PRONTO: Prompt-Based Detection of Semantic Containment Patterns in MLMs

Alessandro De Bellis(✉), Vito Walter Anelli(✉), Tommaso Di Noia, and Eugenio Di Sciascio

Politecnico di Bari, Bari, Italy
a.debellis6@phd.poliba.it,
{vitowalter.anelli,tommaso.dinoia,eugenio.disciascio}@poliba.it

Abstract. Masked Language Models (MLMs) like BERT and RoBERTa excel at predicting missing words based on context, but their ability to understand deeper semantic relationships is still being assessed. While MLMs have demonstrated impressive capabilities, it is still unclear if they merely exploit statistical word co-occurrence or if they can capture a deeper, structured understanding of meaning, similar to how knowledge is organized in ontologies. This is a topic of increasing interest, with researchers seeking to understand how MLMs might internally represent concepts like ontological classes and semantic containment relations (e.g., *sub-class* and *instance-of*). Unveiling this knowledge could have significant implications for Semantic Web applications, but it necessitates a profound understanding of how these models express such relationships. This work investigates whether MLMs can understand these relationships, presenting a novel approach to automatically leverage the predictions returned by MLMs to discover semantic containment relations in unstructured text. We achieve this by constructing a *verbalizer*, a system that translates the model's internal predictions into classification labels. Through a comprehensive probing procedure, we assess the method's effectiveness, reliability, and interpretability. Our findings demonstrate a key strength of MLMs: their ability to capture semantic containment relationships. These insights bring significant implications for MLM application in ontology construction and aligning text data with ontologies.

Keywords: Masked Language Models · Prompt Learning · Ontologies

1 Introduction

Pre-trained Language Models (PLMs) have emerged as a cornerstone in the field of Natural Language Processing (NLP). Leveraging vast amounts of text data, these models undergo extensive pre-training, typically is self-supervised fashion, allowing them to capture intricate patterns and dependencies within language.

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15232, pp. 227–246, 2025.
https://doi.org/10.1007/978-3-031-77850-6_13

Numerous studies have investigated the knowledge embedded within PLMs, demonstrating that these models effectively capture a significant amount of information from their pre-training content, encompassing both factual and ontological knowledge. Motivated by the work of Petroni et al. [20], several works investigate factual [25] and ontological [24] knowledge in PLMs by means of templated prompts. The idea is that given a prompt in the form of "Paris is a `[MASK]`", a PLM might predict "capital" with significantly higher probability than other possible tokens. Regardless of the methodology, many works concur on the idea that PLMs inherently possess some knowledge modeling capabilities that transcend mere textual co-occurrence patterns. Nevertheless, this kind of knowledge is rarely exploited in applicative scenarios and other types of structured knowledge are employed [2,3], as these models are often fine-tuned to achieve competitive levels of performance in downstream tasks.

In this work, we try to develop an understanding of how bidirectional PLMs might be inherently aware of a specific type of semantic relationship: **ontological containment**, i.e. *subclass* relation between ontological classes and *instance of* between an entity (instance) and an ontological class. Ontological containment refers to a hierarchical relationship between entities, where one is more general and encompasses another, indicating an "is a" relationship. This kind of knowledge is particularly interesting, as it is especially useful for various downstream tasks, such as entity typing and ontology completion. In particular, we are interested in whether a PLM can recognize a semantic containment relationship in the case where two entities are explicitly provided in the prompt (e.g. "Paris `[MASK]` city"). In other words, the overarching question of our investigation is: *are PLMs zero-shot semantic containment learners?*

Developing on several trends of prompt-learning and knowledge probing in PLMs, 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.

Our work significantly differs from prior literature. Prior prompt-based evaluation of taxonomical knowledge in PLMs formulates the task as object prediction [14,20,24]. In addition, Jain and Espinosa-Anke [14] analyze zero-shot taxonomy learning in PLMs considering joint probabilities of sentence tokens to predict a broader class for a given instance. Similarly, Huang et al. [13] frame entity typing as a fill-in-the-blanks task, learning a linear mapping between an MLM head and a fixed set of classes. We extend beyond the mapping to capture a general containment relationship that can be applied to an arbitrarily large set of classes. Specifically, we propose a methodology to train (linear and non-linear) verbalizers to detect whether an ontological containment relation (i.e., a different

and broader task) holds between two given instances (i.e., a different prompt). Furthermore, differently from the work of Huang et al., we leverage vanilla PLMs. 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 highlight the significance of the findings by presenting practical applications in zero-shot entity typing.

The extraction of general semantic containment patterns from PLMs has significant potential to enhance downstream tasks at the intersection of text and structured knowledge, such as ontology completion and entity typing. The study may lead to more advanced methodologies for integrating text with structured ontological frameworks, advancing the field of knowledge representation.

2 Related Work

Ontological Knowledge in PLMs. The problem of probing factual and ontological knowledge in PLMs has already been tackled by prior work [25]. Among this vast body of literature, our work is particularly aligned with the line of research that focuses on probing conceptual [7,19] and ontological knowledge [1,8,24] using manually constructed cloze prompts: this kind of approach is based on the construction of ad-hoc prompts to verbalize triples in a KG, where a part of the triple (usually the object entity) is masked (e.g. "*Paris is the capital of* `[MASK]`"); the PLM is then asked to fill in the blanks, evaluating the ability to infer the missing object. This research was initially propelled by the seminal work of Petroni et al. [20]. Subsequent work [15,21] follows the same general idea, proposing refinements over the methodology and datasets. For instance, Jain and Espinosa-Anke [14] test taxonomical knowledge in PLMs by means of prompt-based object prediction and sentence scoring. More recently, Wu et al. [24] proposed a systematic procedure to probe ontological knowledge in PLMs, in terms of real-world classes and relationships between them. A key challenge in prompt-based evaluation of ontological knowledge in PLMs resides in how to construct the prompt and how to map discrete concepts (i.e. entities, classes) to a meaningful textual representation. Most prior works craft prompts manually, while other works [22] enhance the pre-trained embedding vocabulary of existing PLMs with fine-tuned word vectors, i.e. "soft prompts".

Verbalizers and Prompt Learning. A *verbalizer* is a general mapping between a masked language modeling problem and a classification problem. Verbalizers can be manually crafted [23], constructed leveraging external knowledge sources [6,11] or learned [16]. Verbalizers are used as a conditioning signal to tune an MLM prediction head on a downstream task, typically in few-shot settings.

Prompt-Based Entity Typing. Understanding semantic containment in PLMs can enhance tasks fundamentally reliant on the concept of containment, such as entity typing. Prior works [10,13] frame entity typing as a fill-in-the-blank task, using the predictions from a masked language model (MLM) to assess the class membership of an entity mention m in a sentence x (e.g. "x,

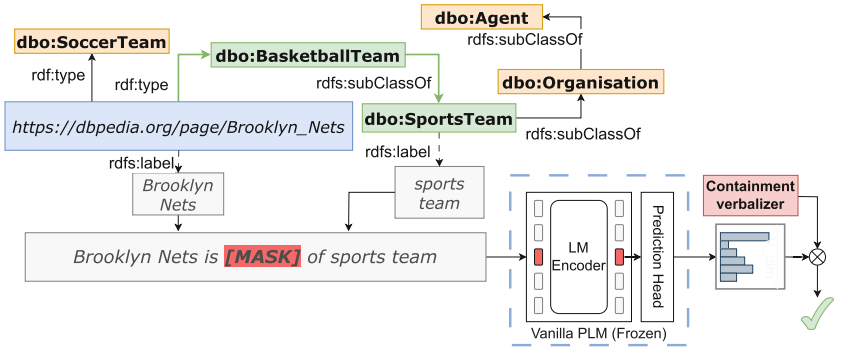


Fig. 1. PRONTO Schematization: given a pair representing a containment in a reference taxonomy, representing a semantic containment relationship ("is-a"), PRONTO predicts the plausibility of the pair with a learned verbalizer.

m is a type of `[MASK]`"). They fine-tune a linear verbalizer that maps the predictions to a limited set of class labels. However, this approach learns a static mapping tied to predefined classes, limiting its applicability to new, unseen classes. *We expand upon the concept of the verbalizer to investigate if it is possible to learn a more general notion of containment extending beyond a fixed set of classes*.

3 Methodology

In this section, we formally introduce our **containment prediction** task that we schematize in Fig. 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:

$$\hat{y} = \begin{cases} 1 & \text{if there exists a path from } v_i \text{ to } v_j \text{ in } G, \\ 0 & \text{otherwise.} \end{cases}$$

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 $\mathrm{V}(v_i)$ and $\mathrm{V}(v_j)$ are two natural language representations for the nodes v_i and v_j, respectively. The symbol $\oplus$ stands for string concatenation. Without loss of generality, in our implementation $\mathrm{V}(v_j)$ is the `rdfs:label` associated to v_j.

3.1 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 [10] 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. A naive 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 [4]:

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

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) [16], 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 modification 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 Eq. (5);
- **PRONTO-LIN**: a naive linear *direct-mapping* approach, based on Eq. (3);
- **PRONTO-WS**: a *direct-mapping* approach where logits are re-weighted before the Softmax as in Eq. (4), and the final label probability is obtained as in Eq. 3;

– **PRONTO-MAV**: a *mapping-free* approach where logits fed into two fully connected layers as in Eq. (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.

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

3.2 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 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. Based on these considerations, 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 $\hat{o}_j \in \hat{\pi}^+(s_i, o_i)$.

Table 3 illustrates some examples of negative pair construction, including a positive pair compared to a Hard, Soft, and Reverse negative pair, respectively.

Table 2. Full statistics for the constructed DBPedia dataset splits.

		Train	Val	Eval	Eval (Hard)
Total Pairs		141090	372	36824	27286
Positive Pairs		18597	182	18416	18416
Negative Pairs	Total	122493	190	18408	8870
	Hard Negatives	59632	96	8870	8870
	Soft Negatives	30556	40	4702	0
	Reverse Negatives	32305	54	4836	0

3.3 Prompt Construction

Prior work has demonstrated the sensitivity of PLM outputs to prompt selection [5]. 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 [22], i.e. word vectors jointly fine-tuned during the optimization process.

3.4 Training Procedure

To account for the significant number of constructed negative pairs in our dataset, we leverage α-balanced focal loss [17] with label smoothing in order to reduce overfitting. We tune the α parameter based on the validation F1 score. All the verbalizers are tuned using Adam optimizer, selecting the optimal learning rate based on validation set performance. L1 regularization is employed for direct mapping verbalizers (as described in Sect. 3.1), while PRONTO-MAV and PRONTO-VF are regularized with weight decay.

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

Table 3. Examples of negative pair construction. The examples are generated from the DBPedia dataset following the strategy described in Sect. 3.2.

	Example Pairs #1	Example Pairs #2
Positive	Baltimore Ravens, american football Team	1,2,6-Hexanetriol, chemical compound
Hard Neg.	Baltimore Ravens, soccer club	1,2,6-Hexanetriol, monoclonal antibody
Soft Neg.	Baltimore Ravens, album	1,2,6-Hexanetriol, animal
Reverse	american football Team, Baltimore Ravens	chemical compound, 1,2,6-Hexanetriol

Dataset. We base our study on the dataset introduced by Wu et al. [24], 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 Sect. 3.2. We construct the set of negative pairs $\mathit{\Pi}^-$ as follows: for each pair in $\mathit{\Pi}^+$, we sample two hard, one soft and one reverse negative. We determine this to be a good ratio empirically through various tests on the validation set (F1-score showed a substantial improvement for the 2-1-1 ratio compared to 1-1-1, 1-2-1, and 1-3-1 for containment prediction; specific details were omitted for brevity). From the union of negative samples and positive samples $\mathit{\Pi} = \mathit{\Pi}^+ \cup \mathit{\Pi}^-$, 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. The statistics of the obtained splits are reported in Table 2.

Table 4. 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 Sect. 3.2. "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

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 [9] and RoBERTa [18], that leverage a masked language modeling objective during their pre-training stage. Extending this study to include unidirectional LMs requires additional considerations due to inherent architec-

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

Table 5. GPT-3.5 turbo results (zero-shot) on the Eval (hard) dataset.

	Acc	P	R	F1
GPT-3.5 turbo (zero-shot)	60.18	96.36	42.61	59.10

tural diversities. This would necessitate constructing "unnatural" prompts, such as "The relation between Paris and Capital is". Moreover, access to prediction heads is not always available. Therefore, we choose to restrict this study to the two aforementioned models. For the chosen models, we employ the case-sensitive variants to ensure a consistent and fair comparison.

Hyperparameter Tuning. We select optimal learning rate among {1e-1, 1e-2, 1e-5}, L1 coefficient among {1e-3, 1e-4} based on the validation F1 score. The training procedure is carried for a maximum number of epochs with early stopping if the validation F1 score does not improve after three consecutive epochs.

4.1 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 Sect. 3 for further details). We report the results in Table 4, 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 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. [20] showed overall better results for larger PLMs in ontological memorization capabilities, while a more recent study [24] proved that model size does not have reasonable impact on stored ontological knowledge.

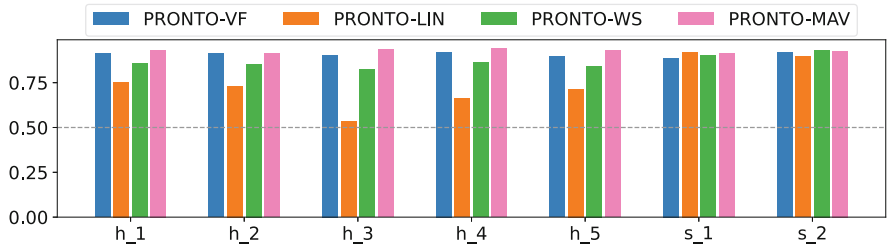


Fig. 2. Comparison of AUC scores for different verbalization strategies on the DBPedia Eval (hard) dataset. The reference model is RoBERTa-Large.

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.*

Table 5 presents the performance of GPT-3.5 turbo on the same containment prediction task. Here, the containment hypothesis is phrased in natural language as "*V(v_i) is kind of V(v_j)*" and GPT-3.5 is asked to judge its truth value (true or false). This evaluation is conducted in a zero-shot scenario, meaning GPT-3.5 receives no task-specific training. Therefore, the results are not directly comparable to the performance of the other verbalizer models discussed earlier. Despite the limitations, this zero-shot evaluation provides a valuable insight into the inherent difficulty of the Eval (hard) dataset. The performance of GPT-3.5 on this task can serve as a baseline for comparison with the other models and potentially indicate the complexity of the overall reasoning required for the task.

Sensitivity to the Relative Positioning of Instances and Classes in the Unstructured Text. To gauge whether the models' predictions extend beyond simply memorizing word co-occurrences, we analyze their performance on a specifically-designed evaluation set. This set consists solely of positive and "reverse negative" examples (see Sect. 3.2 for a detailed definition), where the less specific concept appears on the right-hand side of the prompt (opposite to the positive examples). We compare these results to those obtained on the Eval (hard) set. The AUC scores for both sets are presented in Table 6.

Table 6. Performances comparison with the reverse negatives-only dataset. All results are in terms of AUC score. The reference model is RoBERTa-Large.

Verbalizer	TID	DBPedia Eval (Hard)	DBPedia Eval (Reverse Neg.)
PRONTO-WS	h_1	85.83	97.26
	h_2	85.39	97.79
	h_3	82.54	97.45
	h_4	86.19	98.03
	h_5	84.29	96.53
	s_1	90.12	99.08
	s_2	93.27	99.35

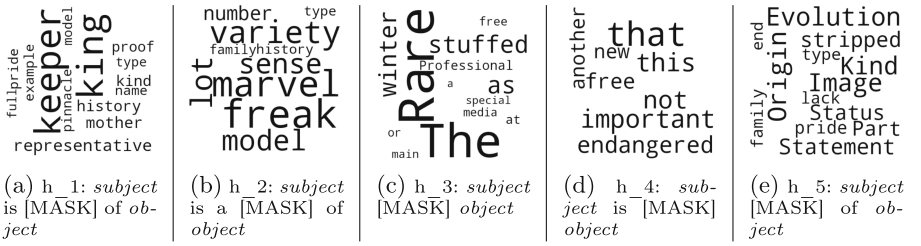
The rationale behind this experiment is to assess whether the verbalizers can understand the underlying meaning of containment relationships beyond just memorizing word co-occurrences. If the verbalizers perform well on "reverse negative evaluation set, it suggests a deeper understanding of containment since reverse negatives flip the concept order compared to positive examples, making memorization less effective.

The results show that the verbalizer performs significantly better on the reverse negative set. This suggests *the verbalizer can indeed distinguish between the relative specificities of concepts within the prompts, even when the relationship is flipped*. This finding supports the notion that the containment relationship depends on the relative positioning of the concepts and models appear to be sensitive to the order in which concepts are presented. The higher AUC scores for reverse negative examples compared to hard negatives indicate that the PLMs are generally adept at identifying specific concepts from more general ones.

Insights from Learned Verbalizer Weights. Figure 3 provides valuable insights into how the PRONTO-WS verbalizer discriminates between positive and negative containment pairs, showing the most important tokens according to the associated verbalizer weights. While some top tokens like "variety", "type", and "kind" are easily interpretable, many others are less intuitive. This suggests that the *model captures nuanced patterns in the data that might not be readily apparent to humans*.

In addition, we found the presence of some counter-intuitive tokens (e.g. "not") that would suggest negating the relation. This is likely due to the model trying to compensate for discrepancies between the MLM prediction and the ground truth —a degree of statistical "disagreement" (i.e. the LM wrongly predicts the "not" token for certain relationships where it lacks background knowledge). This could also be motivated by the presence of specific token bias, although this necessitates further investigation.

These findings support the notion that *containment relationships are more intricate than what can be fully expressed through manually designed verbalizers*. The model appears to leverage a broader range of cues within the vocabulary. Overall, this analysis highlights the necessity of exploring the full vocabulary to develop more effective verbalizers benefitting from a wider range of interactions.



(a) h_1: *subject* is [MASK] of *object*

(b) h_2: *subject* is a [MASK] of *object*

(c) h_3: *subject* [MASK] *object*

(d) h_4: *subject* is [MASK] *object*

(e) h_5: *subject* [MASK] of *object*

Fig. 3. Visualization of learned verbalizers: the wordclouds report the top weighted tokens for the "is-a" relationship in hard templates. The reference model is RoBERTa-Large under the PRONTO-WS setting.

4.2 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 [12]. Here, we aim to infer whether a hypothesis $H(v_i, v_j)$ holds true based on a provided natural language premise $P(v_i, v_j)$. The hypothesis is formulated using the same prompt construction method detailed in Sect. 3.3. 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 7 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.

Table 7. 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. 4). 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 ↓

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.

4.3 Generalizability of Verbalizers (RQ3)

Generalizability to Unseen Instances. To address **RQ3**, we investigate the impact of training data size on the generalizability of the verbalizers to unseen entities. This mimics a real-world scenario of ontology completion, where we want a model to leverage its knowledge to predict relationships for new entities. We transform the containment prediction task into an inductive setting. Here, the model predicts containment relationships for potentially unseen entities not

present in the training data. We process our training dataset as follows: we randomly select 80 % of the instances in the Eval (hard) dataset and we remove all the pairs in the training set with any of these entities. This operation results in a total of 20690 test pairs (≈75% of the total Eval (hard) dataset) with an unseen instance. The resulting (reduced) training set is composed by a total of 38742 pairs (≈27% of the original size). We retrain our verbalizers based on this reduces training split and report the results on the Eval (hard) dataset in Table 8. Overall, reducing training size has a negative impact on performance across all metrics, although the decrease is not substantial. Interestingly, this trend does not hold for certain verbalizer and prompt combinations. For example, PRONTO-MAV outperforms its counterpart that was trained on the full dataset in both F1 score and accuracy when using the h_4 prompt template. Despite this, performance remains comparable with the other verbalizer-prompt combinations, demonstrating robust generalization.

Table 8. 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

Zero-shot Entity Typing. To validate the generalizability of the developed verbalizers, we investigate an entity typing task within a zero-shot setting. The aim is to assess the type of an entity based on its surrounding context. Given a tokenized text sequence $S = (t_1, t_2, \ldots, m_i, \ldots, m_j, \ldots, t_T)$, where $m = (m_i, \ldots, m_j)$ represents an entity mention, the aim is to assign an entity type label y from a predefined set of types Y to the entity mention m. We can reformulate this task as a textual entailment task: given an premise $P = S$, the goal is to predict whether an hypothesis $H = P(m, y)$ holds, where $P(m, y)$ is a cloze prompt constructed as in Eq. 1. We formulate a separate entailment prompt for each of the types in Y, selecting the one that achieves the highest probability as the final entity type prediction. We selected the Few-NERD dataset for our experiments [11], a high-quality, manually annotated NER dataset featuring

both fine-grained and coarse-grained tags. This dataset poses challenges such as varied types that lack clear textual representations and potential overlaps among types. For example, the fine-grained type "Living Thing" may overlap with the fine-grained types within the "Person" coarse type, hence it is not directly exploitable in a zero-shot setting. Consequently, we focus our experiments on the fine-grained splits corresponding to the well-defined and disjoint coarse categories of Person (7 types), Location (6 types), and Organization (9 types). Additionally, we exclude all types labeled as MISC or "Other" due to their ambiguous nature. In our entity typing experiments, we utilize PRONTO-MAV, the model that achieved the highest overall performance in our prior tasks. In this case, no training is conducted, as we directly employ the same verbalizer obtained from our containment prediction task detailed in Sect. 4.1.

Table 9. Entity Typing results (zero-shot) on Few-NERD. The reference model is RoBERTa-Large.

	Few-NERD (COARSE)		Few-NERD (PER)		Few-NERD (ORG)		Few-NERD (LOC)	
	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1
PRONTO-MAV (zero-shot)	50.11	55.91	61.44	68.05	64.06	69.28	49.59	70.67

Table 9 reports the results on Few-NERD (Coarse) and three fine-grained splits (Person, Location, Organization). Despite the considerable challenges posed by the dataset, requiring type inference from sentences where entities are often mentioned without sufficient context, we achieved remarkable results in terms of micro F1 score on the fine-grained splits. On the coarse-grained dataset, the results are lower, due to the presence of semantically similar and possibly overlapping types (e.g. Location and Building, Product and Art).

5 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. This knowledge about containment holds significant potential for tasks where labeled data is scarce, such as ontology completion and entity typing. 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. Additionally, research on improving the relevance of descriptions used for context integration or exploring alternative context integration methods could be fruitful avenues for further advancement.

Supplemental Material Statement: For the sake of reproducibility, we make our code, datasets and evaluation tools available on GitHub.[2]

Acknowledgments. The authors acknowledge partial support of the following projects: OVS: Fashion Retail Reloaded, Lutech Digitale 4.0, Secure Safe Apulia, Patti Territoriali WP1, BIO-D. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

References

1. Anelli, V.W., Biancofiore, G.M., Bellis, A.D., Noia, T.D., Sciascio, E.D.: Interpretability of BERT latent space through knowledge graphs. In: Hasan, M.A., Xiong, L. (eds.) Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17-21 October 2022, pp. 3806–3810. ACM (2022). https://doi.org/10.1145/3511808.3557617
2. Anelli, V.W., Noia, T.D., Lops, P., Sciascio, E.D.: Feature factorization for top-n recommendation: From item rating to features relevance. In: Zheng, Y., Pan, W., Sahebi, S.S., Fernández, I. (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. CEUR Workshop Proceedings, vol. 1887, pp. 16–21. CEUR-WS.org (2017), https://ceur-ws.org/Vol-1887/paper3.pdf
3. Anelli, V.W., Noia, T.D., Sciascio, E.D., Ragone, A., Trotta, J.: Semantic interpretation of top-n recommendations. IEEE Trans. Knowl. Data Eng. **34**(5), 2416–2428 (2022). https://doi.org/10.1109/TKDE.2020.3010215
4. Bałazy, K., Łukasz Struski, Śmieja, M., Tabor, J.: r-softmax: Generalized softmax with controllable sparsity rate (2023)
5. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
6. Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., Huang, F., Si, L., Chen, H.: Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In: Proceedings of the ACM Web Conference 2022, WWW 2022, pp. 2778-2788. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3485447.3511998
7. De Bellis, A.: Structuring the unstructured: an llm-guided transition. In: d'Amato, C., Pan, J.Z. (eds.) Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, 7 November 2023. CEUR Workshop Proceedings, vol. 3678. CEUR-WS.org (2023). https://ceur-ws.org/Vol-3678/paper12.pdf
8. De Bellis, A., et al.: Semantic interpretation of BERT embeddings with knowledge graphs. In: Calvanese, D., et al. (eds.) Proceedings of the 31st Symposium of Advanced Database Systems, Galzingano Terme, Italy, July 2nd to 5th, 2023. CEUR Workshop Proceedings, vol. 3478, pp. 181–191. CEUR-WS.org (2023). https://ceur-ws.org/Vol-3478/paper69.pdf

[2] https://github.com/sisinflab/PRONTO.

9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423
10. Ding, N., Chen, Y., Han, X., Xu, G., Wang, X., Xie, P., Zheng, H., Liu, Z., Li, J., Kim, H.G.: Prompt-learning for fine-grained entity typing. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 6888–6901. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.findings-emnlp.512, https://aclanthology.org/2022.findings-emnlp.512
11. Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H., Liu, Z.: Few-NERD: A few-shot named entity recognition dataset. In: Zong, C., Xia, F., Li, W., Navigli, R. (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), pp. 3198–3213. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.248, https://aclanthology.org/2021.acl-long.248
12. García-Silva, A., Berrío, C., Gómez-Pérez, J.M.: Textual entailment for effective triple validation in object prediction. In: Payne, T.R., et al. (eds.) The Semantic Web - ISWC 2023, pp. 80–100. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-47240-4_5
13. Huang, J., Meng, Y., Han, J.: Few-shot fine-grained entity typing with automatic label interpretation and instance generation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 605-614. KDD '22, Association for Computing Machinery, New York(2022). https://doi.org/10.1145/3534678.3539443
14. Jain, D., Espinosa Anke, L.: Distilling hypernymy relations from language models: On the effectiveness of zero-shot taxonomy induction. In: Nastase, V., Pavlick, E., Pilehvar, M.T., Camacho-Collados, J., Raganato, A. (eds.) Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pp. 151–156. Association for Computational Linguistics, Seattle, Washington (Jul 2022). https://doi.org/10.18653/v1/2022.starsem-1.13, https://aclanthology.org/2022.starsem-1.13
15. Kassner, N., Dufter, P., Schütze, H.: Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3250–3258. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.284, https://aclanthology.org/2021.eacl-main.284
16. Kho, Y., Kim, J., Kang, P.: Boosting prompt-based self-training with mapping-free automatic verbalizer for multi-class classification. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 13786–13800. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.findings-emnlp.921, https://aclanthology.org/2023.findings-emnlp.921

17. Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007. IEEE Computer Society, Los Alamitos, CA, USA (oct 2017). https://doi.org/10.1109/ICCV.2017.324, https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.324
18. Liu, Y., et al.: Roberta: A robustly optimized bert pretraining approach (2019)
19. Peng, H., et al.: COPEN: Probing conceptual knowledge in pre-trained language models. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 5015–5035. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.335, https://aclanthology.org/2022.emnlp-main.335
20. Petroni, F., et al.: Language models as knowledge bases? In: Inui, K., Jiang, J., Ng, V., Wan, X. (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), pp. 2463–2473. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1250, https://aclanthology.org/D19-1250
21. Poerner, N., Waltinger, U., Schütze, H.: E-BERT: Efficient-yet-effective entity embeddings for BERT. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 803–818. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.71, https://aclanthology.org/2020.findings-emnlp.71
22. Qin, G., Eisner, J.: Learning how to ask: querying LMs with mixtures of soft prompts. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 5203–5212. Association for Computational Linguistics, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.410
23. Schick, T., Schütze, H.: Exploiting cloze-questions for few-shot text classification and natural language inference. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 255–269. Association for Computational Linguistics, Online (Apr 2021). https://doi.org/10.18653/v1/2021.eacl-main.20, https://aclanthology.org/2021.eacl-main.20
24. Wu, W., Jiang, C., Jiang, Y., Xie, P., Tu, K.: Do PLMs know and understand ontological knowledge? In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3080–3101. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.173, https://aclanthology.org/2023.acl-long.173
25. Youssef, P., Koraş, O., Li, M., Schlötterer, J., Seifert, C.: Give me the facts! a survey on factual knowledge probing in pre-trained language models. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 15588–15605. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.findings-emnlp.1043, https://aclanthology.org/2023.findings-emnlp.1043