: The Impact of Context (RQ2)

Building upon the learned verbalizers, we investigate the feasibility of leveraging textual descriptions from our knowledge graph (KG) to potentially improve their performance in addressing **RQ2**. This exploration is rooted in the hypothesis that enhancing our prompts with relevant context about the entities involved can reinforce the model's understanding of the underlying semantic relationships and lead to better discrimination between positive and negative containment pairs.

To address RQ2, we reformulate the original containment prediction task as a textual entailment task [16]. Here, we aim to infer whether a hypothesis $H(v_i, v_j)$ holds true based on

Table 3
Results over all combinations of Verbalizer-Template on the DBPedia Eval (hard) dataset, in the "with context" setting. The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. ↓ and ↑ denote, respectively, a decrease or an improvement in F1-score with respect to the results in absence of context (Tab. 2). In **bold**, we report the improved F1-scores.

DBPedia (Hard) w. context																
	PRONTO-VF				PRONTO-LIN				PRONTO-WS				PRONTO-MAV			
TID	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
h_1	70.25	89.30	63.53	74.24 ↓	53.58	79.37	42.20	55.10 ≈	65.18	84.67	59.13	69.63 ↓	72.35	88.83	67.52	76.72 ↓
h_2	76.20	90.80	72.05	80.34 ↓	57.12	78.56	50.17	61.23 ↓	68.93	86.34	64.10	73.58 ↓	76.21	89.91	72.94	80.54 ↓
h_3	77.23	92.27	72.31	81.08 ≈	40.87	74.55	18.80	**30.03** ↑	67.38	85.00	62.74	72.20 ↓	81.07	93.23	77.59	84.69 ↓
h_4	71.40	93.20	62.17	74.58 ↓	51.84	81.34	37.16	**51.02** ↑	65.37	86.15	58.02	69.34 ↓	75.99	94.20	68.65	79.42 ↓
h_5	77.30	88.12	76.71	**82.02** ↑	56.17	83.73	43.53	**57.28** ↑	72.49	83.78	73.46	**78.28** ↑	79.96	89.14	80.06	84.35 ↓
s_1	68.99	90.40	60.47	72.47 ↓	73.99	87.52	71.68	78.81 ↓	69.05	86.74	63.92	73.60 ↓	71.75	91.46	64.14	75.40 ↓
s_2	78.40	92.25	74.23	82.27 ↓	58.88	91.21	43.24	58.67 ↓	76.23	87.66	75.40	81.07 ↓	78.91	91.32	75.98	82.95 ↓

a provided natural language premise $P(v_i, v_j)$. The hypothesis is formulated using the same prompt construction method detailed in Section 2. For the premise, we leverage the textual descriptions associated with entities v_i and v_j from the KG. We construct the premise by concatenating the textual descriptions for entities v_i and v_j. Specifically, we use the `dbo:abstract` property for the instances (v_i) and `rdfs:comment` property for the classes (v_j) from DBPedia's Eval (hard) dataset, if available. Since PLMs have a maximum window size, we further process the dataset by removing textual descriptions exceeding 50 tokens in length.

Table 3 presents the final results on the Eval (hard) dataset after incorporating textual descriptions from the knowledge graph (KG). The results reveal an interesting trend. Contrary to expectations, *adding context generally leads to a decrease in performance* across most verbalizer-prompt combinations. This suggests that the KG descriptions might be introducing noise rather than providing beneficial information. This negative impact can be attributed to architectural factors. The prediction heads of the PLMs used may be sensitive to variations in input data, struggling to integrate the additional context effectively. Moreover, the verbalizers themselves might be susceptible to changes in the input, hindering their ability to leverage the supplementary information. Interestingly, direct-mapping verbalizers (like PRONTO-LIN) are less affected by the inclusion of context, showing improvements for specific prompts (h_3, h_4, h_5). This experiment highlights the *need for further investigation into effective strategies for incorporating contextual information* from knowledge graphs.

3.3. Generalizability of Verbalizers (RQ3)

Generalizability to Unseen Instances. To address **RQ3**, we examine how training data size affects verbalizers' generalizability to unseen entities, simulating ontology completion. We adopt an inductive setting, where the model predicts relationships for unseen entities. We modify our training data by removing all training pairs containing randomly selected entities from 80% of the Eval (hard) dataset. We retrain verbalizers on this split and report results in Table 4. Reducing training size negatively impacts performance across metrics, though not substantially. Interestingly, PRONTO-MAV outperforms its full-dataset counterpart in F1 score

Table 4
Results over all combinations of Verbalizer-Template on the DBPedia Eval (hard) dataset, when **80% of instances present in the evaluation set have zero occurrences in the training set**. The reference model is RoBERTa-Large. "Acc", "P", "R" and "F1" denote, respectively, accuracy, precision, recall and F1-score. In **bold**, we report the best results over all verbalizers.

DBPedia (Hard) w. 80% unseen instances																
Verb.	PRONTO-VF				PRONTO-LIN				PRONTO-WS				PRONTO-MAV			
TID	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
h_1	78.61	93.99	72.98	82.16	55.77	89.91	38.82	54.23	75.07	90.54	70.42	79.22	81.47	93.60	77.87	85.01
h_2	79.09	93.20	74.46	82.78	58.59	85.96	46.19	60.09	74.69	89.80	70.50	78.99	81.97	94.17	78.12	85.40
h_3	75.73	94.99	67.60	78.99	41.32	71.16	21.95	33.55	70.18	90.07	62.73	73.95	79.72	96.04	72.96	82.92
h_4	81.48	95.60	76.07	84.72	52.48	88.54	33.99	49.12	72.56	92.40	64.67	76.08	**84.69**	**89.20**	**87.97**	**88.58**
h_5	75.47	94.75	67.40	78.77	50.53	89.89	30.08	45.08	75.02	92.13	68.88	78.82	80.04	92.41	76.73	83.85
s_1	75.30	92.88	68.66	78.96	58.01	86.82	44.54	58.88	81.83	92.66	79.37	85.50	79.85	92.93	75.93	83.58
s_2	78.62	94.50	72.55	82.08	77.58	92.48	72.69	81.40	80.75	94.16	76.22	84.24	80.74	94.37	75.99	84.19

and accuracy with the h_4 prompt, showing strong generalization.

Zero-shot Entity Typing. We evaluate the generalizability of verbalizers through a zero-shot entity typing task, assigning an entity type y to a mention m based on its context. Reformulating this as a textual entailment task, we construct a cloze prompt for each type in Y and select the one with the highest probability. For experiments, we use the Few-NERD dataset [17], a manually annotated NER dataset with fine- and coarse-grained tags. Due to type overlaps (e.g., "Living Thing" under "Person"), we focus on well-defined, disjoint categories: Person (7 types), Location (6 types), and Organization (9 types), excluding ambiguous types like MISC. PRONTO-MAV, our top-performing model, is used without additional training, leveraging the verbalizer from our containment prediction task.

4. Conclusion

This study investigated the ability of pre-trained Masked Language Models (MLMs) to understand hierarchical semantic relationships. The findings suggest that MLMs exhibit some grasp of ontological containment, as evidenced by consistent patterns in the prediction heads. We explored the generalizability of this approach, including learning specific verbalizers, inductive containment prediction, and zero-shot entity typing. While non-linear verbalizers showed remarkable performance, there is room for further exploration on developing more advanced verbalization strategies to better integrate textual information with structured ontological frameworks.

Acknowledgments

The authors acknowledge partial support of the following projects: OVS: Fashion Retail Reloaded and ePansa. We acknowledge the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. D. Bellis, V. W. Anelli, T. D. Noia, E. D. Sciascio, PRONTO: prompt-based detection of semantic containment patterns in mlms, in: G. Demartini, K. Hose, M. Acosta, M. Palmonari, G. Cheng, H. Skaf-Molli, N. Ferranti, D. Hernández, A. Hogan (Eds.), The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II, volume 15232 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 227–246. URL: https://doi.org/10.1007/978-3-031-77850-6_13. doi:10.1007/978-3-031-77850-6_13.

[2] P. Youssef, O. Koraş, M. Li, J. Schlötterer, C. Seifert, Give me the facts! a survey on factual knowledge probing in pre-trained language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 15588–15605. URL: https://aclanthology.org/2023.findings-emnlp.1043. doi:10.18653/v1/2023.findings-emnlp.1043.

[3] W. Wu, C. Jiang, Y. Jiang, P. Xie, K. Tu, Do PLMs know and understand ontological knowledge?, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 3080–3101. URL: https://aclanthology.org/2023.acl-long.173. doi:10.18653/v1/2023.acl-long.173.

[4] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. URL: https://aclanthology.org/D19-1250. doi:10.18653/v1/D19-1250.

[5] V. W. Anelli, G. M. Biancofiore, A. D. Bellis, T. D. Noia, E. D. Sciascio, Interpretability of BERT latent space through knowledge graphs, in: M. A. Hasan, L. Xiong (Eds.), Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022, ACM, 2022, pp. 3806–3810. URL: https://doi.org/10.1145/3511808.3557617. doi:10.1145/3511808.3557617.

[6] V. W. Anelli, T. D. Noia, P. Lops, E. D. Sciascio, Feature factorization for top-n recommendation: From item rating to features relevance, in: Y. Zheng, W. Pan, S. S. Sahebi, I. Fernández (Eds.), Proceedings of the 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning co-located with ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27, 2017, volume 1887 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 16–21. URL: https://ceur-ws.org/Vol-1887/paper3.pdf.

[7] V. W. Anelli, T. D. Noia, E. D. Sciascio, A. Ragone, J. Trotta, Semantic interpretation of top-n recommendations, IEEE Trans. Knowl. Data Eng. 34 (2022) 2416–2428. URL: https://doi.org/10.1109/TKDE.2020.3010215. doi:10.1109/TKDE.2020.3010215.

[8] V. W. Anelli, V. Bellini, T. D. Noia, W. L. Bruna, P. Tomeo, E. D. Sciascio, An analysis on time- and session-aware diversification in recommender systems, in: M. Bieliková, E. Herder, F. Cena, M. C. Desmarais (Eds.), Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017, Bratislava, Slovakia, July 09 - 12, 2017, ACM, 2017, pp. 270–274. URL: https://doi.org/10.1145/3079628.3079703. doi:10.1145/3079628.3079703.

[9] N. Ding, Y. Chen, X. Han, G. Xu, X. Wang, P. Xie, H. Zheng, Z. Liu, J. Li, H.-G. Kim, Prompt-learning for fine-grained entity typing, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 6888–6901. URL: https://aclanthology.org/2022.findings-emnlp.512. doi:10.18653/v1/2022.findings-emnlp.512.

[10] K. Bałazy, Łukasz Struski, M. Śmieja, J. Tabor, r-softmax: Generalized softmax with controllable sparsity rate, 2023. arXiv:2304.05243.

[11] Y. Kho, J. Kim, P. Kang, Boosting prompt-based self-training with mapping-free automatic verbalizer for multi-class classification, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 13786–13800. URL: https://aclanthology.org/2023.findings-emnlp.921. doi:10.18653/v1/2023.findings-emnlp.921.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[13] G. Qin, J. Eisner, Learning how to ask: Querying LMs with mixtures of soft prompts, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 5203–5212. URL: https://aclanthology.org/2021.naacl-main.410. doi:10.18653/v1/2021.naacl-main.410.

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. arXiv:1907.11692.

[16] A. García-Silva, C. Berrío, J. M. Gómez-Pérez, Textual entailment for effective triple validation in object prediction, in: T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón,

G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023, Springer Nature Switzerland, Cham, 2023, pp. 80–100.

[17] N. Ding, G. Xu, Y. Chen, X. Wang, X. Han, P. Xie, H. Zheng, Z. Liu, Few-NERD: A few-shot named entity recognition dataset, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3198–3213. URL: https://aclanthology.org/2021.acl-long.248. doi:10.18653/v1/2021.acl-long.248.