POLITECNICO DI TORINO
Repository ISTITUZIONALE

Open-World Knowledge Graph Completion with Linguistic Noise

Original
Open-World Knowledge Graph Completion with Linguistic Noise / Riva, D., Ferrara, A., Montanelli, S.. - (2026), pp. 106-113. (International Conference on AI x Data and Knowledge Engineering Laguna Hills, CA (USA) 02-04 February 2026) [10.1109/AIxDKE67294.2026.00026].

Availability:
This version is available at: 11583/3012636 since: 2026-07-07T10:59:01Z

Publisher:
IEEE

Published
DOI:10.1109/AIxDKE67294.2026.00026

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

28 November 2026

Open-World Knowledge Graph Completion with Linguistic Noise

Davide Riva
*Department of Computer Science
Università degli Studi di Milano
Milano, Italy
*Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
0009-0003-9681-9423

Alfio Ferrara
Department of Literary Studies, Philology and Linguistics
Università degli Studi di Milano
Milano, Italy
0000-0002-4991-4984

Stefano Montanelli
Department of Computer Science
Università degli Studi di Milano
Milano, Italy
0000-0002-6594-6644

***Abstract*—Reasoning over Knowledge Bases is a well-established task in Artificial Intelligence, but when it comes to perform reasoning in an open-world scenario, in which entities involved in logical statements may be unknown to the reasoner, exact reasoning does not apply. In this work we deal with the case of Open-World Knowledge Graph Completion (KGC): a set of tasks concerning the assessment of the validity of new triples with unseen entities given a pre-existing Knowledge Graph. The use of Large Language Models to represent triples in a vector space is widespread, but most approaches either focus on injecting knowledge in the Language Model at training stage or adopt the LLMs to perform the tasks altogether. The main drawback of such approaches is the wrong prioritization of the LLM's internal knowledge over the information present in the KG, which is also less subject to various forms of linguistic noise. In this paper we study a method to counteract this issue by adopting the Logic Tensor Network framework to learn predicate groundings, defined as standalone neural models, and refine entity embeddings, previously obtained through an LLM. We provide a preliminary investigation of two main properties of such a framework: its potential in performing specifically the Triple Validation task, and its robustness to linguistic noise injected in the entity embeddings. We compare our LTN-based method with a Deep Learning fully supervised approach and an LLM-based approach, finding promising results for further development.**

***Index Terms*—knowledge graphs, knowledge graph completion, logic tensor networks**

I. Introduction

Reasoning over language is a core human ability. Having words in context, we can link them to the concrete entities or abstract concepts they refer to and reason over them, or we can add new entities/concepts to expand our knowledge. For a machine, however, this is not a trivial task. Recent developments in Natural Language Processing (NLP) with the proliferation of Large Language Models (LLMs) make it possible to represent words in a way that mimics their semantics in context as well as to perform reasoning in an internally-consistent way. [1]

One of the most relevant NLP tasks that involves reasoning over language is Knowledge Base Completion (KBC), in which the machine can rely on a background Knowledge Base (KB) made of entities and predicates acting on them, and aims at expanding it without violating consistency with its axioms. Shi and Weninger [2] introduced **Open-World Knowledge Graph Completion** (Open-World KGC, sometimes also referred to as Inductive KGC) as a particular case of KBC applied to Knowledge Graphs (KGs), KBs with at most binary relations and unseen entities[1]. For instance, assuming that we have a KG of animal species which has never seen the entity POLAR BEAR, we may want to insert the triple <POLAR_BEAR, IS_A, MAMMAL>.

Formally, the task is defined as follows: *given a Knowledge Graph* $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$*, where* $\mathcal{E}$ *and* $\mathcal{R}$ *are the entity and predicate sets respectively, and* $\mathcal{T} = \{< e_i, r, e_j >: e_i, e_j \in \mathcal{E}, r \in \mathcal{R}\}$*, expand* $\mathcal{G}$ *by finding a set of missing triples* $\mathcal{T}' = \{< e_i, r, e_j >: e_i, e_j \in \mathcal{E}', r \in \mathcal{R}\}$*, where* $\mathcal{E}' \supset \mathcal{E}$. Such missing triples can be either inferred from the ones in $\mathcal{T}$ or extracted from some external source.

Exploiting LLMs for this task is a widespread approach [2], [4]–[6], but most methodologies either directly apply the LLM to perform zero-shot or few-shot reasoning, or recast the LLM by injecting background knowledge from one or more KGs directly into the language model parameters at training stage. These ways of accounting for background knowledge have

[1]Notes: In the following we will use the terms "relation" and "predicate" interchangeably. Furthermore, as done in the literature, we use "Open-World KGC" and "Inductive KGC" to refer to KGC with unseen entities, while the case of unseen predicates/relations is called "Zero-Shot KGC". [3]

their own drawbacks, the former allowing for conflicts between the internal LLM knowledge and the external, letting the LLM adopt implicit and opaque prioritization rules, the latter contributing to the proliferation of highly specialized LLMs that lack generalization capability.

In this paper, we propose a method that addresses Open-World KGC without re-training nor fine-tuning. We assume to be given new triples, i.e. predicates $r \in \mathcal{R}$ applied on new entities $e'_i, e'_j \in \mathcal{E}'$, obtained by any means, e.g. expert knowledge, entity-relation extraction from text, ontology alignment, etc. We embed triples by transforming them into pseudo-text[2] and feeding them into a small transformer model. We add two neural modules on top of the encoder of such model: an Entity Learner module that maps token embeddings produced by the encoder into entity embeddings, and a Predicate Learner, based on the Logic Tensor Network (LTN) framework [7], that encodes First-Order Logic (FOL) formulas in an embedding space and verifies the satisfiability of new triples with respect to the background knowledge in an approximated way. These modules operate directly on top of pretrained embedding models without the need to inject knowledge in them.

We test two main features of our approach:

1) we verify its ability to perform a crucial KGC sub-task, i.e. Triple Validation (KGC-TV);
2) we verify the robustness of this approach when we add *linguistic noise* to the pseudo-text generated from KG triples. Indeed, due to the contextual nature of the embeddings produced by a transformer model, linguistic noise is relevant when the new triples are extracted from natural language, as in Information Extraction tasks. We experiment specifically with 6 categories of linguistic noise that do not alter the logical validity of the triples: adverbs of affirmation, adverbs of doubt, double negation, logically-neutral adjectives, change of verbal tense, modal verbs.

The rest of this paper is organized as follows: Section II presents previous work on Open-World KGC and on the integration of language and knowledge models, as well as our contributions in such context; Section III presents the methodology in detail; Section IV describes the data, the setting and the results of our experiments; and finally Section V draws the conclusions and sketches future work.

II. Related Work

Literature related to our work encompasses two topics: Open-World KGC and the integration of language and knowledge models.

A. Open-World Knowledge Graph Completion

Knowledge Graphs (KGs) constitute a category of Knowledge Bases (KBs) organized in form of a graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$. Knowledge Graph Completion (KGC) refers to the task of mining from external sources or inferring from $\mathcal{G}$ itself new triples $< e'_i, r', e'_j >$ that are consistent with $\mathcal{G}$, i.e. that don't undermine its satisfiability. With the term **Open-World KGC** [2] we indicate the case in which the relation set is unaltered, $r' \in \mathcal{R}$, while the entity set is expanded to new, unseen entities, i.e. $e'_i, e'_j \in \mathcal{E}'$ with $\mathcal{E}' \supset \mathcal{E}$. A common case of interest in this scenario is the one in which entities are retrieved from textual sources via Information Extraction (IE) systems, requiring downstream validation through reasoning before being inserted in the KG.

Reasoning over KBs is traditionally carried out by symbolic algorithms such as SAT solvers. [8] These algorithms are able to verify the satisfiability of a new formula in an exact way, but cannot deal with entities and/or relations that do not belong to the KB. Furthermore, even in the *Closed-World* scenario, approaches that strive to link entity mentions to KB entities with the aim of reasoning are subject to error propagation from the Entity Linking step. To address these and other issues, approximate solutions based on Machine Learning and fuzzy logics have been recently explored. In the context of KGs, such solutions typically rely on KG Embedding techniques, which map entities and relations from a KG into a vector space where reasoning is carried out through algebraic operations. The first method to do so, TransE [9], is an example of a *translation-based approach* as it evaluates the probability of a triple using the distance between embeddings. *Semantic matching* models such as Logic Tensor Networks (LTNs) [7] or Capsule Networks [10], on the other hand, represent each relation, or n-ary predicate, as a trainable neural model, learning also the embedding of entities in the training process. Both approaches have been proven successful in KGC as well as Open-World KGC, when allowing for the integration of textual data representations from Language Models. [4], [11]

Most approaches of this kind exploit the availability of entity definitions so to use their semantic embeddings in a traditional translation-based approach. For instance, OWE [12] uses plain entity names and contextual definition embeddings, SDT [13] projects non-contextual definition embeddings onto entity type-related subspaces so to exploit hierarchical type information, Caps-OWKG [14] applies a fusion layer and a capsule network on contextual definition embeddings, MIA [15] combines word, part-of-speech, entity and definition contextual embeddings with engineered features to feed multiple attention layers, EmReCo [16] adopts a relation-specific attention mechanism to derive entity embeddings from definition token embeddings. These *definition-based* approaches all share the drawback of assuming the availability of quality definitions, while performance degrades significantly when using entity names only.

Finally, an ever-increasing portion of research focuses on prompt engineering in LLMs to solve the Open-World KGC task. [17] Antecedents to this line of research treat KGC tasks as sequence-to-sequence (seq2seq) problems: KGT5 [18] and KG-S2S [19] address Link Prediction and Question Answering problems training a T5 and an ad-hoc model, respectively, on textual labels of entities. Notable examples of prompt engineering KGC approaches are GenKGC [20], which per-

[2]We use the term "pseudo-text" to refer to text that does not necessarily follow grammatical structure.

forms the Link Prediction task providing examples for the relation to be predicted in the prompt, and KoPA. [21], which translates structural information through a function of entity and relation embeddings into a prefix for the prompt to the LLM that will finally perform the KGC tasks. While powerful, these methods suffer from a lack of efficiency, due to the high hardware requirements to run such LLMs locally, as well as transparency, especially relevant when conflicts arise between the information contained in the KB and the internal knowledge of the model. [1], [22], [23]

B. Integration of Language and Knowledge Models

The integration of language and knowledge models is one of the crucial problems in Artificial Intelligence, as it aims at simultaneously overcoming the intrinsic ambiguity of the former and the stiffness of the latter. Approaches to this problem often rely on the representation of data from different sources in a unified vector space to allow for interaction and fusion, and have been referred to as either "Tensorization" or "Neural : Symbolic $\rightarrow$ Neural" approaches in Neural-Symbolic AI literature. [24]

We may divide *text-oriented* techniques and *knowledge-oriented* techniques. The former are based on the injection of background knowledge in LMs, obtaining Knowledge-enhanced LMs, while the latter exploit LMs to inform knowledge models, e.g. to embed entities in KGs.

In *text-oriented* approaches, knowledge injection is often performed at pre-training stage by the addition of new training objectives to Masked Language Modelling. Models such as KnowBERT [25], ERNIE [26], WKLM [5], LUKE [6], and SPOT [27] are trained to predict entities whose mentions in the text have been masked or replaced, with one or more multi-head attention layers specifically dedicated to this objective. ERICA [28], having Relation Extraction as main purpose, considers link prediction and relation disambiguation with distant supervision from Wikidata as additional training problems.

Knowledge-oriented approaches, on the other hand, typically exploit LMs to embed the textual form(s) of entities and/or relations of a KG. These embeddings may be directly given as input to a neural reasoner such as LTNs [11]. How to synthesize entity embeddings from word or sub-word embeddings is an open problem which has been addressed for instance via Convolutional Neural Networks (CNNs) [2], Contrastive Learning [29], [30], or by selecting the vector representation of special tokens which have been conveniently concatenated to entity and relation textual labels. [4] More recent approaches enrich or compress textual descriptions of entities to extend the power of KG embedding methods. [31] Definition-based Open-World KGC approaches may also belong to this category, though the range of addressed tasks is broader here.

Finally, some *hybrid* models are built to deal with both language and knowledge-related tasks. KEPLER [32], for instance, joins the original objectives of masked language modelling and KG embedding, exploiting the availability of textual descriptions of entity and relations to link the two. GREASELM [33], instead, lies on parallel training of two models, a Transformer architecture for language modelling and a Graph Neural Network (GNN) for KG embedding, designing a special token and a special node to allow for deep and mutual interaction.

C. Contributions

While aforementioned models have outperformed plain pre-trained LMs on many language and/or knowledge tasks, including Open-World KGC, they share the main disadvantage of involving data integration at training stage, which makes it necessary to re-train the entire model when dealing with different KGs or changing the LM architecture the model is based upon. Our method aims at avoiding this drawback by adding entity embedding refinement, through an Entity Learner module, and relational reasoning, through a Predicate Learner module, on top of a pre-trained small LM instead of inside it, requiring a relatively small parameter overhead. At the same time, training such modules following the LTN framework, which injects logical reasoning in each predicate model, allows to completely avoid error propagation from explicit Entity Linking by relying on fuzzy neural reasoners. Furthermore, such modules, plus the adoption of multilingual and polysemy-robust LM encoder as base textual embedder, help mitigate the reliance on textual forms of entities, as they induce relational semantics (i.e. similarity of entities in the same place of the same relation) to emerge rather than imposing it.

Most importantly, we aim to propose a framework that does not rely on the availability of textual definitions and challenges the predictive power of LLMs, while clearly separating the influence of the LM internal knowledge, implicit in entity embeddings, and the KG-induced reasoning ability, learnt by the predicate models.

III. Methodology

Our approach is based on the transformation of triples $t =< e_i, r, e_j >$ from a KG into pseudo-text, which is simply the concatenation $\tilde{t} = e_i \oplus r \oplus e_j$ of the textual labels associated to entities and predicates. For instance, the triple <Polar Bear, Is A, Mammal> is transformed into the sentence *Polar bear is a Mammal* and the triple <Hour, Has Part, Minute> is transformed into *Hour has part minute*.

The resulting sentences are then embedded in a d-dimensional vector space via a pretrained Small LM[3] denoted by M. Upon M two modules are added:

- an **Entity Learner** L_e that constructs entity embeddings $(\mathbf{e}_i, \mathbf{e}_j)$ from token embeddings $\{\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{n(i)}^{(i)}\}, \{\mathbf{x}_1^{(j)}, \ldots, \mathbf{x}_{n(j)}^{(j)}\}$, where $n(\cdot)$ denotes the number of tokens per entity. This consists of a self-attention layer trained with a contrastive objective on the entities appearing in the KG;
- a **Relation Learner** L_r, consisting of a Logic Tensor Network that represents each relation $r \in \mathcal{R}$ as a feedforward neural network model f_r. Given a triple

[3]We refer to Small LMs as LMs with a number of parameters $|M| < 10^9$.

$t =< e_i, r, e_j >$, or a logical formula containing more triples linked to one another by logical connectives, L_r maps the embedding pair $(\mathbf{e}_i, \mathbf{e}_j)$ onto a truth value $y \in [0; 1]$ evaluating first each atomic relation and then the FOL connectives $(\neg, \vee, \wedge)$ and quantifiers $(\forall, \exists)$.

The overall architecture and workflow are summarized in Figure 1.

We first notice that the overall number of parameters $|L_e| + |L_r|$ is in the order of $O(2dRh)$ with $O(2dh)$ active parameters at a time, where $R = |\mathcal{R}|$, d is the entity embedding size, and h is the depth of the relation neural models, whereas fine-tuning, even with Parameter-Efficient Fine-Tuning (PEFT) techniques, typically requires $O(Edh's)$, with $E = |\mathcal{E}|$ and s being the sparsity of the fine-tuning method. Assuming an equal depth $h' = h$, our approach is parameter-efficient for $R < \frac{1}{2}Es$, which is the case for most large-enough KGs and a variety of PEFT techniques.

Finally, we describe how we inject linguistic noise into the test data by pre-defining a set of linguistic rules that alter the pseudo-text generated from triples.

A. Entity Learner

The Entity Learner is responsible for transforming token vector sequences, encoded by model M, into a single vector for each entity. Calling $x_1^{(i)}, \ldots, x_{n(i)}^{(i)}$ the tokens that make up an entity mention e_i in a text, with embeddings $\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{n(i)}^{(i)}$, then the Entity Learner is a function $L_e : X_d \to \mathbb{R}^d$, where X_d is the set of variable-length sequences of vectors in $\mathbb{R}^d$.

An approach as simple as mean pooling has often been adopted to summarize variable-length sequences of embeddings into the single embedding vector $\mathbf{e}_i$. Here, in order to enhance the flexibility of our approach, a self-attention layer is trained to produce a weighted mean of the vector sequence. Formally, for an entity e_i:

$$\mathbf{A}_i = \frac{1}{\sqrt{d}} \mathbf{W_Q} \mathbf{1} (\mathbf{W_K} \mathbf{X^{(i)}})^\top \tag{1}$$

$$\mathbf{e}_i = \mathbf{A}_i \mathbf{W_V} \mathbf{X}^{(i)} \tag{2}$$

where $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$ are trainable matrices and $\mathbf{1}$ indicates a conveniently-shaped matrix of ones.

Exploiting the fact that entities in a KG are unambiguous, and thus two equal entity mentions refer to the same entity, we train the attention layer with a contrastive objective: we first extract pairs of equal entity mentions (e_i, e'_i), which in principle have different embeddings, then perform negative sampling to find, for each pair in the training set, a different entity e_k which never appears in the same triples as e_i, i.e. with the same relation and the same other entity involved.

We train the Entity Learner to minimize the margin loss:

$$l(e_i, e'_i, e_k) = max(\delta_{cos}(\mathbf{e}_i, \mathbf{e'}_i) - \delta_{cos}(\mathbf{e}_i, \mathbf{e}_k) + \mu, 0) \tag{3}$$

where μ is the margin we want to reach between the two cosine distances, indicated by $\delta_{cos}(\cdot, \cdot)$. Furthermore, the gradient of the Predicate Learning loss also acts on the entity embeddings with the same weight as the contrastive loss.

In the example, the transformer model may output a vector for token *pol*, another for the token *##ar* and another one for token *bear*. The Entity Learner will produce a single vector $\mathbf{e}_{Polarbear}$ from them. Notice the vector is still contextual, in the sense that a different occurrence of the entity *Polar bear* will be actually mapped into a different vector, but the model is trained to produce similar vectors if the token embeddings are themselves similar.

B. Predicate Learner

The Predicate Learner aims at learning one function f_r per predicate r that maps pairs of entity embeddings $(\mathbf{e}_i, \mathbf{e}_j)$ (or single entity embeddings if r is unary) into a satisfiability score $y \in [0; 1]$. In the spirit of Logic Tensor Networks (LTN), a relation of arity 2 is represented as a feedforward neural network comprising: a positional encoding layer, a feature expansion layer which concatenates $[\mathbf{e}_i, \mathbf{e}_j, \frac{1}{\sqrt{d}}\mathbf{e}_i \odot \mathbf{e}_j, tanh|\mathbf{e}_i - \mathbf{e}_j|]$, an input normalized projection layer, and finally a Multi-Layer Perceptron (MLP) with final sigmoid activation. A relation of arity 1, as a consequence, is mapped into a similar function without the positional encoding and the feature expansion layers.

The model is trained to maximize the likelihood y of a set of formulas $\Phi^{\mathcal{T}} = \{\phi_1, \ldots, \phi_N\}$ derived directly from KG triples $\mathcal{T}$ and negative samples of false formulas $\Phi^{\neg\mathcal{T}} = \{\phi_1^\neg, \ldots, \phi_N^\neg\}$ derived by mutual exclusion (i.e. non-contradiction) from $\mathcal{T}$. For the time being, the derivation work is done by hand, the inclusion of automated and more complex logical inference is left to future work. For instance, the triple <POLAR BEAR, IS A, MAMMAL> is translated into the formula MAMMAL(POLAR BEAR) since the objects of predicate IS A can be treated as unary predicates, i.e. superclasses. As IS A is mutually exclusive for the specific KG construction, we can easily derive additional rules Φ^+ such as MAMMAL(POLAR BEAR) $\Rightarrow$ ANIMAL(POLAR BEAR). Similarly, for triple <HOUR, HAS PART, MINUTE> we can derive the formulas HASPART(HOUR, MINUTE) as well as HASPART(HOUR, MINUTE) $\Rightarrow \neg$ HASPART(MINUTE, HOUR).

Logical connectives and quantifiers from FOL can be handled with fuzzy logic operators as in the LTN framework. [34] In our experiments, we adopt Van Krieken's configuration of fuzzy logical operators as the best trade-off between logical semantics and functional differentiability, [34] thus transforming the output y_i from a formula ϕ_i, containing one or more f_r models, in the logit space instead of the probability space. The definitions are the following:

$$y_1 \wedge y_2 := y_1 + y_2 - ln(e^{y_1} + e^{y_2}) \qquad \forall y_i := \sum_i y_i$$

$$y_1 \vee y_2 := ln(e^{y_1} + e^{y_2} + e^{y_1 + y_2}) \qquad \exists y_i := sign(\sum_i y_i) \cdot \sqrt{(\sum_i y_i^2)}$$

$$y_1 \Rightarrow y_2 := ln(e^{-y_1} + e^{y_2} + e^{y_2 - y_1}) \qquad \neg y := -y$$

Finally the logit SAT values of different formulas are aggregated via a $\forall$ aggregator and an exponential function.

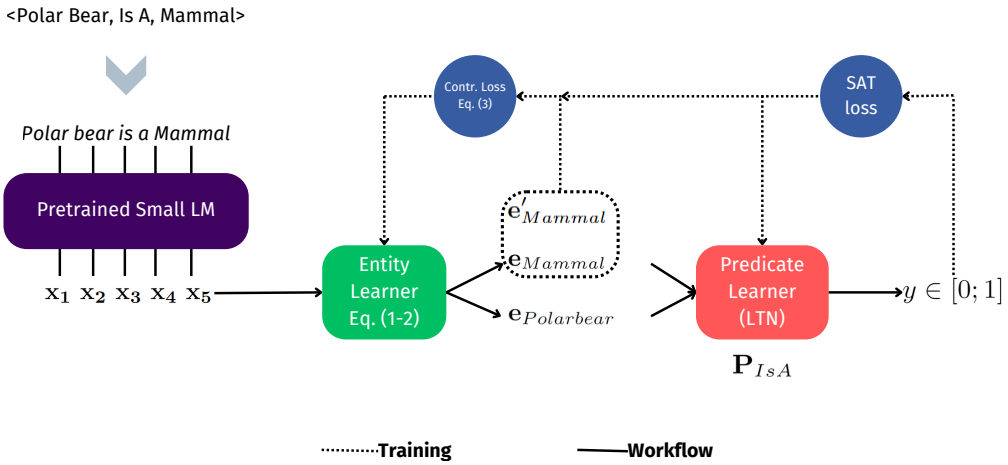


Fig. 1: The architecture and workflow of the overall approach. For simplicity, we hide the fact that there is one predicate learner for each predicate. The entity embedding $\mathbf{e}'_{Mammal}$ comes from the Entity Learner applied on another triple. SAT loss is a convenient way to say 1 - SAT.

C. Triple Validation

KGC-TV is one of the most simple but also of the most crucial KGC sub-tasks. In our method, it amounts to verifying the output probability of a given triple $\mathbb{P}(< e'_i, r, e'_j >) = f_r(\mathbf{e}'_i, \mathbf{e}'_j)$ via its corresponding predicate model. For unary triples $\mathbb{P}(< s, r, o >) = f_o(\mathbf{s})$. The task then works as a binary classification task where we set the rounding threshold at $1/2$ for simplicity.

D. Linguistic Noise

One of the purposes of our analysis is to evaluate the robustness of our approach in situations that induce noise in the entity embeddings, i.e. to test the robustness of our approach in a noisy environment. Instead of perturbating embeddings artificially from a known distribution, we induce noise by injecting specific and realistic linguistic elements into the pseudo-text generated from test triples. This way the noise is non-parametric and akin to the real-world noise intrinsic in textual data processing. We group linguistic elements into 6 categories, each of which contains a list of terms or modifications that are characterized for not affecting the logical semantics (in terms of FOL) of the sentence while changing its phrasal, intentional and cognitive semantics:

- adverbs of affirmation (AA): *definitely, certainly, for sure, clearly, obviously, etc*;
- adverbs of doubt (DA): *probably, likely, maybe, perhaps, with some confidence, etc*;
- double negation (DN): *is not a non*, *does not ... non*;
- logically-neutral adjectives (NA): *nice, interesting, wonderful, awful, mysterious, etc*;
- change of verbal tense (VT): *was, has been, will be* instead of *is*.
- modal verbs (MV): *may, might, could, must, ought to*.

When applying noise from one of these categories to a pseudo-text proposition, we draw one term or modification at random from the corresponding list.

We don't include frequency adverbs (e.g. *sometimes, rarely, ...*) as they affect the logical semantics of the pseudo-sentences. One may argue that also adverbs of doubt, change of verbal tense, and modal verbs might change the logical semantics of the sentences. Despite the existence of modal logics and temporal logics, for the moment being these categories are still included in linguistic noise. In particular, we consider the reasoner as not dealing with modal nor temporal predicates. Furthermore, investigating whether the injection of doubt and uncertainty in the textual data affects the reasoner's outcome fits within the scope of the present analysis.

In order to differentiate the impact of such categories from the impact of simply adding information, we experiment also with mixed patterns: if the model values the quantity of information rather than its quality, it will perform better on cases in which more words are added, regardless of the content. In order to assess whether this is the case, we explore two combinations that add quantity of text: *adverbs of affirmation + adverbs of doubt*, and *modal verbs + adverbs of affirmation + adverbs of doubt*.

IV. Experiments

In this section we describe the data, the baselines and the experimental setting as well as the results of our experiments.

A. Data

We perform experiments on two datasets that contain unary and binary relations:

- Bianchi et al[4]: taxonomy of living organisms from DBPedia, containing only taxonomical relations. We address only setting (3) mentioned in their work, as it is the most relevant to the case of Open-World KGC. In this setting, the training set is made of 1756 triples, of which 941 positive and 815 negative ones. The test set contains

[4]https://github.com/vinid/logical_commonsense

33390 propositions: 17361 positive and 16029 negative ones; [11]

- WN18RR[5]: dataset from English WordNet with 18 predicates. For the purpose of this analysis, the dataset is filtered to keep only the predicates that appear at least in 5% of the triples, resulting in 4 predicates. Finally, the dataset is re-arranged so to have 10305 triples (5087 positive and 5218 negative) in the training set and 92753 triples (46442 positive and 46311 negative) in the test set; [9]

The main advantage of these datasets is that they offer a real open-world scenario. Indeed, the test set is almost always much larger than the training set, and the distribution of entities differs significantly between the training and the test set, allowing for the actual verification of the generalization capability of the model. In table I we present a detailed view of occurrences of each predicate and the number of distinct entities involved in triples with that predicate in each dataset, together with the percentage of unseen entities in the test set.

Dataset	Unseen Test	Predicate	# pred.	# subj.	# obj.
Bianchi et al	70%	IS A	35133	5019	7
WN18RR	62%	HYPERNYM	71726	27026	7658
		M.MERONYM	15296	3151	7506
		HAS PART	9970	1955	3878
		I.HYPERNYM	6066	2365	402

TABLE I: Predicate statistics. # head and # tail refer to the number of distinct entities involved in triples with a specific predicate as a head and as an tail respectively.

B. Baselines

We compare our approach against two baselines: a semantic matching model and a prompt-based approach. The former uses ConMask [2] as deep learning model constructed for Open-World KGC which addresses the Link Prediction and KGC-TV sub-tasks, differing from our approach in that ConMask uses embedding vector averaging as entity learning and a single deep convolutional network instead of multiple predicate models in the LTN framework. The latter - instead - exploits a LLM (GPT-5-mini) both with the following plain prompt: "*You are given the sentence $\tilde{t}$, from which the triple t is extracted. Please discriminate whether the triple is true. Answer only yes or no.*".

C. Experimental Setting

In our experiments we adopt `granite-embedding-278m-multilingual`[6] as embedding model M, which has an embedding dimension $d = 768$ and can easily run even on a single GPU.[7] The Entity Learner L_e is trained with margin $\mu = 1.0$ and each relation model f_r with arity α_r has hidden layers with SiLU activations and sizes $\left(\lfloor \alpha_r^2 \frac{d}{10} \rfloor, \frac{d}{\beta}, \frac{d}{\beta^2}, \ldots, 1\right)$, where the first layer has the task to mix features from the feature expansion layer into a low rank representation and optimal β is such that the overall number of trainable parameters doesn't exceed training set size, e.g. $\beta = 8$ for Bianchi et al dataset. Predicates with $\alpha_r > 1$ are also equipped with rotational positional encodings. Models are trained for 100 epochs with AdamW optimizer setting learning rate $\eta = 10^{-3}$ and cosine annealing scheduler.

[5]https://github.com/ZhenfengLei/KGDatasets

[6]https://huggingface.co/ibm-granite/granite-embedding-278m-multilingual

[7]All experiments were run on a Ubuntu 24.04 machine endowed with one H100 GPU and CUDA 13.1. Code will be made available upon request due to concurrency with other works. In the specified setting, the average training time is 16 ms/triple and test time is in the order of <1 ms/triple for KGC-TV.

D. Results

Since the KGC-TV task is a binary classification task, we measure results in terms of F1 scores and their variability. The summary of results is shown in Table II.

Interestingly, results seem to point to opposite directions in the two datasets. In Bianchi et al, which contains only unary predicates and encyclopedic knowledge, GPT-5-mini largely outperforms both our method and ConMask, displaying also a good robustness to the envisioned forms of linguistic noise. In WN18RR, which contains only binary predicates and lexical knowledge, it often fails in comparison to our method, which also shows the lowest variability against linguistic noise. Looking more closely, GPT-5-mini struggles whenever doubt adverbs and modal verbs are present, situations in which our approach seems to improve instead of degrading.

This outcome is in line with what we discussed about separation between internal LM knowledge, external KG knowledge and linguistic information. Being able by construction to distinguish between the three, with LM knowledge and linguistic information encapsulated in token embeddings then mitigated by the Entity Learner and KG knowledge prioritized by the Predicate Learner, our proposed method gains where the syntax may convey an ambiguous meaning. Note we do not claim it is generally wrong to predict that, for instance, the triple <VILLA, HAS_HYPERNYM, HOUSE> is false given the pseudo-text "*Villa maybe has hypernym house*". The results show only that, while GPT-5-mini has a tendency to replicate this behavior, our method shows promising results in counteracting this tendency.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel approach to Open-World KGC that operates without retraining or fine-tuning LLMs. Our method leverages pseudo-text transformation of triples, a small transformer encoder, and two specialized neural modules: an Entity Learner and a LTN-based Predicate Learner that verifies triple satisfiability through FOL formulas in embedding space.

Our experimental evaluation on the Triple Validation subtask revealed that, while our approach did not entirely surpass the performance of GPT-5-mini, it demonstrated competitive results and exhibited particularly promising behavior in noisy environments. This noise-resilience is especially valuable for real-world applications where triples are extracted from natural language sources through Information Extraction pipelines, where such linguistic variations are commonplace.

Dataset	Model	P	AA	DA	DN	NA	VT	MV	AA+DA	MV+AA+DA	IQR^{-1}
Bianchi et al.	Ours	0.639	0.600	0.515	0.605	0.685	0.532	0.639	0.515	0.687	9.346
	ConMask	0.624	0.648	0.551	0.700	0.665	0.549	0.635	0.647	0.626	**41.67**
	GPT-5-mini	**0.833**	**0.837**	**0.804**	**0.867**	**0.835**	**0.828**	**0.803**	**0.803**	**0.787**	31.25
WN18RR	Ours	0.586	0.548	**0.667**	0.536	0.607	0.561	**0.625**	**0.642**	**0.590**	**15.63**
	ConMask	0.492	0.542	0.649	0.487	0.628	0.473	0.514	0.617	-	7.752
	GPT-5-mini	**0.731**	**0.752**	0.539	**0.799**	**0.732**	**0.750**	0.539	0.533	0.576	4.739

TABLE II: F1 scores for our approach and the two baselines in each linguistic noise setting (P = Plain, AA = Affirmative Adverbs, DA = Doubt Adverbs, DN = Double Negation, NA = Neutral Adjectives, VT = Verb Tense, MV = Modal Verb). IQR^{-1} is the inverse of the inter-quartile range for the results, used as a measure of robustness against the forms of linguistic noise included in this study. The best result for each dataset is in bold.

The architectural choice of adding a small parameter overhead through specialized modules on top of a general purpose small LM offers several advantages: it avoids conflicts between internal and external knowledge, provides more transparent reasoning mechanisms, and prevents the proliferation of over-specialized models that lack generalization capability.

However, our results also highlight important limitations. The performance gap with state-of-the-art LLMs suggests that there remains room for improvement in how our Entity and Predicate Learner modules capture and reason over semantic relationships. Additionally, our current evaluation focuses solely on Triple Validation, leaving other crucial KGC sub-tasks unexplored. Future work will address these limitations through several directions. First, we plan to conduct more comprehensive evaluations across diverse datasets, including domain-specific KGs. Second, we aim to extend our framework to handle additional KGC sub-tasks, particularly Link Prediction and Relation Prediction, by integrating vector search technologies that can efficiently retrieve candidate entities and relations without training ad-hoc inference modules. Third, we intend to enhance the Predicate Learner module by expanding rule inference capabilities prior to training, enabling the system to automatically leverage trivial logical consequences of KB facts.

Despite not achieving state-of-the-art performance, our approach represents a meaningful step toward more modular, interpretable, and noise-robust KGC systems that can bridge the gap between symbolic reasoning and neural language understanding in an Open-World scenario.

ACKNOWLEDGMENT

This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 118/2023.

REFERENCES

[1] B. Bi, S. Liu, Y. Wang, Y. Xu, J. Fang, L. Mei, and X. Cheng, "Parameters vs. context: Fine-grained control of knowledge reliance in language models," *arXiv preprint arXiv:2503.15888*, 2025.

[2] B. Shi and T. Weninger, "Open-world knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[3] F. Cornell, C. Zhang, J. Karlgren, and Š. Girdzijauskas, "Challenging the assumption of structure-based embeddings in few-and zero-shot knowledge graph completion," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6300–6309, 2022.

[4] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proceedings of the Web Conference 2021*, pp. 1737–1748, 2021.

[5] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model," 2020.

[6] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "LUKE: Deep contextualized entity representations with entity-aware self-attention," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 6442–6454, Association for Computational Linguistics, Nov. 2020.

[7] L. Serafini and A. S. d'Avila Garcez, "Learning and reasoning with logic tensor networks," in *Conference of the Italian Association for Artificial Intelligence*, pp. 334–348, Springer, 2016.

[8] K. Claessen, N. Een, M. Sheeran, and N. Sorensson, "Sat-solving in practice," in *2008 9th International Workshop on Discrete Event Systems*, pp. 61–67, IEEE, 2008.

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[10] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[11] F. Bianchi, M. Palmonari, P. Hitzler, and L. Serafini, "Complementing logical reasoning with sub-symbolic commonsense," in *Rules and Reasoning: Third International Joint Conference, RuleML+ RR 2019, Bolzano, Italy, September 16–19, 2019, Proceedings 3*, pp. 161–170, Springer, 2019.

[12] H. Shah, J. Villmow, A. Ulges, U. Schwanecke, and F. Shafait, "An open-world extension to knowledge graph completion models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 3044–3051, 2019.

[13] X. Chen, S. Jia, L. Ding, H. Shen, and Y. Xiang, "Sdt: An integrated model for open-world knowledge graph reasoning," *Expert Systems with Applications*, vol. 162, p. 113889, 2020.

[14] Y. Wang, W. Xiao, Z. Tan, and X. Zhao, "Caps-owkg: a capsule network model for open-world knowledge graph," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 1627–1637, 2021.

[15] L. Niu, C. Fu, Q. Yang, Z. Li, Z. Chen, Q. Liu, and K. Zheng, "Open-world knowledge graph completion with multiple interaction attention," *World Wide Web*, vol. 24, pp. 419–439, 2021.

[16] J. Wang, J. Lei, S. Sun, and K. Guo, "Embeddings based on relation-specific constraints for open world knowledge graph completion," *Applied Intelligence*, vol. 53, no. 12, pp. 16192–16204, 2023.

[17] L. Yao, J. Peng, C. Mao, and Y. Luo, "Exploring large language models for knowledge graph completion," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2025.

[18] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-sequence knowledge graph completion and question answering," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (S. Muresan, P. Nakov, and A. Villavicencio, eds.), (Dublin, Ireland), pp. 2814–2828, Association for Computational Linguistics, May 2022.

[19] C. Chen, Y. Wang, B. Li, and K.-Y. Lam, "Knowledge is flat: A Seq2Seq generative framework for various knowledge graph completion," in *Proceedings of the 29th International Conference on Computational Linguistics* (N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner,

K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, eds.), (Gyeongju, Republic of Korea), pp. 4005–4017, International Committee on Computational Linguistics, Oct. 2022.

[20] X. Xie, N. Zhang, Z. Li, S. Deng, H. Chen, F. Xiong, M. Chen, and H. Chen, "From discrimination to generation: Knowledge graph completion with generative transformer," *Companion Proceedings of the Web Conference 2022*, 2022.

[21] Y. Zhang, Z. Chen, L. Guo, Y. Xu, W. Zhang, and H. Chen, "Making large language models perform better in knowledge graph completion," in *Proceedings of the 32nd ACM international conference on multimedia*, pp. 233–242, 2024.

[22] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, and J. Xu, "Knowledgeable or educated guess? revisiting language models as knowledge bases," *arXiv preprint arXiv:2106.09231*, 2021.

[23] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*, pp. 15696–15707, PMLR, 2023.

[24] K. Hamilton, A. Nayak, B. Božić, and L. Longo, "Is neuro-symbolic ai meeting its promises in natural language processing? a structured review," *Semantic Web*, no. Preprint, pp. 1–42, 2022.

[25] M. E. Peters, M. Neumann, I. RobertL.Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[26] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 1441–1451, Association for Computational Linguistics, July 2019.

[27] J. Li, Y. Katsis, T. Baldwin, H.-C. Kim, A. Bartko, J. McAuley, and C.-N. Hsu, "Spot: Knowledge-enhanced language representations for information extraction," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1124–1134, 2022.

[28] Y. Qin, Y. Lin, R. Takanobu, Z. Liu, P. Li, H. Ji, M. Huang, M. Sun, and J. Zhou, "ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning," in *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)* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), (Online), pp. 3350–3363, Association for Computational Linguistics, Aug. 2021.

[29] L. Kong, C. de Masson d'Autume, W. Ling, L. Yu, Z. Dai, and D. Yogatama, "A mutual information maximization perspective of language representation learning," *ArXiv*, vol. abs/1910.08350, 2019.

[30] L. Wang, W. Zhao, Z. Wei, and J. Liu, "Simkgc: Simple contrastive knowledge graph completion with pre-trained language models," in *Annual Meeting of the Association for Computational Linguistics*, 2022.

[31] R. Yang, J. Zhu, J. Man, L. Fang, and Y. Zhou, "Enhancing text-based knowledge graph completion with zero-shot large language models: A focus on semantic enhancement," *Knowledge-Based Systems*, vol. 300, p. 112155, 2024.

[32] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[33] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. D. Manning, and J. Leskovec, "Greaselm: Graph reasoning enhanced language models," in *International conference on learning representations*, 2021.

[34] E. van Krieken, E. Acar, and F. van Harmelen, "Analyzing differentiable fuzzy logic operators," *Artificial Intelligence*, vol. 302, p. 103602, 2022.