LREC-COLING 2024

# DLnLD: Deep Learning and Linked Data @LREC-COLING-2024

Workshop Proceedings

Editors
Gilles Sérasset, Hugo Gonçalo Oliveira and Giedre Valunaite Oleskeviciene

21 May, 2024
Torino, Italia

# Proceedings of the Workshop on DLnLD: Deep Learning and Linked Data @LREC-COLING-2024

Copyright ELRA Language Resources Association (ELRA), 2024
These proceedings are licensed under a Creative Commons
Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0)

ISBN 978-2-493814-16-6
ISSN 2951-2093 (COLING); 2522-2686 (LREC)

Jointly organized by the ELRA Language Resources Association
and the International Committee on Computational Linguistics

# Preface by the Program Chairs

Since the appearance of transformers (Vaswani et al., 2017), Deep Learning (DL) and neural approaches have brought a huge contribution to Natural Language Processing (NLP) either with highly specialized models for specific application or via Large Language Models (LLMs) (Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023) that are efficient few-shot learners for many NLP tasks. Such models usually build on huge web-scale data (raw multilingual corpora and annotated specialized, task related, corpora) that are now widely available on the Web. This approach has clearly shown many successes, but still suffers from several weaknesses, such as the cost/impact of training on raw data, biases, hallucinations, lack of explainability, among others (Nah et al., 2023).

The Linguistic Linked Open Data (LLOD) (Chiarcos et al., 2013) community aims at creating/distributing explicitly structured data (modelled as RDF graphs) and interlinking such data across languages. This collection of datasets, gathered inside the LLOD Cloud (Chiarcos et al., 2020), contains a huge amount of multilingual ontological (e.g. DBpedia (Lehmann et al., 2015)); lexical (e.g., DBnary (Sérasset, 2015), Wordnet (McCrae et al., 2014), Wikidata (Vrandečić and Krötzsch, 2014)); or linguistic (e.g., Universal Dependencies Treebank (Nivre et al., 2020; Chiarcos et al., 2021), DBpedia Abstract Corpus (Brümmer et al., 2016)) information, structured using common metadata (e.g., OntoLex (McCrae et al., 2017), NIF (Hellmann et al., 2013), etc.) and standardised data categories (e.g., lexinfo (Cimiano et al., 2011), OliA (Chiarcos and Sukhareva, 2015)).

Both communities bring striking contributions that seem to be highly complementary. However, if knowledge (ontological) graphs are now routinely used in DL, there is still very few research studying the value of Linguistic/Lexical knowledge in the context of DL. We think that, today, there is a real opportunity to bring both communities together to take the best of both worlds. Indeed, with more and more work on Graph Neural Networks (Wu et al., 2023) and Embeddings on RDF graphs (Ristoski et al., 2019), there is more and more opportunity to apply DL techniques to build, interlink or enhance Linguistic Linked Open Datasets, to borrow data from the LLOD Cloud for enhancing Neural Models on NLP tasks, or to take the best of both worlds for specific NLP use cases.

This led us to propose this workshop aims at gathering researchers that work on the interaction between DL and LLOD in order to discuss what each approach has to bring to the other. All application domains (Digital Humanities, FinTech, Education, Linguistics, Cybersecurity…) as well as approaches (NLG, NLU, Data Extraction…) were welcome, provided that the work is based on the use of BOTH Deep Learning techniques and Linguistic Linked (meta)Data.

The DLnLD workshop builds on four editions of previous workshops, namely:

- Workshop on Deep Learning and Neural Approaches for Linguistic Data, collocated with the 3rd Nexus Linguarum Plenary Meeting, in Skopje, North Macedonia and online, September 2021;
- Workshop on Linguistic Knowledge Processing with Deep Learning, hosted at the Nexus Workshop days in Jerusalem, Israel, May 2022;
- 2nd Workshop on Deep Learning and Neural Approaches for Linguistic Linked Data, collocated with the LLOD Approaches for Language Data Research and Management Conference (LLODREAM2022), in Vilnius, Lithuania, and online, September 2022;

- Workshop on Deep Learning, Relation Extraction and Linguistic Data, collocated with the Language, Data and Knowledge Conference (LDK), in Vienna, Austria, September 2023.

However, in DLnLD, the objectives were expanded to not only study how Deep Learning may be used **for** Linguistic Linked Data but also explore how Linguistic Linked Data may be leveraged by Deep Learning approaches.

The papers that are presented in this volume show that the two domains do indeed cross fertilize each other with researchers using Language Models for Linguistic Linked Data modelling or generation and others leveraging Linked Data for evaluation or post-hoc verification of LLM outputs, while others do study Graph Neural Networks as a mean to merge both worlds in specific use cases.

We do think that this is only a beginning and that research will continue towards a better entanglement of both worlds and hope this workshop only witnesses the beginning of a research trend.

This workshop is organised in the scope of COST Action CA18209 NexusLinguarum[1], supported by COST (European Cooperation in Science and Technology).

Gilles Sérasset,
Hugo Gonçalo Oliveira,
Giedre Valunaite Oleskeviciene

---

[1] `https://nexuslinguarum.eu/`

# Organizing Committee

## Workshop Chairs

- Gilles Sérasset, *Université Grenoble Alpes, France*
- Hugo Gonçalo Oliveira, *University of Coimbra, Portugal*
- Giedre Valunaite Oleskeviciene, *Mykolas Romeris University, Lithuania*

## Program Committee

- Mehwish Alam, *Télécom Paris, Institut Polytechnique de Paris, France*
- Milana Bolatbek, *Al-Farabi Kazakh National University, Kazakhstan*
- Milan Dojchinovski, *Czech Technical University in Prague, Czech Republic*
- Basil Ell, *University of Oslo, Norway*
- Radovan Garabík, *L'. Štúr Institute of Linguistics, Slovak Academy of Sciences, Slovakia*
- Katerina Gkirtzou, *Athena Research Center, Maroussi, Greece*
- Jorge Gracia del Río, *University of Zaragoza, Spain*
- Dagmar Gromann, *University of Vienna, Austria*
- Dangis Gudelis, *Mykolas Romeris University, Lithuania*
- Ilan Kernerman, *Lexicala by K Dictionaries, Israel*
- Chaya Liebeskind, *Jerusalem College of Technology, Israel*
- Marco C. Passarotti, *Università Cattolica del Sacro Cuore, Milan, Italy*
- Heiko Paulheim, *University of Mannheim, Germany*
- Alexandre Rademaker, *IBM Research Brazil and EMAp/FGV, Brazil*
- Georg Rehm, *DFKI GmbH, Berlin, Germany*
- Didier Schwab, *Université Grenoble Alpes, France*
- Andon Tchechmedjiev, *IMT Mines Alès, France*
- Dimitar Trajanov, *Ss. Cyril and Methodius University – Skopje, Macedonia*
- Nicolas Turenne, *Guangdong University of Foreign Studies, China*

# Table of Contents

*Investigating the Impact of Different Graph Representations for Relation Extraction with Graph Neural Networks*
Moritz Blum, Gennaro Nolano, Basil Ell and Philipp Cimiano . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

*TaxoCritic: Exploring Credit Assignment in Taxonomy Induction with Multi-Critic Reinforcement Learning*
Injy Sarhan, Bendegúz Toth, Pablo Mosteiro and Shihan Wang . . . . . . . . . . . . . . . . . . . . . . . 14

*Combining Deep Learning Models and Lexical Linked Data: Some Insights from the Development of a Multilingual News Named Entity Recognition and Linking Dataset*
Emmanuel Cartier and Emile Peetermans . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

*Deductive Verification of LLM Generated SPARQL Queries*
Alexandre Rademaker, Guilherme Lima, Sandro Rama Fiorini
and Viviane Torres da Silva . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

*How to Turn Card Catalogs into LLM Fodder*
Mary Ann Tan, Shufan Jiang and Harald Sack . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

*Evaluating Large Language Models for Linguistic Linked Data Generation*
Maria Pia di Buono, Blerina Spahiu and Verginica Barbu Mititelu . . . . . . . . . . . . . . . . . . . . . . . 66

*Towards Automated Evaluation of Knowledge Encoded in Large Language Models*
Bruno Carlos Luís Ferreira, Hugo Gonçalo Oliveira and Catarina Silva . . . . . . . . . . . . . . . . . 76

*Self-Evaluation of Generative AI Prompts for Linguistic Linked Open Data Modelling in Diachronic Analysis*
Florentina Armaselu, Chaya Liebeskind and Giedre Valunaite Oleskeviciene . . . . . . . . . . . 86

# Workshop Program

**Tuesday May 21, 2024**

| | |
|---|---|
| **09:00–10:30** | **Session 1**<br>Chair: Hugo Gonçalo Oliveira |
| 9:00–9:20 | *Investigating the Impact of Different Graph Representations for Relation Extraction with Graph Neural Networks*<br>Moritz Blum, Gennaro Nolano, Basil Ell and Philipp Cimiano |
| 9:20–9:40 | *TaxoCritic: Exploring Credit Assignment in Taxonomy Induction with Multi-Critic Reinforcement Learning*<br>Injy Sarhan, Bendegúz Toth, Pablo Mosteiro and Shihan Wang |
| 9:40–10:00 | *Combining Deep Learning Models and Lexical Linked Data: Some Insights from the Development of a Multilingual News Named Entity Recognition and Linking Dataset*<br>Emmanuel Cartier and Emile Peetermans |
| 10:00–10:20 | *Deductive Verification of LLM Generated SPARQL Queries*<br>Alexandre Rademaker, Guilherme Lima, Sandro Rama Fiorini and Viviane Torres da Silva |
| **11:00–13:00** | **Session 2**<br>Chair: Gilles Sérasset |
| 11:00–11:20 | *How to Turn Card Catalogs into LLM Fodder*<br>Mary Ann Tan, Shufan Jiang and Harald Sack |
| 11:20–11:40 | *Evaluating Large Language Models for Linguistic Linked Data Generation*<br>Maria Pia di Buono, Blerina Spahiu and Verginica Barbu Mititelu |
| 11:40–12:00 | *Towards Automated Evaluation of Knowledge Encoded in Large Language Models*<br>Bruno Carlos Luís Ferreira, Catarina Silva and Hugo Gonçalo Oliveira |
| 12:00–12:20 | *Self-Evaluation of Generative AI Prompts for Linguistic Linked Open Data Modelling in Diachronic Analysis*<br>Florentina Armaselu, Chaya Liebeskind and Giedre Valunaite Oleskeviciene |
| **12:20–13:00** | ***Discussions and further work on the topic*** |

# Investigating the Impact of Different Graph Representations for Relation Extraction with Graph Neural Networks

[1]**Moritz Blum, Gennaro Nolano,** [1,2]**Basil Ell,** [1]**Philipp Cimiano**
[1]Bielefeld University, [2]University of Oslo
Germany, Norway
{mblum, bell, cimiano}@techfak.uni-bielefeld.de, nolanogenn@gmail.com

## Abstract

Graph Neural Networks (GNNs) have been applied successfully to various NLP tasks, particularly Relation Extraction (RE). Even though most of these approaches rely on the syntactic dependency tree of a sentence to derive a graph representation, the impact of this choice compared to other possible graph representations has not been evaluated. We examine the effect of representing text though a graph of different graph representations for GNNs that are applied to RE, considering, e. g., a fully connected graph of tokens, of semantic role structures, and combinations thereof. We further examine the impact of background knowledge injection from Knowledge Graphs (KGs) into the graph representation to achieve enhanced graph representations. Our results show that combining multiple graph representations can improve the model's predictions. Moreover, the integration of background knowledge positively impacts scores, as enhancing the text graphs with *Wikidata* features or *WordNet* features can lead to an improvement of close to $0.1$ in $F_1$.

**Keywords:** Relation Extraction, Graph Neural Networks, Background Knowledge

## 1. Introduction

The task of Relation Extraction (RE) consists of predicting the relation between two entities mentioned in a text. It represents an essential subtask for Information Extraction from text, and the result is used in several downstream tasks such as Question Answering (Yu et al., 2017; Xu et al., 2016) or Knowledge Base Population (Nguyen et al., 2018). Recently, approaches based on LSTMs (Hochreiter and Schmidhuber, 1997) and Transformers such as *BERT* (Devlin et al., 2019) have achieved state-of-the-art performance on RE by exploiting contextual information contained in the text around the entities (Wang and Yang, 2020; Baldini Soares et al., 2019; Wu and He, 2019).

A separate line of works makes use of Graph Neural Networks (GNNs), using neural network-based techniques to process graph-structured inputs. GNNs have been applied to RE, typically relying on the syntactic dependency tree of a sentence as graph representation. It has been argued that relying on a syntactic dependency tree i) facilitates dealing with long-distance phenomena (Tian et al., 2021; Miwa and Bansal, 2016), and ii) increases the robustness and generalizability of models (Xu et al., 2015; Marcheggiani and Titov, 2017).

So far, most GNN approaches relied on the syntactic dependency tree of a sentence as a graph, and the impact of different graph representations has not been systematically evaluated. To address this gap, in this work, the impact of different graph representations, as well as combinations thereof, are investigated on three separate datasets.

Most RE approaches do not take into account background knowledge, e. g., from Knowledge Graphs (KGs). GNN-based approaches for RE generally emphasize on the graph representation of sentences (e.g., syntactic trees), and do not use the entity information and the graph context contained in external KGs. However, KGs may provide valuable knowledge about the entities for the RE task (Sun et al., 2020). Moreover, if we train a model such that it can make use of background knowledge, then, under some circumstances, this enables to improve the performance of a model without full retraining. For example, if a fact is missing that a model could use to correctly classify a relation, or if a wrong fact leads to a model incorrectly classifying a relation, than adding or replacing that fact can lead to the model making better predictions.

Therefore, in addition to different graph representations of the sentence, we also investigate enhanced graph representations by injection of KG facts into these graph representations by adding nodes and edges form the KG.

We show that combining multiple graph representations can outperform the models that only use the regular syntactic dependencies. Furthermore, we show that incorporating information from KGs like *Wikidata* (Vrandečić and Krötzsch, 2014) or *WordNet* (Fellbaum, 1998) improves results significantly.

## 2. Related Work

The integration of structured information, such as syntactic dependencies (Tian et al., 2021), semantic dependencies (Chan and Roth, 2011), and back-

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 1–13
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

ground knowledge (Zhang et al., 2021; Peters et al., 2019; Tokuhisa et al., 2022; Wang and Pan, 2020; Sun et al., 2020; Wang and Pan, 2020), is an important topic in NLP.

Recently, much attention has been paid to the incorporation of KG information in language models (Yasunaga et al., 2022; Peters et al., 2019; Tokuhisa et al., 2022). For example, Yasunaga et al. (2022) use a joint language-knowledge foundation model in order to allow the NLP component to incorporate facts from the KG.

While this integration can be implemented as a training task (Yasunaga et al., 2022; Tokuhisa et al., 2022) or by finetuning and adapting pretrained linguistic models (Houlsby et al., 2019; Wang et al., 2020), this usually requires complex architectures and comes with increased computational costs (Hamilton et al., 2022).

Another option is to directly operate on the symbolic graph structure by encoding the information in a graph and then processing it with Graph Neural Networks (GNNs) (Zhang et al., 2018a). GNNs allow to directly learn over graph structure (Dai et al., 2016; Gori et al., 2005; Li et al., 2016; Scarselli et al., 2009; Hamilton et al., 2017) and can be easily combined with standard neural network layers (Defferrard et al., 2016; Gong and Cheng, 2019).

One of the first GNN approaches was proposed by Kipf and Welling (2016), namely a Graph Convolutional Network (GCN), followed by the extension Relational Graph Convolutional Network (R-GCN) (Schlichtkrull et al., 2018), that takes into account edge types. Furthermore, the Relational Graph Attention Network (R-GAT) (Busbridge et al., 2019) adds an attention mechanism to the R-GCN model. GNNs have been applied to a variety of tasks, such as Link Prediction (Schlichtkrull et al., 2018), Neural Machine Translation (Bastings et al., 2017; Marcheggiani et al., 2018), and Semantic Role Labeling (Marcheggiani and Titov, 2017).

Zhang et al. (2018a) have been one of the first to apply GNNs to RE. Their model applies a GNN encoder over syntactic dependency paths with unlabeled edges, and achieves comparable results to approaches based on bidirectional LSTMs and LLMs. Guo et al. (2019) and Tian et al. (2021) extended the use of GNNs for RE by applying a GNN with an attention mechanism and the capacity to encode labeled edges. Nadgeri et al. (2021), instead, explores the integration of external textual information (e. g., from *Wikidata*) into a GNN model for RE.

Recently, Yu et al. (2022) have shown linguistic knowledge fusion for downstream tasks by comparing different kinds of graph structures for several tasks in the GLUE benchmark. They investigate syntactic dependencies, semantic dependencies, binary balance trees, and linear chains of tokens. The work by Yu et al. (2022) does not investigate the impact of the representations on RE approaches and previous work on RE still mainly focuses on syntactic dependency trees. Therefore, the literature lacks a thorough evaluation of different graph structures and their combinations for RE with GNNs.

We present a deep investigation of several graph representations for the RE task and analyze them individually and in combinations. We build upon the research conducted by Yu et al. (2022) as we investigate different graph representations for RE. Furthermore, we go beyond by examining enhanced graph representations that incorporate KG facts.

# 3. Models and Graph Representations

In our experiments, we utilize a GNN architecture comprising two stacked GNN layers with a linear layer for relation classification. The architecture is shown in Figure 1.

The GNN layers encode the graph representation of the input sentence containing the two entities to be classified. We use Glove token embeddings, or a pre-trained but non-trainable BERT to derive token embeddings, and RDF2Vec for the KG entities. These embeddings serve as node features for the given graph.

To focus this investigation on the different graph representations, we decided to freeze the encoding model and do not investigate trainable encoders, like an end-to-end trainable BERT encoder, to derive token embeddings. GNN-based RE models that use an end-to-end trainable encoder are able to achieve state-of-the-art performance (Zhang et al., 2018a; Guo et al., 2019; Tian et al., 2021).

After the two GNN layers, the resulting representations of the subject and object entities are used as input to the linear classification layer. In the case of multi-word entities, we rely on the representation of the token with the largest number of outgoing syntactic dependencies.

## 3.1. Graph Representations

In order to apply this GNN model for RE, we represent tokens as nodes and connect them through (typed) edges to obtain a graph. The investigated graph structures are:

**1)** Tokens connected in a linear chain (*chain*), in the same order as they occur in the text.

**2)** Every token connected to every other token, what leads to a fully connected graph (*fully*) and allows every token to access the features of every other token.

**3)** Tokens connected according to syntactic dependencies (*syn*).

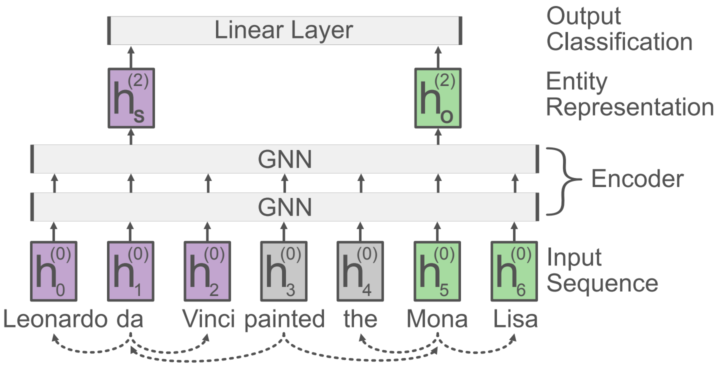


Figure 1: GNN model architecture.The model operates over a given graph with given input node features $h_i^0$ that are derived by embedding the token or KG entity by a suitable embedding model. $h_i^{(l)}$ denotes the features of token $i$ at layer $l$ of the GNN. In this context, $h_s^{(2)}$ and $h_o^{(2)}$ denote the feature representation of subject and object entity after two GNN layers. $h_s^{(2)}$ and $h_o^{(2)}$ are then feed into the linear layer for relation classification.

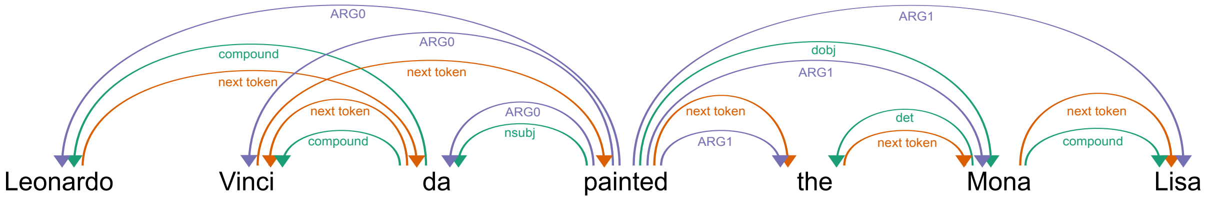


Figure 2: An example of a graph that combines the three graph representations chain, syn, and sem (colored *orange*, *green*, *purple*).

**4)** Tokens connected through higher order syntactic dependency relations (*highsyn*) according to Tian et al. (2022). Here, tokens are related if there are at most two tokens in between when traversing the syntactic dependency tree, directly connecting tokens that are syntactically close. We refer to Tian et al. (2022) and App. A for more details.
**5)** Tokens are connected according to their semantic dependencies (*sem*) in the form of the latent PropBank-based (Palmer et al., 2005) predicate argument structure derived by means of Semantic Role Labelling (Shi and Lin, 2019).

We evaluate all possible combinations of these methods. An example graph is shown in Figure 2.

### 3.2. Graph Representations with Additional Background Knowledge

We investigate the integration of background knowledge from a KG into the graphs. To do so, new nodes are created, representing the entities involved in the relations. The node features are derived from the background KG (*nodes*). These nodes are then connected to the corresponding entity mentions. We only connect subject and object entity mentions to their corresponding KG entities. Additionally, we consider adding the shortest paths (*s.p.*) to the graph. We use the shortest path between subject and object entity in the KG, and include any external entity present on these paths as additional nodes, as well as any edge connecting them. We refrain from explicitly adding an edge between a node on the shortest path and any node representing an entity mentioned in the text. An example is shown in Figure 3.

## 4. Experiments

We investigate the impact of using different graph structures as graph representations on the task of RE by training and evaluating multiple GNN models.[1]

In order to derive syntactic dependencies, we rely on Spacy,[2] while for semantic dependencies we make use of the AllenNLP library[3] described in (Gardner et al., 2018). As node features, we use $100$ dimensional *Glove* embeddings[4] (Pennington et al., 2014), or $768$ dimensional contextual *BERT* embeddings (Devlin et al., 2019).

We automatically determine the best GNN hyperparameter settings using the hyperparameter

---

[1] We use PyTorch Geometric to implement our GNNs, `github.com/pyg-team/pytorch_geometric`.

[2] See `spacy.io/`.

[3] See `github.com/allenai/allennlp`.

[4] We also experimented with the $300$ dimensional embeddings, and found the results to be interchangeable. For runtime optimization reasons, we opted for the lower dimensional embeddings in the final experiments.

search framework ASHA (Li et al., 2020), which applies intelligent early-stopping and supports large-scale parallelization. The main hyperparameter of our model is the type of the GNN layers (i.e., GCN, R-GCN, R-GAT) as described in Section 2. Furthermore, these models have hyperparameters like the dimensionality of the GNN layers and linear layers ($64$, $120$, $240$), the learning rate ($8$ samples from $10^{-3}$ to $10^{-5}$), and the batch size ($32$, $64$, $128$). In addition, we evaluate the impact of i) adding reverse edges, ii) adding self-loops to each node such that its previous feature vector can be accessed by itself, and iii) exploiting the labels of edges.

We used the $F_1$ score as the criterion to select the best model. Since unpromising runs are terminated at an early stage, not all model configurations are trained until convergence and evaluation results are not produced for all the considered model configurations.

We evaluate the graph representations on two English RE datasets that are linked to *Wikidata* and on the commonly used RE benchmark *SemEval 2010 Task 8* dataset to validate our models.

The required property of the evaluation datasets was that all subjects and objects of a relation are annotated with their corresponding *Wikidata* ID, such that background information can be used. However, there is a lack of RE datasets that are annotated with *Wikidata* entities as most datasets are annotated with *Freebase* entities and relations (Mintz et al., 2009). Therefore, we created our own datasets based on *FewRel* and *T-REx*.

Moreover, to validate that our models are solving the RE task sufficiently, we run the standard evaluation without background knowledge on the *SemEval 2010 Task 8* dataset.

In detail, we consider the following datasets:

**1) FewRel (custom)**: *FewRel* (Han et al., 2018; Gao et al., 2019) is a large RE dataset with entity mentions and relations annotated with their corresponding *Wikidata* IDs. It was created through a combination of distant supervision and human annotation. Originally developed for few-shot RE, we repurpose it for standard RE by merging its `train` and `val` splits. These splits encompass sentences expressing $64$ and $16$ distinct relations, each with $700$ examples, totaling $56,000$ sentences. The combined dataset is then randomly split into `train/dev/test` splits with percentages $70/15/15$. In *FewRel*, all subjects and objects of a relation are annotated with their corresponding *Wikidata* ID, and, therefore, there cannot be a subject or object which has no Wikidata ID in our *FewRel (custom)* dataset, too.

**2) T-REx (custom)**: We randomly sampled $1000$ sentences for each relation occurring at least $1,000$ times from the *T-REx* dataset (Elsahar et al., 2018), which was created by an automatic alignment of *Wikipedia* abstracts and *Wikidata* triples. We only selected sentences in which both, subject and object are annotated with *Wikidata* IDs. Therefore, all subjects and objects of a relation are annotated with their corresponding *Wikidata* ID. This dataset contains $228,000$ sentences expressing $228$ different relations.

**3) SemEval 2010 Task 8**: This dataset consists of $8,000$ human-annotated training and $2,717$ human-annotated test sentences with a relation between two given nominals. We use $20\%$ of the train set for validation (Hendrickx et al., 2010). However, the publicly available test set was not modified to ensure comparability to other work on RE. Since this dataset is not annotated with any KG IDs, we use it only to evaluate the different types of graph representations for RE, and not the knowledge injection.

As background knowledge, we rely on two KGs, namely *Wikidata*[5] (Vrandečić and Krötzsch, 2014) and *WordNet* (Fellbaum, 1998). *Wikidata* is build by many editors and partially automatic. It encompasses data about entities such as people, places, organizations, or abstract topics, along with details about their interconnections and relationships. *WordNet* is a manually created lexical database that categorizes nouns, verbs, adjectives, and adverbs into synsets. These synsets are connected through conceptual-semantic and lexical relations, forming a KG that captures the interconnections between different linguistic elements.

The features for the added KG nodes are derived via *RDF2Vec* (Ristoski and Paulheim, 2016).[6] *RDF2Vec* is a method that derives embeddings for the entities and relations in a KG. In case that the KG contains facts that are relations-to-be-predicted, they are removed from the dataset, so they do not affect the embeddings. We remove triples contained in the RE datasets from our *Wikidata* graph before we derive the embeddings. No triples needed to be removed to derive the *WordNet* features, as no relation in *WordNet* can inadvertently reveal the relations that will be predicted for any of our datasets. The derived features share the same dimensionality of the other nodes' embeddings and are used as vector of newly created nodes which are connected to their associated entity mention's tokens.

For *Wikidata*, the integration of shortest paths between entities in KGs can be valuable for RE. Therefore, nodes are created for every entity on the path between the mentions, and connected among themselves, as shown in Figure 3. In the majority of cases, the shortest paths consist of only one

---

[5] We use the *Wikidata* dump from October 2022.

[6] *RDF2Vec* embeddings are trained using the *jRDF2Vec* implementation described by Portisch et al. (2020), `https://github.com/dwslab/jRDF2Vec`.

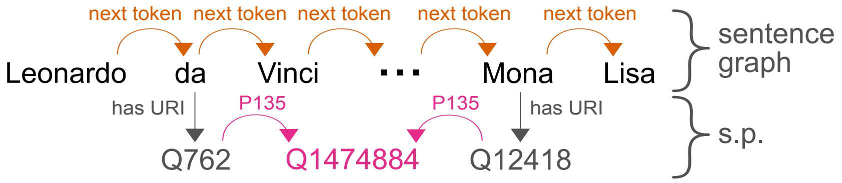


Figure 3: Integration of the shortest path (shown in pink) between *Leonardo da Vinci* and *Mona Lisa* in *Wikidata* into the chain graph. The *Wikidata* entity IDs *Q762*, *Q12418*, and *Q1474884* represent *Leonardo da Vinci*, *High Renaissance*, and *Mona Lisa*, whereas the *Wikidata* property ID *P135* expresses the *movement* relation.

entity positioned between the subject and object of the relation intended for classification.

We evaluated two simple models, both using a feedforward neural network with two layers and a classification layer (denoted as *Linear-NN*), to compare our GNN models that encode graph structure against those operating on text-based embeddings. One model takes as input the concatenation of the word embeddings of the two entity mentions, while the second one uses *RDF2Vec* features for subject and object.

Our code is available on GitHub.[7]

## 5. Results

The best performing model across all experiments is based on a two layer R-GCN encoder with self-loops and reverse edges, and a hidden dimension of $120$ (*Glove*), respectively $240$ (*BERT*) for the GNN layers and linear layers. The model is trained with a batch size of $64$ and a learning rate of $0.0001$.

**Graph Representations** Our results, shown in Table 1, Table 3, and Table 4, show that the models using *BERT* features perform better compared to those using *Glove* features for all graph representations. The input graph representations do not lead to consistent performance across all datasets.

The evaluation results of the graph structures on *FewRel (custom)*, displayed in Table 1, shows that the best performance is reached by using syntactic dependencies with an $F_1$ score of $0.754$, followed by higher order syntactic dependencies ($F_1$ of $0.745$), and by the linear chain ($F_1$ of $0.703$). Regarding the models that operate on a combination of multiple graph representations, the combination of syntactic dependencies and the fully connected token graph leads the best results and achieves an $F_1$ score of $0.764$.

On the *T-REx (custom)* dataset, the evaluation scores are shown in Table 3, the best performance is achieved by the model operating over syntactic dependencies with an $F_1$ score of $0.697$, followed by the fully connected graph ($F_1$ of $0.693$) and by the linear chain ($F_1$ of $0.689$). For the combined graph representations, the combination of the linear chain and higher order syntactic dependencies shows the best results with an $F_1$ score of $0.761$.

The scores for the *SemEval* dataset, shown in Table 4, show that the best performance is reached by a model operating over the graph of syntactic dependencies with an $F_1$ score of $0.786$. The second-best model uses the higher order dependencies ($F_1$ of $0.766$), and the third-best model uses the semantic dependencies ($F_1$ of $0.745$). For the combined representations, the combination of syntactic dependencies and semantic dependencies shows the best results with an $F_1$ score of $0.786$.

**Graph Representations with Additional Background Knowledge** The impact of adding KG features to the graph consisting of the fully connected graph and the syntactic dependency graph (denoted *fully+syn*) on the performance of the models is shown in Table 2 for the *FewRel (custom)* dataset and the *T-REx (custom)* dataset. The *SemEval* dataset was not evaluated with additional KG features, as the entities in this dataset are not linked to a KG.

The additional *Wikidata* or *WordNet* features lead to an improvement of the scores in all cases.

All GCN models that use *Wikidata* or *WordNet* features outperform the NN baselines that use *Wikidata RDF2Vec* features only or *word embedding* features only.

On the *FewRel (custom)* dataset, the *BERT* model that uses the combined graph of syntactic dependencies and the fully connected graph can be improved from an $F_1$ score of $0.764$ to an $F_1$ of $0.82$ (additional Wikidata nodes), $0.859$ (additional Wikidata shortest path), respectively $0.763$ (additional WordNet nodes) by using additional background knowledge.

By adding additional background knowledge to the combined graph of syntactic dependencies and the fully connected graph, the $F_1$ scores of the *BERT* model on the *T-REx (custom)* dataset improve from $0.714$ to $0.746$ (additional Wikidata nodes), $0.791$ (additional Wikidata shortest path), respectively $0.729$ (additional WordNet nodes).

---

[7] See `github.com/Nolanogenn/re_with_gcn`.

Table 1: General evaluation of the different graph representation and their combinations on the *FewRel (custom)* dataset.

| **Graph Representation** | **Glove** | | | **BERT** | | |
|---|---|---|---|---|---|---|
| | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ |
| chain | 0.44 | 0.445 | 0.468 | 0.703 | 0.704 | 0.71 |
| fully | 0.484 | 0.476 | 0.515 | 0.699 | 0.705 | 0.71 |
| syn | 0.566 | 0.566 | 0.579 | 0.754 | 0.757 | 0.757 |
| sem | 0.388 | 0.407 | 0.418 | 0.667 | 0.666 | 0.676 |
| highsyn | 0.594 | 0.539 | 0.612 | 0.745 | 0.746 | 0.749 |
| chain + syn | 0.571 | 0.57 | 0.589 | 0.75 | 0.753 | 0.754 |
| chai + sem | 0.516 | 0.525 | 0.533 | 0.723 | 0.726 | 0.728 |
| fully + syn | **0.611** | **0.612** | **0.627** | **0.764** | **0.766** | **0.767** |
| fully + sem | 0.489 | 0.483 | 0.516 | 0.708 | 0.711 | 0.715 |
| syn + sem | 0.574 | 0.569 | 0.591 | 0.753 | 0.754 | 0.756 |
| chain + highsyn | 0.574 | 0.565 | 0.6 | 0.745 | 0.747 | 0.749 |
| fully + highsyn | 0.603 | 0.6 | 0.622 | 0.75 | 0.753 | 0.754 |
| highsyn + sem | 0.577 | 0.582 | 0.608 | 0.743 | 0.744 | 0.748 |
| chain + syn + sem | 0.583 | 0.587 | 0.6 | 0.751 | 0.754 | 0.754 |
| fully + syn + sem | 0.608 | 0.612 | 0.623 | 0.753 | 0.755 | 0.757 |
| chain + highsyn + sem | 0.578 | 0.572 | 0.603 | 0.745 | 0.747 | 0.748 |
| fully + highsyn + sem | 0.586 | 0.576 | 0.611 | 0.746 | 0.749 | 0.748 |

Table 2: Evaluation of graph representations enhanced with additional KG features from *Wikidata* and *WordNet* on *FewRel (custom)* and *T-REx (custom)*. The *Glove* models are provided with $100$ dimensional embeddings, whereas the *BERT* models are provided with $768$ dimensional embeddings.

| **Model & Graph Representation** | **Glove** | | | **BERT** | | |
|---|---|---|---|---|---|---|
| | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ |
| | FewRel (custom) | | | | | |
| Linear-NN: word embeddings | 0.277 | 0.323 | 0.294 | 0.382 | 0.436 | 0.3 |
| Linear-NN: RDF2Vec embeddings | 0.597 | 0.618 | 0.606 | 0.664 | 0.675 | 0.669 |
| GCN: syn | 0.566 | 0.566 | 0.579 | 0.754 | 0.757 | 0.757 |
| GCN: fully + syn | 0.611 | 0.612 | 0.627 | 0.764 | 0.766 | 0.767 |
| + *Wikidata* nodes | 0.784 | 0.778 | 0.803 | 0.82 | 0.823 | 0.823 |
| + *Wikidata* shortest path | **0.835** | **0.834** | **0.845** | **0.859** | **0.86** | **0.861** |
| + *WordNet* nodes | 0.684 | 0.685 | 0.697 | 0.763 | 0.765 | 0.766 |
| | T-REx (custom) | | | | | |
| Linear-NN: BERT | 0.082 | 0.15 | 0.103 | 0.239 | 0.304 | 0.251 |
| Linear-NN: RDF2Vec | 0.438 | 0.488 | 0.475 | 0.506 | 0.548 | 0.525 |
| GCN: syn | 0.406 | 0.399 | 0.451 | 0.697 | 0.698 | 0.72 |
| GCN: fully + syn | 0.45 | 0.436 | 0.498 | 0.714 | 0.708 | 0.735 |
| + *Wikidata* nodes | 0.661 | 0.64 | 0.712 | 0.746 | 0.35 | 0.776 |
| + *Wikidata* shortest path | **0.685** | **0.664** | **0.734** | **0.791** | **0.782** | **0.814** |
| + *WordNet* nodes | 0.561 | 0.572 | 0.592 | 0.729 | 0.73 | 0.746 |

Overall, the best result can be achieved when adding the shortest path in *Wikidata* between subject and object to the graph representation.

## 6. Discussion

**Graph Representations** The input graph representations do not lead to consistent performance across all datasets. This might be caused by the sentence structure or sentence complexity in the datasets.

We observe that the models that use *Glove* features show good scores for the combined representation of *fully* + *syn* + *sem* across all datasets, and always perform slightly better than the individual representations.

For the *BERT* models, the different representations do lead to different performances across the datasets, and we can not observe a general trend. However, on the *FewRel (custom)* dataset and the *T-REx (custom)* dataset, combining syntactic dependencies with other graph representations leads

Table 3: General evaluation of the different graph representation and their combinations on the *T-REx (custom)* dataset.

| Graph Representation | Glove | | | BERT | | |
|---|---|---|---|---|---|---|
| | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ |
| chain | 0.351 | 0.362 | 0.386 | 0.689 | 0.695 | 0.709 |
| fully | 0.366 | 0.359 | 0.417 | 0.693 | 0.696 | 0.712 |
| syn | 0.406 | 0.399 | 0.451 | 0.697 | 0.698 | 0.72 |
| sem | 0.33 | 0.338 | 0.362 | 0.663 | 0.665 | 0.688 |
| highsyn | 0.433 | 0.419 | 0.483 | 0.674 | 0.673 | 0.703 |
| chain + syn | 0.42 | 0.408 | 0.465 | 0.714 | 0.718 | 0.731 |
| chain + sem | 0.389 | 0.38 | 0.431 | 0.688 | 0.686 | 0.711 |
| fully + syn | 0.45 | 0.436 | **0.498** | 0.714 | 0.708 | 0.735 |
| fully + sem | 0.39 | 0.386 | 0.433 | 0.687 | 0.686 | 0.712 |
| syn + sem | 0.426 | 0.41 | 0.472 | 0.699 | 0.696 | 0.721 |
| chain + highsyn | 0.431 | 0.414 | 0.483 | **0.761** | **0.763** | **0.762** |
| fully + highsyn | 0.431 | 0.411 | 0.488 | 0.666 | 0.66 | 0.699 |
| highsyn + sem | 0.424 | 0.41 | 0.477 | 0.645 | 0.645 | 0.693 |
| chain + syn + sem | 0.436 | 0.417 | 0.483 | 0.697 | 0.689 | 0.721 |
| fully + syn + sem | **0.453** | **0.443** | 0.497 | 0.658 | 0.646 | 0.689 |
| chain + highsyn + sem | 0.43 | 0.409 | 0.484 | 0.653 | 0.639 | 0.687 |
| fully + highsyn + sem | 0.427 | 0.407 | 0.482 | 0.646 | 0.63 | 0.681 |

Table 4: General evaluation of the different graph representation and their combinations on the *SemEval 2010 Task 7* dataset.

| Graph Representation | Glove | | | BERT | | |
|---|---|---|---|---|---|---|
| | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ |
| chain | 0.686 | 0.689 | 0.688 | 0.715 | 0.716 | 0.718 |
| fully | 0.656 | 0.669 | 0.658 | 0.69 | 0.691 | 0.692 |
| syn | 0.756 | 0.76 | 0.757 | **0.786** | 0.783 | **0.791** |
| sem | 0.705 | 0.72 | 0.702 | 0.745 | 0.749 | 0.745 |
| highsyn | 0.752 | 0.758 | 0.75 | 0.766 | 0.769 | 0.766 |
| chain + syn | 0.752 | 0.757 | 0.749 | 0.776 | 0.775 | 0.779 |
| chain + sem | 0.746 | 0.752 | 0.743 | 0.777 | 0.777 | 0.78 |
| fully + syn | 0.76 | 0.766 | 0.756 | 0.776 | 0.775 | 0.781 |
| fully + sem | 0.716 | 0.722 | 0.718 | 0.748 | 0.75 | 0.749 |
| syn + sem | 0.764 | 0.768 | 0.762 | **0.786** | **0.788** | 0.787 |
| chain + highsyn | 0.745 | 0.747 | 0.746 | 0.761 | 0.763 | 0.762 |
| fully + highsyn | 0.742 | 0.747 | 0.741 | 0.771 | 0.772 | 0.773 |
| highsyn + sem | 0.751 | 0.754 | 0.749 | 0.771 | 0.77 | 0.776 |
| chain + syn + sem | 0.759 | 0.765 | 0.755 | 0.781 | 0.779 | 0.784 |
| fully + syn + sem | **0.768** | **0.773** | **0.766** | 0.785 | 0.785 | 0.789 |
| chain + highsyn + sem | 0.741 | 0.752 | 0.747 | 0.768 | 0.766 | 0.775 |
| fully + highsyn + sem | 0.754 | 0.757 | 0.753 | 0.764 | 0.768 | 0.763 |

to improved scores compared to using only syntactic dependencies, whereas this worsens the scores on the *SemEval* dataset.

For the *Glove* models, the combination of multiple graph representations generally leads to better scores than using the individual representations. This trend can not be observed for the *BERT*, and we can assume that this information is already encoded in the *BERT* embeddings. Therefore, the graph representations do not provide additional context information, but rather confuse the model by adding redundant information.

All in all, combinations of graph representations can add additional information that can be used by GNNs for RE. But it must be noted, that the benefit is low and differs depending on the dataset. However, we were able to show that even simple graph representations without linguistic knowledge, like a linear chain of tokens or the fully connected graph of tokens, still lead to adequate models.

Our GNN model is limited to two layers, which leads to a receptive field of two graph hops. Therefore, models operating over representations that connect distant entities should clearly outperform those models that can only access a certain number of tokens in the graph neighborhood. However, this assumption is not always the case. Therefore, we assume carefully selecting a suitable graph representation instead of simply providing all available tokens might be valuable.

**Graph Representations with Additional Background Knowledge** We evaluate different representations acquired through parsing the sentence structure, and enriched by background knowledge.

Adding *Wikidata* information could make a direct comparison seem unfair. The additional KG information could be helpful for the model as they provide additional information about the subject and object entity not expressed in the sentence. But the additional information could also be a drawback to the models. Background knowledge might contain irrelevant information for the task, potentially introducing noise and complicating the model's focus on relevant features.

In contrast, the integration of *WordNet* features does not add any unfair advantages to the model, as this is commonly done in NLP. Providing additional external resources like *WordNet* information, part-of-speech tags, dependency information, and named entity tags is often done for RE (Shen and Huang, 2016; Zhang et al., 2015).

Nevertheless, to prevent confusion, we present the results of models with additional background knowledge separately in dedicated tables.

In general, the incorporation of information from *Wikidata* as additional nodes connected to the subject and object nodes, or as the shortest path between both, has a positive impact. For instance, on *FewRel (custom)*, the best model that uses the *Wikidata* shortest paths achieves an $F_1$ score of $0.859$. This is an increase of $0.095$ in $F_1$ compared to the base *fully+syn* GNN model without KG features.

*WordNet* features do increase the performance of all models, too. *WordNet* provides additional information about synonyms and related concepts, as well as various semantic relationships between words to the RE model. According to our results, the *WordNet* information is helpful for GNN-based RE.

However, adding the richer and more diverse *Wikidata* features to the graph increases the scores more than adding *WordNet* features. This might be because Wikidata provides more background knowledge, i.e., the relations entities are involved in the KG, which might be more valuable than the *WordNet* information.

All GCN models that use some sentence graph representation and additional KG features outperform the *Wikidata RDF2Vec*-based and *word embedding based* NN model. Especially, adding the *Wikidata shortest path* leads to best scores. This shows a successful fusion of text and KG information in a common graph representation, as the model that applies fusion outperforms the individual models.

## 7. Conclusion

Our results show that combining multiple graph representations can improve the model's predictions. Although our experiments revealed that none of the graph representations consistently performs best across multiple datasets, we can clearly see i) that most representations improve the performance compared to the standard graph representation, and ii) that the representations have a strong impact on performance, which makes the type of graph representation an important hyperparameter that is worth to be tuned.

Furthermore, the integration of background knowledge from *Wikidata* or *WordNet* positively impacts scores and can lead to an improvement of close to $0.1$ in $F_1$.

In future work, we will investigate methods to integrate structured background knowledge beyond additional subject and object nodes and shortest paths between them. Furthermore, we will investigate how the model performance can be improved by removing wrong facts and adding missing facts to the KG.

## 8. Limitations

The present work has some minor limitations that should be acknowledged.

Firstly, our models do not reach state-of-the art performance. However, beating state-of-the-art performance was not the goal of this work. Instead, we investigate of different graph representations. As the difference to state-of-the-art is small, one can assume our GNN model to be set up correctly.

Secondly, even though our GNN models have significantly fewer parameters than *BERT* (*9M* vs. *110M*), our best models rely on token features derived from *BERT*. However, our training is faster than training *BERT* from scratch.

Thirdly, it is important to note that incorporating facts from a KG could make the model biased to the information stored in the form of triples in the KG instead of the information expressed in the sentence context. Future research could use explainability methods or attention mechanisms to determine which information the model prioritizes.

## Ethics

Any potential biases present in the relation extraction datasets or knowledge graphs used in our approach can impact the fairness and accuracy of the extracted relations. However, it is important to note that our work primarily focuses on the evaluation of different graph representations, which do not introduce new ethical biases themselves. Nevertheless, careful consideration should still be given to the potential biases inherited from the datasets and knowledge graph sources to ensure the ethical and unbiased nature of our approach.

## Acknowledgements

This work was carried out in the context of DiProMag, a BMBF (German Federal Ministry of Education and Research) funded research project under Grant No. 13XP5120B. This work was partially supported by: the Research Council of Norway through its Centres of Excellence scheme, Integreat – Norwegian Centre for knowledge-driven machine learning, project number 332645; the Ministry of Culture and Science of the State of North Rhine-Westphalia (Germany) through SAIL, Grant No. NW21-059A; the Deutsche Forschungsgemeinschaft (DFG) through the priority program RATIO (SPP-1999), Grant No. 376059226.

## 9. Bibliographical References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Razvan Bunescu and Raymond Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*.

Yee Seng Chan and Dan Roth. 2011. Exploiting Syntactico-Semantic Structures for Relation Extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2702–2711. JMLR.org.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA. Curran Associates Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards More Challenging Few-Shot Relation Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Liyu Gong and Qiang Cheng. 2019. Exploiting edge features for graph neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9203–9211.

M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. 2022. Is neuro-symbolic AI meeting its promises in natural language processing? A structured review. *Semantic Web*, pages 1–42.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. 2020. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018. Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th*

*Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Abhishek Nadgeri, Anson Bastos, Kuldeep Singh, Isaiah Onando Mulang', Johannes Hoffart, Saeedeh Shekarpour, and Vijay Saraswat. 2021. KGPool: Dynamic Knowledge Graph Context Selection for Relation Extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 535–548, Online. Association for Computational Linguistics.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333, New Orleans, Louisiana. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Jan Portisch, Michael Hladik, and Heiko Paulheim. 2020. RDF2Vec Light–A Lightweight Approach for Knowledge Graph Embeddings. *International Semantic Web Conference, Posters and Demos*.

Petar Ristoski and Heiko Paulheim. 2016. RDF2Vec: RDF Graph Embeddings for Data Mining. In *The Semantic Web – ISWC 2016*, pages 498–514, Cham. Springer International Publishing.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536, Osaka, Japan. The COLING 2016 Organizing Committee.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving Relation Extraction through Syntax-induced Pre-training with Dependency Masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886, Dublin, Ireland. Association for Computational Linguistics.

Ryoko Tokuhisa, Keisuke Kawano, Akihiro Nakamura, and Satoshi Koide. 2022. Enhancing Contextual Word Representations Using Embedding of Neighboring Entities in Knowledge Graphs. In

*Proceedings of the 29th International Conference on Computational Linguistics*, pages 3175–3186, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10):78–85.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. Cite arxiv:2002.01808.

Wenya Wang and Sinno Jialin Pan. 2020. Integrating deep learning with logic fusion for information extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9225–9232.

Zihan Wang and Bo Yang. 2020. Attention-based Bidirectional Long Short-Term Memory Networks for Relation Classification Using Knowledge Distillation from BERT. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 562–568.

Shanchan Wu and Yifan He. 2019. Enriching Pre-Trained Language Model with Entity Information for Relation Classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 536–540, Lisbon, Portugal. Association for Computational Linguistics.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question Answering on Freebase via Relation Extraction and Textual Evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336, Berlin, Germany. Association for Computational Linguistics.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D Manning, Percy S Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. *Advances in Neural Information Processing Systems*, 35:37309–37323.

Changlong Yu, Tianyi Xiao, Lingpeng Kong, Yangqiu Song, and Wilfred Ng. 2022. An Empirical Revisiting of Linguistic Knowledge Fusion in Language Understanding Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

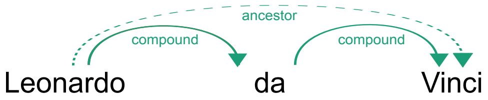
Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved Neural Relation Detection for Knowledge Base Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581, Vancouver, Canada. Association for Computational Linguistics.

Congcong Zhang, Gaofei Xie, Ning Liu, Xiaojie Hu, Yatian Shen, and Xiajiong Shen. 2021. Automatic Hypernym-Hyponym Relation Extraction With WordNet Projection. In *2021 7th International Conference on Systems and Informatics (ICSAI)*, pages 1–6.

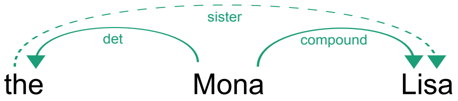
Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018a. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*,



(a) Ancestor



(b) Sister

Figure 4: Higher order syntactic dependencies. Solid lines represent the syntactic first order dependencies, whereas the dashed lines represent the second order dependencies.

pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

## A. Higher Order Syntactic Dependencies

We implement higher order syntactic dependencies as proposed by Tian et al. (2022). The syntactic dependencies serve as first order dependencies. Based on those, second and third order dependencies are added.

For example, second order dependencies establish directed connections between two tokens, $token_i$ and $token_j$, if there exists a single token, $token_x$, along the non-directional shortest path connecting $token_i$ and $token_j$. In detail, we define two distinct relation types based on the direction of the edges in the graph. If the connection between the tokens is $token_i \rightarrow token_x \rightarrow token_j$, we establish the *ancestor* relation pointing from $token_i$ to $token_j$. If the relations are $token_i \leftarrow token_x \rightarrow token_j$, we add the *sister* relation between $token_i$ and $token_j$. Examples of the two relations are shown in Figure 4.

The third order dependencies are defined similarly for the case of two tokens in between $token_i$ and $token_j$, along the non-directional shortest between them.

We do not add inverse relations, as this is a hyperparameter of the graph preprocessing.

# TaxoCritic: Exploring Credit Assignment in Taxonomy Induction with Multi-Critic Reinforcement Learning

**Injy Sarhan[1,2], Bendegúz Toth[2], Pablo Mosteiro[2], Shihan Wang[2]**
[1]Shell Global Solutions International B.V., Amsterdam, The Netherlands
[2]Utrecht University, Utrecht, The Netherlands
injy.sarhan@shell.com, toth.bendeguz@outlook.com, {p.mosteiro, s.wang2}@uu.nl

## Abstract

Taxonomies can serve as a vital foundation for several downstream tasks such as information retrieval and question answering, yet manual construction limits coverage and full potential. Automatic taxonomy induction, particularly using deep Reinforcement Learning (RL), is underexplored in Natural Language Processing (NLP). To address this gap, we present TaxoCritic, a novel approach that leverages deep multi-critic RL agents for taxonomy induction while incorporating credit assignment mechanisms. Our system uniquely assesses different sub-actions within the induction process, providing a granular analysis that aids in the precise attribution of credit and blame. We evaluate the effectiveness of multi-critic algorithms in experiments regarding both accuracy and robustness performance in edge identification. By providing a detailed comparison with state-of-the-art models and highlighting the strengths and limitations of our method, we aim to contribute to the ongoing development of automatic taxonomy induction while exploring the usage of deep RL techniques in this field.

**Keywords:** Taxonomy Induction, Reinforcement Learning, Credit Assignment, Actor-Critic

## 1. Introduction

A domain's taxonomy categorizes concepts based on "is-a" relationships (Brachman, 1983), forming acyclic graphs. Nodes represent terms, and directed edges signify relationships. In this context, *"P is-a Q"* implies that term *P* (hyponym) is a subclass or more specific instance of term *Q* (hypernym). Taxonomies provide a hierarchical organization of concepts that enables more efficient data categorization and retrieval. Recent advances aim to automatically create faceted taxonomies to support more nuanced classifications and facilitate easier search and navigation within linked data applications (Zong et al., 2017). Additionally, several NLP methods utilize term taxonomies to support knowledge-rich applications, such as information extraction (Demeester et al., 2016) and question answering (Harabagiu et al., 2003), demonstrating the importance of structured knowledge that is embodied in taxonomies. Integrating taxonomies into linked datasets can significantly enhance interoperability and semantic depth, contributing to improved understanding, reasoning, and performance on complex NLP tasks.

Manual taxonomy construction is a resource-intensive and time-consuming task that requires domain knowledge. There have been efforts to handcraft large taxonomies, such as WordNet (Miller, 1995), yet ensuring comprehensive coverage remains a challenge. Automatically constructing a high-quality taxonomy is non-trivial. The goal is to infer a taxonomy graph from a set of background resources. This involves two subtasks (Wang et al., 2017): **(a) Hierarchy detection:** Identifying *"is-a"* relations between terms. Various combinations of candidate words are tested with the aid of a background corpus to uncover domain-specific relations. **(b) Hierarchy construction:** Organizing extracted pairs from (a) in a tree-like structure presents challenges, including representing transitive relations[1], and ensuring the taxonomy remains an acyclic graph with a single root node, to which all other nodes can trace a path.

The asymmetrical nature of the hypernym relation leads to two possibilities: (1) The parent node (hypernym) exists, enabling the addition of its pair as a child node (hyponym); (2) The child node is already in the taxonomy, requiring the addition of its parent, which is more complex due to the taxonomy's graph structure. Since the taxonomy is a tree with a single *root*, all nodes inherently have a parent, making it non-trivial to add a new parent node for an arbitrary child node. Consequently, most methods allow the insertion of a child node into an existing parent, and not the reverse.

Unlike conventional approaches, deep Reinforcement Learning (RL) allows for simultaneous optimization of both hierarchy detection and organization tasks, minimizing error propagation (Mao et al., 2018). Despite its potential, deep RL's application in taxonomy induction is limited (Mao et al., 2018; Han et al., 2021). In taxonomy induction, an RL action involves selecting a term (child node) from the remaining set and adding it to another term (parent node) in the taxonomy. Previous work unified these

---

[1] A transitive relation is defined as: if $a$ "is-a" $b$ and $b$ "is-a" $c$ then also $a$ "is-a" $c$. Entity ambiguity complicates these relations in automated taxonomies (Liang et al., 2017).

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 14–30
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

actions (Mao et al., 2018; Han et al., 2021). We posit that both components (chosen child and parent nodes) must be correct for meaningful learning of a single action. Actions, rather than nodes, are deemed correct or incorrect as a whole. However, in certain cases, one of the sub-actions[2] might be contextually accurate for the taxonomy being constructed. This leads us to the problem of credit assignment which involves identifying the cause of a certain outcome (Minsky, 1961). Proper credit assignment is crucial for pinpointing the component in the action that originates the error. Without it, the model's learning process and performance can be hindered. In this paper, we delve into the crucial aspect of credit assignment, and explore how credit assignment along with multi-critic can better attribute blame to specific sub-actions.

We introduce TaxoCritic, a novel deep RL method for automatic taxonomy induction[3]. Our goal is to enhance this task using multi-critic RL, emphasizing improved credit assignment. Our contributions are: 1) Introduce a novel RL formalization that considers parent and child nodes of the action in taxonomy induction simultaneously, in contrast to prior methods. 2) Conduct a thorough experimental evaluation of credit assignment in taxonomy induction. 3) Propose a multi-critic approach to highlight the effectiveness of credit assignment in taxonomy induction, leading to improved robustness.

The paper is structured as follows. In Section 2 we present an overview of previous work upon which we build. Section 3 describes our methodology. Section 4 describes our dataset and presents the results of our experiments. We draw our conclusions in Section 5, and discuss our limitations in Section 6.

## 2. Related Work

Taxonomy induction methods can broadly be categorized into traditional approaches and RL-based techniques. In this section, we briefly overview traditional approaches before focusing on advances using RL. Traditional methods for hierarchical detection are pattern-based (Hearst, 1992), offering high precision but low recall, or statistical, using background text statistics for identifying relations without manual syntax specification. For example, Fu et al. (2014) uses the spatial properties of embeddings like GloVe (Pennington et al., 2014) or Word2vec (Mikolov et al., 2013) to detect hypernym-hyponym pairs. For more information on traditional methods, please refer to Page 49 of Weikum et al. (2021).

Limited research exists on RL in taxonomy induction. Mao et al. (2018) argue that the two-phase taxonomy-induction setup, i.e. hierarchy detection and construction, is inherently suboptimal due to one-directional information flow. Their system, TaxoRL, unifies both phases, training a REINFORCE (Williams, 1992) agent to select and append a child node to a pre-existing parent in the taxonomy, which is also chosen by the agent. DTaxa (Han et al., 2021) builds on TaxoRL with an actor-critic approach, using a variant of the DDPG agent instead of REINFORCE, for faster learning and better performance. TaxoRL and DTaxa achieve competitive performance on taxonomy induction benchmarks.

Both TaxoRL and DTaxa face a common drawback in their action representations. They treat the selection of a term and its position in the taxonomy as a single action, missing the ability to discern different types of errors. For example, choosing a child node without a parent in the tree could lead to multiple incorrect parent choices. We argue that adjusting action handling to better align with the problem's structure and semantics could enhance taxonomy induction.

## 3. Methodology

We formulate taxonomy induction as an RL problem. The goal is to create a taxonomy that accurately organizes a given set of terms, aligning with the golden taxonomy. To achieve this, the model is provided with a large background corpus, allowing it to incorporate information about the relations of words as features in its action representation.

### 3.1. Problem formulation

Taxonomy induction is formulated as a finite and discrete Markov Decision Process (MDP) (Bellman, 1957). At the beginning of every episode, we start with a taxonomy tree containing only a single word (also known as a term). We expand the tree at each time step by appending a word from the term set as a child to one of the nodes in the tree until all the terms are added (i.e. the end of an episode). At each time step $t$, there is a set of words that are nodes in the taxonomy tree $U_t$, a set of remaining terms that are not yet part of the tree $V_t$ and a set of edges $E_t$, each of which connects two nodes in the taxonomy: $E_t \subseteq \{(U_t \times U_t)\}$. Furthermore, a root node $\text{ROOT}_t \in U_t$ serves as the root of the taxonomy tree. To give a concrete example of the notion of a state, we refer to Figure 1. We follow the standard definition in Sutton and Barto (2018)

---

[2] We refer to the choice of either one of the two terms as a "sub-action", while the complete action refers to the choice of both the child and parent terms.

[3] Our implementation of TaxoCritic and the Appendix file are publicly available at `https://github.com/BendeguzToth/taxonomy-construction`.

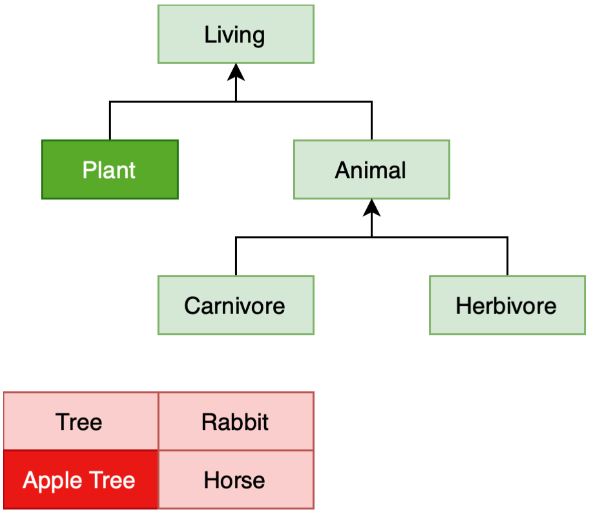


Figure 1: Example of an action $a_t$ = *(Apple Tree, Plant)*, where $root_t$ = {*Living*}, $U_t$ = {*Living, Plant, Animal, Carnivore, Herbivore*}, $V_t$ = {*Tree, Rabbit, Apple Tree, Horse*}, and $E_t$ = {*(Plant, Living), (Animal, Living), (Carnivore, Animal), (Herbivore, Animal)*}. After the execution of this action, $root_{t+1}$ = {*Living*}, $U_{t+1}$ = {*Living, Plant, Animal, Carnivore, Herbivore, Apple Tree*}, $V_{t+1}$ = {*Tree, Rabbit, Horse*}, and $E_{t+1}$ = {*(Plant, Living), (Animal, Living), (Carnivore, Animal), (Herbivore, Animal), (Apple Tree, Plant)*}.

and define the key elements as follows:

**State** The MDP contains a set of observed *states* $S$. The state $s_t \in S$ at any time step $t$ represents the taxonomy at time $t$, consisting of a collection of edges $E_t$, as well as the remaining term set $V_t$. Notably, there is no need to explicitly include the terms that are already part of the tree (the nodes) as they are implicitly represented by the edges. The state is formally denoted as $s_t = (E_t, V_t)$.

**Action** There is a set of *actions* $A$. An action $a_t$ ($a_t \in A$) can fall into one of two types:

- **Adding a new node as a child** In this case, the action $a_t$ takes the form $(v, u) \in (V_t \times U_t)$. The new term $v$ is added to the taxonomy as a child node to $u$. The update to the taxonomy at time step $t+1$ is as follows:

$$U_{t+1} = U_t \cup \{v\}, \quad V_{t+1} = V_t \setminus \{v\}$$
$$E_{t+1} = E_t \cup \{(v, u)\}, \quad \text{ROOT}_{t+1} = \text{ROOT}_t$$

- **Adding a new node as root** Alternatively, the current root is the child, and a new term is appended as its parent (resulting in the new term becoming the new root). This action $a_t$ is represented as $(\text{ROOT}_t, v)$ where $v \in V_t$. The common updates to set $U$ and $V$ at time step $t+1$ are the same as in the previous action. The specific updates to $E$ and ROOT are:

$$E_{t+1} = E_t \cup \{(\text{ROOT}_t, v)\}, \quad \text{ROOT}_{t+1} = v$$

By combining those two action possibilities, an action takes the following form:

$$a_t \in (V_t \times U_t) \cup (\{\text{ROOT}_t\} \times V_t) \tag{1}$$

**Transition** The transition from one state to another is deterministic, i.e. $Pr(s_{t+1}|s_t, a_t) = 1$. Thus, following the two action possibilities, the next state is determined by the updated taxonomy:

$$s_{t+1} = (E_{t+1}, V_{t+1})$$

**Reward** Similar to (Mao et al., 2018) and (Han et al., 2021), we utilize the difference in *Edge-F1* at each time step as the deterministic reward signal. *Edge-F1* is defined in Equation 2, where $E^*$ is the set of edges present in the golden taxonomy and $\overline{E}$ is the set of edges predicted by the model. The reward at time step $t$ is then $F_1^t - F_1^{t-1}$.

$$\begin{aligned} P &= \frac{|\overline{E} \cap E^*|}{|\overline{E}|}, \quad R = \frac{|\overline{E} \cap E^*|}{|E^*|} \\ F_1 &= \frac{2 \cdot P \cdot R}{P + R} \end{aligned} \tag{2}$$

### 3.2. Design Architecture

To address the issue of proper credit assignment in previous methods, we propose TaxoCritic, a single-actor and multi-critic RL algorithm that individually evaluates both sub-actions. This approach allows for assigning rewards (either positive or negative) to the two sub-actions independently, leading to better credit assignments. Inspired by existing multi-critic RL techniques (Martinez-Piazuelo et al., 2020; Mysore et al., 2021), we integrate the idea of multiple critics into the domain of taxonomy induction for the first time. Instead of relying on a single critic to estimate the value of an action, our algorithm incorporates two distinct critics, each dedicated to one of the sub-actions and their outputs are combined to produce the final estimate. More precisely, one of the critics assesses the choice of the parent node, while the other evaluates the choice of the child node. This design allows the sub-critics to be independent and simplifies model optimization by backpropagating only once from the combined action value. The actor, on the other hand, remains undivided and determines the best joint action to take. An overview of the TaxoCiritc framework is illustrated in Figure 2.

#### 3.2.1. Actor

In our method, the actor is a fully connected 2-layer feed-forward neural network. The design of the actor architecture poses a unique challenge due to

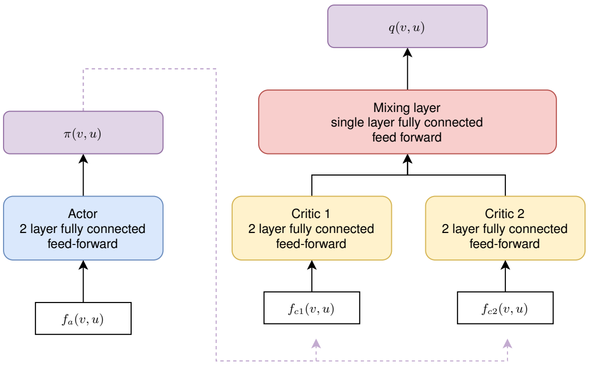


Figure 2: An overview of the TaxoCritic method. $f_a$, $f_{c_1}$ and $f_{c_2}$ represent the feature representations (vectors) of the inputs for the actor and two critics respectively. The **actor** (a two-layer fully connected feed-forward neural network) takes the encoding of a state as the input and outputs the policy $\pi$. Following this policy, the environment executes a sampled action which contains two sub-actions. As shown on the right side, the **critic** network features two sub-critics and a mixing layer. Considering the state and two sub-actions, one critic evaluates the child's choice, while the other evaluates the parent's choice. The mixing layer combines those results from both critics and produces the action value $q$.

the dynamic nature of our action space. Unlike agents trained for tasks like playing Atari games or controlling robotic arms, where the number of actions remains constant throughout the task (Mnih et al., 2015; Franceschetti et al., 2021), taxonomy induction demands a more flexible approach. As described in Section 3.1, the actions are defined by the number of terms left to be added to the tree, as well as the current nodes that are present in the taxonomy. These quantities are dynamic and change at each time step, making it impractical to adopt a standard architecture where the neural network takes only the state representation as input and outputs a probability distribution over a fixed number of possible actions. Thus, in our method, the policy network takes the features of a possible state-action pair $(s_t, a_t)$ as the input, generates the probability of taking that specific action in the given state (i.e. $Pr(a_t|s_t)$). This design allows accommodating an arbitrary number of actions. During the construction of a taxonomy, all possible action pairs at the current state are fed through the network, outputting a probability distribution over the valid action space through a Softmax function.

Moreover, a challenge arises from the variability in action semantics across different episodes. For instance, when constructing a taxonomy for the animal kingdom, and subsequently another one for different kinds of mining equipment, all the actions would have entirely different semantic meanings, despite the action space size remaining constant. In other words, the action corresponding to the first output value will probably have an entirely different interpretation in taxonomy $a$ than in taxonomy $b$. Thus, it is crucial to explicitly encode the actions themselves, as relying on constant positions is no longer sufficient. By using a network that incorporates action embeddings, the semantic meaning of each action can be communicated in the TaxoCritic.

### 3.2.2. Critic

In taxonomy induction, an action is expressed as an edge $(v, u)$ to be added to the taxonomy graph. An action can be split naturally into two sub-actions. One component of the action, $v$, denotes the new term that shall be added to the taxonomy as a child node, while $u$ denotes the parent node to connect the child node. This clear division between the two parts of the action allows us to employ distinct critics for assessing each sub-action independently. Therefore, in our model, the critic is divided into two distinct sub-critics. Each of the two sub-critics can only observe a part of the feature space (depending on which part of the action they focus on) and, as such, are responsible for rating different components of the action. The outcomes of these sub-critics are then merged with a single feed-forward layer neural network to obtain the final $q$ value estimate.

## 3.3. Feature Representation

In taxonomy induction, both states and actions are tuples of words. To capture the semantic features of the words, we use the embedding of state and action as inputs of neural networks. We generate the feature representations as in TaxoRL (Mao et al., 2018), adapted by DTaxa (Han et al., 2021). A syntax-level feature vector is constructed for every possible action, consisting of eight features: Capitalization, Endswith, Contains, LCS, LD, Normalized frequency difference, and Generality difference; see Appendix A for detailed information. The final vector is the concatenation of the embeddings for the vectors $v$ and $u$ corresponding to the action, the dependency path, and the syntax-level feature vector. Figure 3 depicts an overview of a feature vector for action $a_t = (v, u)$.

As previously outlined in Section 3.1, a state of the MDP has the form $s_t = (E, V)$. However, incorporating the remaining term set $V$ as part of the state feature is redundant since the action encoding already encapsulates this data. Therefore, we simplify the state representation to only $E$. The edges of a taxonomy at a given time $t$ correspond to the taken actions since each action effectively

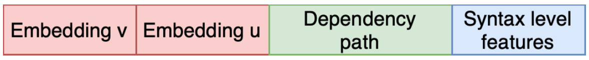


Figure 3: The action feature vector $a_t = (v, u)$ concatenates the word embeddings —using GloVe (Pennington et al., 2014)— for terms $v$ and $u$, their dependency path from the corpus, and syntactic features into one vector.

adds a new edge to the tree. The state $s_t$ is represented as the sequence of actions taken up to $t$:

$$s_t = (a_1, a_t, ..., a_{t-1}) \tag{3}$$

To represent this, all action embeddings $(a_1, a_2, ..., a_{t-1})$ are input to a single-layer Long short-term memory (LSTM) network that combines those values into one vector. As the model is fully differentiable, backpropagation into the LSTM parameters is straightforward.

For all the possible actions at time step $t$, the **actor** takes the feature representations of each pair of state and action and concatenates them as one feature vector input. On the other side, both **sub-critics** utilize different feature vectors based on the state and action representations. These two vectors are similar, essentially mirroring each other. The rationale behind this design is to ensure that when evaluating one sub-action, no assumptions are to be made about the other part of the action. This leads to two changes in feature representation compared to the one employed by the actor:

**Word embeddings:** Each sub-critic includes the word vector for only the term it evaluates. Specifically, one sub-critic uses the embedding of $v$, and the other uses the embedding of $u$.

**Relational features:** The dependency path and syntax level features can no longer be used, as they rely on knowing both words of the action. Instead of leaving those features out, they are modified in a way that requires knowledge of only one term. This is achieved by summarizing the relations between the known word and all its potential pairs. For example, if the chosen action by the policy network is the tuple $(v_i, u_j)$, then the critic responsible for the choice of the child node would take the relations of $v_i$ with every possible $u$ and average them to obtain an approximation. This averaged feature is called the *average shared feature* of critic 1. The average shared feature of critic 2 is constructed in a similar way, except the choice of $u_j$ is known, and the average is taken over all possible choices of $v$. A comprehensive formal definition of the shared features for the sub-critics is provided in Appendix B.

### 3.4. Training

The training of our model is done simultaneously by training the two sub-critic networks and the policy network. Both critics are trained jointly, with the gradient being distributed by the mixing function[4]. The loss is computed using the output of the mixing function, which aggregates the output values of both sub-critics. We refer to the entire value network (both sub-critics and the mixing layer) as *combined critic*. A comprehensive outline of the joint training algorithm is in Appendix C. All the experiments were run on a Linux virtual machine powered by an Intel Xeon Platinum CPU with 2 cores and 32 GB RAM. With this setup, running a single epoch took 50-60 minutes on average. Running the full experiment up to 300 epochs took over 11 days.

## 4. Experimental Results

To evaluate our model's performance and compare it with previous methods, we conducted a series of experiments. Our goal is to gain insights into the characteristics, strengths, and weaknesses of the algorithms by not only examining the final performance metrics but also by conducting qualitative analyses of the resulting taxonomies. We conducted three analyses:

1. **Ablation analysis.** We conducted an ablation analysis to evaluate specific features in our model, focusing on their individual contributions to overall performance.
2. **Performance assessment.** In this experiment, we evaluated the performance of our final model, as well as two of the baseline models —TaxoRL and DTaxa—, on a set of taxonomy induction tasks. We compared their results based on two evaluation metrics, examining the accuracy (i.e. edge F1 score) in individual runs as well as the robustness (i.e. consistency score) across different runs.
3. **Credit assignment analysis.** One of the motivations for choosing a multi-critic approach for our model was to improve credit assignment in the critic. In this qualitative analysis, we showcase how our critics have effectively learned the behavior patterns we outlined above.

### 4.1. Experimental Setup

**Experimental Environment** In our experiments, the dataset was split into training, validation, and

---

[4] Multiple constructions of the mixing layer have been experimented with e.g. QMIX-like architecture (Rashid et al., 2020). Here we only present the selected architecture with superior performance.

test sets with a distribution of 70/15/15, corresponding to 533/114/114 taxonomies respectively. Each model was trained for 300 epochs, and the resulting weights were saved for subsequent qualitative analysis. During an epoch, a single training episode is executed for every taxonomy in the training set, amounting to 533 episodes per epoch. An episode involves constructing a single taxonomy. At the start of each episode, a set of terms is provided, and the goal is to build up a taxonomy from said terms that match the target *golden taxonomy* as closely as possible. The agent's action interface only allows for the extension of an already existing taxonomy, requiring at least one node (*root*) to begin construction. To address this, like TaxoRL, we chose to start each episode with a randomly selected root node. The agent can then attach a new root node on top of it by selecting the node to be added as a parent and the current node as a child. This approach is intended to improve the model's robustness by reducing the potential for overfitting to a specific construction sequence for each taxonomy as it aims to allow the model to adapt to various starting points. All results are averaged over multiple turns.

**Hyperparameters** We explored various layer sizes and learning rates for both the actor and the critic networks. A similar trend was observed in the results, we therefore selected the optimal ones for the following results. Both the critic networks and the actor network consist of a single-neuron two-layer and a multi-neuron first layer (with 64 and 60 hidden nodes respectively). We set a learning rate of $5 \times 10^{-4}$ for the actor and $1 \times 10^{-4}$ for the critic. Moreover, we employed a ReLU activation function and the Adam optimizer (Kingma and Ba, 2014) for training. The discount factor is set to 0.95. For more information about parameter optimization, please refer to Appendix D.

## 4.2. Dataset

We used the WordNet taxonomy (Bansal et al., 2014), also utilized by TaxoRL and DTaxa. It encompasses a set of 761 taxonomies sampled from WordNet (Miller, 1995), each with a depth of three, built up from 10-50 nodes. While this dataset provides word sets and their corresponding target taxonomies, it does not specify the underlying background corpus. The agent's performance on the benchmark is heavily influenced by this background corpus, which is essential for extracting statistical relations among terms forming a crucial aspect of the feature representation during training. To ensure meaningful comparisons with prior methodologies, we opted to utilize the same background text as TaxoRL, which is an aggregation of Wikipedia dump, the UMBC web-based corpus (Han et al., 2013) and the One Billion Language Modelling Benchmark (Chelba et al., 2013).

## 4.3. Evaluation Metrics

**Edge-F1 score:** Similar to Mao et al. (2018), we first evaluate the Edge-F1 score. At the end of the episode, once all terms are incorporated into the taxonomy tree, the final construction is evaluated against the gold taxonomy. A detailed explanation of the edge score follows:

- Edge set in the constructed taxonomy: $E_{pred}$
- Edge set in the gold taxonomy: $E_{gold}$
- Edge precision: $P_e = |E_{pred} \cap E_{gold}|/|E_{pred}|$
- Edge recall: $R_e = |E_{pred} \cap E_{gold}|/|E_{gold}|$
- Edge-F1: $F1_e = 2 \cdot P_e \cdot R_e/(P_e + R_e)$

**Consistency scores:** To assess the model's robustness we introduce a consistency score, denoted as $C_{\text{root}}$, which measures the model's ability to converge consistently across different runs. It is calculated as a ratio of the number of consistent convergences where the model consistently identified the correct root $R_{\text{consistent}}$ to the total number of experimental runs $R_{\text{total}}$.

$$C_{\text{root}} = \frac{R_{\text{consistent}}}{R_{\text{total}}} \tag{4}$$

In addition, we also introduce a 'Consistency in Edge Score' $C_{\text{edge}}$, as the ratio of the number of correct edges identified across all runs to the total number of edges in the experiment $E_{total}$.

$$C_{\text{edge}} = \frac{\sum_{i=1}^{totalRun} |E_{pred_i} \cap E_{gold}|}{E_{total}} \tag{5}$$

## 4.4. Results

**Experiment #1 Ablation Analysis** In this experiment we assess the impact of two features: sibling embeddings and history inclusion. Sibling nodes $m$ and $n$ share the same parent node, and their embeddings are averaged and added to the feature vector of action $(v, u)$ if sibling embeddings are employed. The history feature encompasses a summary of past actions in the feature vector.

Surprisingly, omitting the history representation led to enhanced performance. We observed a consistent pattern when testing using TaxoRL algorithm. Making use of sibling embeddings, on the other hand, positively impacts the performance. Based on this analysis, we decided to leave out the history representation from both our model and

TaxoRL for the main experiment [5]. The results of this analysis can be seen in Table 1.

| History Usage | Sibling Usage | F1 after 150 epochs | F1 after 200 epochs |
|---|---|---|---|
| No | No | 0.3233 | 0.3328 |
| **No** | **Yes** | **0.3301** | **0.3434** |
| Yes | No | 0.1649 | 0.1724 |
| Yes | Yes | 0.2506 | 0.2596 |

Table 1: The result table showcases the Edge-F1 score of TaxoCritic model when certain features are omitted.

**Experiment #2 Accuracy Performance** We trained our model, TaxoRL, and DTaxa* [6] on the dataset described in Section 4.2. We average the results of each algorithm over three runs. The training results are illustrated in Figure 4. Despite its initial slow start, our method eventually outperforms TaxoRL in the experiment. The slower convergence at the beginning shows a characteristic difference between the two algorithms.TaxoRL trains its policy network based on sampled returns at each transition, yielding a noisy but unbiased return estimate. In contrast, our agent updates its policy network using the critic's output, which begins as random due to the critic's initial random initialization. However, once the critic's estimates stabilize, our policy's convergence accelerates and surpasses TaxoRL's speed. The graph further illustrates that DTaxa* significantly outperforms both other methods. A potential reason for DTaxa*'s better performance is attributed to the use of an efficient actor-critic algorithm DDPG. Table 2 shows the results of all methods after a specific number of epochs. For additional evaluation results, see Appendix E.

**Experiment #3 Robustness Performance** We assess our model's consistent convergence across various runs on a randomly selected taxonomy sample. We conducted five runs with five different initial root words. In TaxoCritic, the correct root was correctly identified in 3 out of 5 cases. In the remaining two instances, other terms were chosen as roots. Among the 55 total edges from the 5 runs, 31 were correctly identified. Among the incorrect edges, a pattern emerged: 11 out of the 24 erroneous edges had *guestroom* as their parent, indicating a systematic bias in the model. This bias is more manageable for practical use, as domain experts can focus on potentially flawed parts of the final taxonomy. This is especially beneficial for larger and more intricate trees. While we demonstrate this with a smaller example for clarity, similar principles can apply to more complex domains. The selected taxonomy sample and all the resulting structures are depicted in Appendix F.

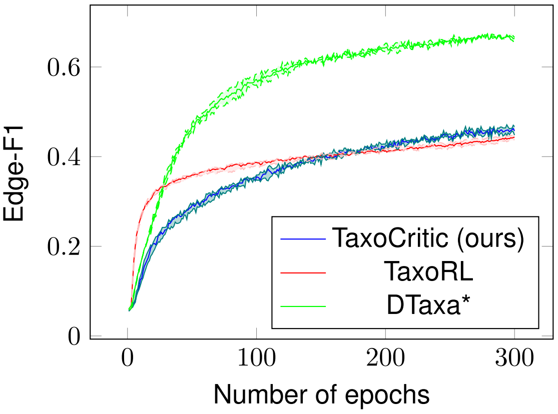


Figure 4: The training performance comparison graph of TaxoCritic (ours), TaxoRL, and DTaxa. The central darker line represents the average performance, while the lighter lines above and below indicate the range of minimum and maximum values across the runs.

| **Model** | **Epochs** | | |
|---|---|---|---|
| | **100** | **200** | **300** |
| TaxoRL | 0.386 | 0.413 | 0.443 |
| DTaxa* | **0.571** | **0.643** | **0.664** |
| TaxoCritic | 0.349 | 0.421 | 0.459 |

Table 2: Edge-F1 scores on the training set performance of the algorithms at different epochs.

We repeated the experiment using the same taxonomy with the two benchmark models. DTaxa* displayed a similar overall correctness, with 29 correct edges. However, it consistently struggled to identify the correct root node across all 5 runs, exhibiting a bias towards the term *connecting room*, frequently assigning it numerous children despite it being a leaf node in the golden taxonomy, resulting in 11 incorrect edges featuring it as their parent. TaxoRL achieved the lowest overall performance by correctly identifying 26 edges. However, it consistently identified the correct root in all 5 cases. The results are summarized in Table 3. Refer to Appendix F for the generated trees by DTaxa* and TaxoRL.

---

[5] Note that this analysis was run with an earlier version of the model before it was fully optimized, therefore the results are slightly lower than in the final experiment.

[6] In our effort to access the code associated with the DTaxa paper, we made attempts to contact the authors, but unfortunately, we did not receive a response. To conduct our experiments, we undertook the task of recreating their model to the best of our abilities, based on the limited information provided in the paper. We refer to this recreated model as DTaxa*. The code of DTaxa* is also provided in our project repository.

| Model | $C_{\text{root}}$ | $C_{\text{edge}}$ |
|---|---|---|
| TaxoRL | **1** | 0.47 |
| DTaxa* | 0 | 0.53 |
| TaxoCritic | 0.6 | **0.56** |

Table 3: Robustness Scores for the three Models.

**Experiment #4 Credit Assignment** The multi-critic architecture was adopted to enhance convergence speed and effective credit assignment by the critic. This concept was demonstrated through a hypothetical scenario where one sub-action could be held responsible for an incorrect action $(p, c)$ while the other sub-action could be a suitable choice. We examined our model's feasibility to exhibit this attribute by analyzing the construction of a small example taxonomy.

Figure 5 depicts the state of the tree at the specific point of interest. In this analysis, we will look at the sub-critic output values for potential actions within the partial taxonomy. With $V = \{$*nursery, day nursery, connecting room, adjoining room*$\}$, allowable actions are: Each term within $V$ can serve as a child node, either linked to an existing taxonomy node (denoted in yellow and blue), or any term from $V$ can function as a new node, becoming the parent of the current root node *bedroom*.

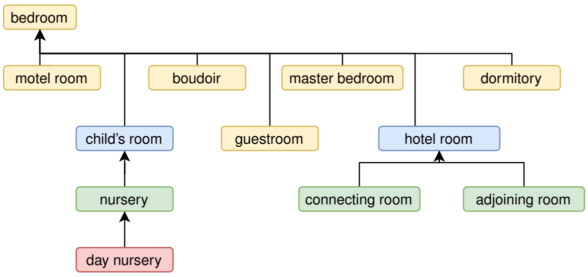


Figure 5: A simple partially constructed bedroom hierarchy. Yellow and Blue nodes (top three rows) denote correctly placed terms. Green and Red nodes (bottom two rows) are yet to be placed, while indicating their intended positions.

To analyze our approach's ability to learn credit assignment for sub-actions, we assess intermediate output values from two critics. To demonstrate the credit assignment of an individual sub-action, we focus on the average output of the child critic, responsible for estimating the value of selecting each potential node $v$ in $V$ as the child node.

In this example, selecting any of the blue terms as a child node is beneficial due to the presence of correct parent nodes for each of them in the tree. However, the red node *day nursery* lacks a suitable parent. We intuitively expect that the sub-critic for choosing the child node should assign lower values to actions involving *day nursery* compared to others. We therefore investigate the intermediate values assigned by the term-choice critic for each possible action. There are 8 potential parents for each child candidate term. Table 4 displays the average action values for every possible child candidate. Notably, *day nursery* has the lowest value of -5.14, while the average for green terms (with valid parents) is -4.32. This observation indicates that our critic prioritizes the green nodes as potential children, aligning with expectations. Meanwhile, since *day nursery* cannot be properly attached to any existing nodes, the critic assigns it a lower rating. For a similar analysis of the parent node selection, please refer to Appendix G, where we observe a similar trend.

This analysis demonstrates that our multi-critic algorithm is correctly assigning the blame when one sub-action is primarily responsible for the choice of incorrect action, without penalizing sub-actions that are conceptually correct but fail due to the wrong choice of the other sub-action. This property contributes to maintaining consistent action value estimates during training.

| Terms | Action value |
|---|---|
| nursery | -3.72 |
| connecting room | -4.76 |
| adjoining room | -4.47 |
| day nursery | -5.14 |

Table 4: Inverted action values show the critic's rating of each node's suitability as a child.

## 5. Conclusion

In this paper, we introduce TaxoCritic, a deep Reinforcement Learning-based approach for taxonomy induction that utilizes a multi-critic algorithm. Unlike previous methods treating all actions as independent, TaxoCritic divides actions into two distinct sub-parts, each assigned to its own critic. This framework enhances credit assignment by accurately attributing blame to the responsible action component in case of errors. While our approach did not surpass all baselines in learning performance, the enhanced credit assignment analysis and overall robustness performance highlight the potential of multi-critic strategies for taxonomy induction. In conclusion, we believe that our method can serve as a good foundation for further research on applying deep RL techniques to taxonomy induction in a promising direction. Further, we encourage the research community to explore integrating taxonomy induction methods into the linked data ecosystem to improve knowledge representation. This effort should ensure that the resulting structures are interoperable, semantically rich, and could be easily integrated into existing datasets.

## 6. Limitations

We anticipated that the critics in our methodology would easily adapt to their more constrained role and would be able to work together efficiently, and our credit assignment analysis confirmed this expectation as the critics effectively identified correct and incorrect sub-actions, a promising outcome. However, this success did not translate proportionally to overall performance. Despite outperforming TaxoRL, our model's performance was notably inferior to DTaxa*, a single-critic method. This is an unexpected outcome considering that both sub-critics performed as intended. Trying to pinpoint the reasons behind this performance deficit might be an interesting follow-up research. We also propose exploring the impact of employing an alternative mixing function to effectively merge insights from the sub-critics for the final value. In addition, although we follow the existing works to use the WordNet taxonomy for evaluation which has a depth limited to three, exploring our method's generalizability on datasets with varied depth levels would be an intriguing direction.

## Ethics Statement

Our data are taken from publicly available sources. For this reason, we do not expect that there are ethical issues or conflicts of interest in our work.

## 7. Bibliographical References

Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. 2014. Structured learning for taxonomy induction with belief propagation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051.

Richard Bellman. 1957. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684.

Ronald J. Brachman. 1983. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *Computer*, 16(10):30–36.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Lifted rule injection for relation embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1389–1399, Austin, Texas. Association for Computational Linguistics.

Andrea Franceschetti, Elisa Tosello, Nicola Castaman, and Stefano Ghidoni. 2021. Robotic arm control and task training through deep reinforcement learning. In *International Conference on Intelligent Autonomous Systems*, pages 532–550. Springer.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209.

Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52.

Yongming Han, Yanwei Lang, Minjie Cheng, Zhiqiang Geng, Guofei Chen, and Tao Xia. 2021. Dtaxa: An actor–critic for automatic taxonomy induction. *Engineering Applications of Artificial Intelligence*, 106:104501.

Sanda M Harabagiu, Steven J Maiorano, and Marius A Paşca. 2003. Open-domain textual question answering techniques. *Natural Language Engineering*, 9(3):231–267.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jiaqing Liang, Yi Zhang, Yanghua Xiao, Haixun Wang, Wei Wang, and Pinpin Zhu. 2017. On the transitivity of hypernym-hyponym relations in data-driven lexical taxonomies. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1185–1191. AAAI Press.

Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-end reinforcement learning for automatic taxonomy induction. *arXiv preprint arXiv:1805.04044*.

Juan Martinez-Piazuelo, Daniel E Ochoa, Nicanor Quijano, and Luis Felipe Giraldo. 2020. A multi-critic reinforcement learning method: An application to multi-tank water systems. *IEEE Access*, 8:173227–173238.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Siddharth Mysore, George Cheng, Yunqi Zhao, Kate Saenko, and Meng Wu. 2021. Multi-critic actor learning: Teaching rl policies to act with style. In *International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203.

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian Suchanek. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations and Trends® in Databases*, 10(2-4):108–490.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Nansu Zong, Hong-Gee Kim, and Sejin Nam. 2017. Constructing faceted taxonomy for heterogeneous entities based on object properties in linked data. *Data & Knowledge Engineering*, 112:79–93.

# A. Actors's Features

In Table 5, we list a detailed description of the features utilized by the Actor.

Table 5: Features and Descriptions

| **Features** | **Description** |
|---|---|
| Capitalization | Whether any (or both) of the words are capitalized. |
| Endswith | If the second word ends with the first word (for example, for the pair (bear, polar bear), this would fire.) |
| Contains | If the second word contains the first word. |
| Suffix match | The number of matching trailing letters. |
| LCS | The length of the longest continuous substring contained by both words. |
| LD | Length difference between the words. $10 * \frac{\lvert w_1\rvert - \lvert w_2\rvert}{\lvert w_1\rvert + \lvert w_2\rvert}$ |
| Normalized frequency difference | The ratio between the frequency of pair $(v, u)$ and the most frequent parent of $v$, $u'$: $\frac{\text{freq}(v,u)}{\max_{u'} \text{freq}(v,u')}$. |
| Generality difference | The generality $g(v)$ of term $v$ is the logarithm of the number of its distinct hyponyms. The generality difference of the pair $(v, u)$ is defined as $g(u) - g(v)$. |

# B. Sub-critics' Features

In this appendix section, Figures 6 and 7 illustrates how the features outlined in Equation 6 are used by the two sub-critics. The shared features for the sub-critics as mentioned in Section 3.2.2 are defined as:

$$
\begin{aligned}
&f(v,u) : \text{dependency path and syntax features of the term pair } (v,u)\\
&f_{c1}(v) : \text{The average shared feature of critic 1, where the child is } v.\\
&\text{The mean is taken of all feature vectors with } v \text{ as child.}\\
&f_{c2}(u) : \text{The average shared feature of critic 2, where the parent is } u.\\
&\text{The mean is taken of all feature vectors with } u \text{ as parent.}\\
&f_{c1}(v) = \frac{\sum_{u \in U} f(v,u)}{|U|}\\
&f_{c2}(u) = \frac{\sum_{v \in V} f(v,u)}{|V|}
\end{aligned}
\tag{6}
$$

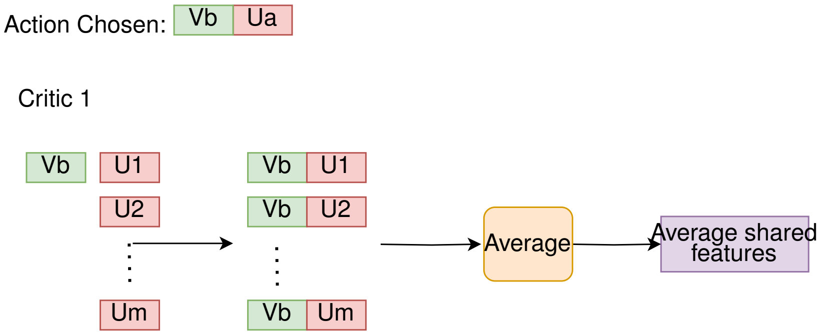


Figure 6: Shared feature summary of sub-critic 1. This sub-critic is only aware of the choice of the child term. To obtain the dependency path and syntax level features, it takes the features with all possible parent terms, then averages them.

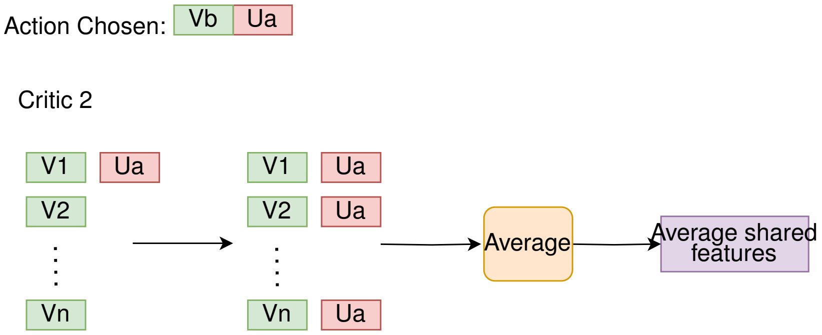


Figure 7: Shared feature summary of sub-critic 2. This sub-critic is only aware of the choice of the parent term. To obtain the dependency path and syntax level features, it takes the features with all possible child terms, then averages them.

### B.1. Network architecture

The value network is built up of three distinct parts. This is illustrated in Figure 8. There is one network for both critics. Those share the same architecture, with two fully connected layers and a ReLU in between. The input vector contains a word embedding (the embedding of $v$ in the case of critic1 and the embedding of $u$ in the case of critic2), the appropriate average shared features, and the state representation. The input size is 140. The first fully connected layer consists of 64 neurons, while the second layer is just a single neuron. The output is interpreted as the value of the sub-action. The last part of the critic is the mixing layer. It is a simple, single-layer feed-forward neural network that takes the two sub-action values and combines them into the final action value. We experimented with different mixing functions, most notably a QMIX-like architecture (Rashid et al., 2020), but we found a simple fully-connected layer to be more performant.

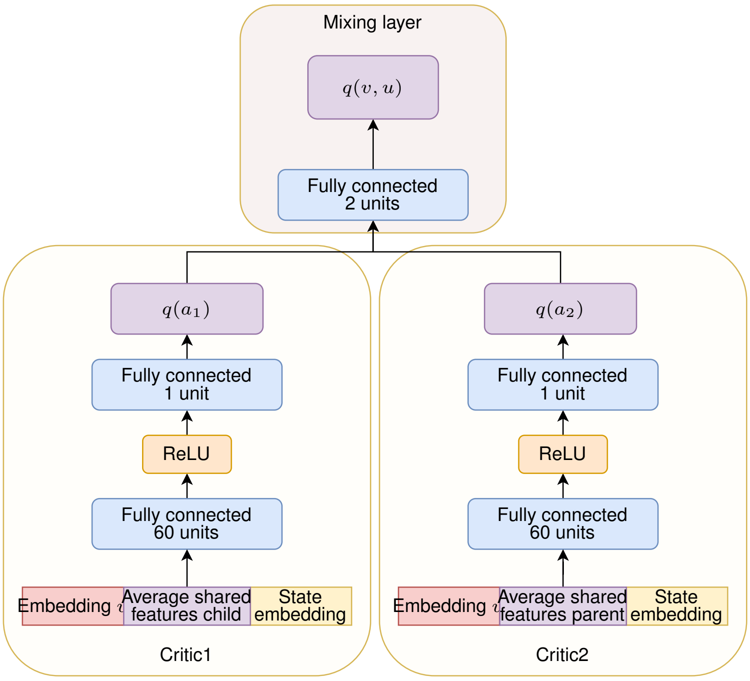


Figure 8: The architecture of the critic. $q(v, u)$ is the action-value of the action $(v, u)$, and $q(a_i)$ is the value of sub-action $i$.

## C. TaxoCritic Training Algorithm

Algorithm 1 describes the joint training process of the agent in detail. The two critics are trained jointly, with the gradient being distributed by the mixing function. The loss is calculated based on the output of the mixing function, that combines the output values of both sub-critics. In the pseudo-code below we refer to the entire value network (both sub-critics and the mixing layer) as *combined critic*.

```
DEFINITIONS;
$D$ : Training dataset;
$\alpha_c, \alpha_a$ : Critic and actor learning rate;
$\gamma$ : Rewards discount rate;
$\tau$ : Target update rate;
$\mu$ : Actor parameters;
$\theta, \theta'$ : Combined critic and combined critic target parameters;
$\pi, q$ : Policy and value functions;
$s, r$ : State and reward representations;
buff: Replay buffer;
INITIALIZATION;
buff $\leftarrow \emptyset$;
Initialize $\theta, \mu$ randomly;
$\theta' \leftarrow \theta$;
for $(V, U, E) \in D$;                          // For each taxonomy in the training set.
 do
    while $|V| > 0$;                          // Repeat until the remaining term set is empty
     do
        $A = (V \times U) \cup (\{\text{ROOT}\} \times V)$;              // 'A' is the set of all actions.
        LP $\leftarrow \{\pi_\mu(s, a)$ for all $a \in A\}$;     // 'LP' is the vector of log probabilities of all
         actions.
        $(v, u) \leftarrow$ sample(Softmax(LP));                       // Sample action
        $V \leftarrow V \setminus \{v\}$;                    // Update the taxonomy with the selected action
        $U \leftarrow U \cup \{v\}$;
        $E \leftarrow E \cup \{(v, u)\}$;
        buff.add($s, (v, u), r, s'$);        // Add (state, action, reward, next state) to buffer
    end
    ;                                        // After the episode ends train on all transitions.
    for $(s, a, r, s') \in$ buff;                           // For each transition in buffer
     do
        target $= r + \gamma q_{\theta'}(s', a)$;                    // Calculate the critic target
        $L_c = (\text{target} - q_\theta(s, a))^2$;                         // Combined critic loss
        $L_a = \ln(\pi_\mu(s, a)) \cdot q_\theta(s, a)$;                            // Actor loss
        $\theta \leftarrow \theta + \alpha_c * \nabla_\theta L_c$;                       // Updating the parameters
        $\mu \leftarrow \mu + \alpha_a * \nabla_\mu L_a$;
    end
    $\theta' \leftarrow \tau \cdot \theta + (1 - \tau) \cdot \theta'$;                        // Updating target params
    buff $\leftarrow \emptyset$;
end
```

**Algorithm 1:** Joint training of the actor and the combined critic

# D. Hyperparameters

## D.1. Learning Rate

Specifically, we conducted experiments to determine the optimal learning rate pairs for both the actor and critic networks. The outcomes of this analysis are detailed in Table 6.

## D.2. Path LSTM Dimensions

The path LSTM is an important part of the model, as it is responsible for summarizing the information about the relation of two words into a fixed-size representation.The dimension of this LSTM layer significantly affects the final performance. To determine the optimal dimensionality, we conducted an experiment, and the results are presented in Table 7.

| Learning Rate | | Edge-F1 |
|---|---|---|
| **Actor** | **Critic** | **(150 epochs)** |
| $1 \times 10^{-4}$ | $1 \times 10^{-4}$ | 0.2816 |
| $5 \times 10^{-4}$ | $1 \times 10^{-4}$ | **0.3301** |
| $1 \times 10^{-3}$ | $1 \times 10^{-4}$ | 0.2041 |

Table 6: Results of learning rate analysis for the TaxoCritic model

| Path LSTM Dimension | Edge-F1 at 150 epochs | Edge-F1 at 200 epochs |
|---|---|---|
| 60 | 0.3301 | 0.3434 |
| 128 | 0.3353 | 0.3354 |
| 256 | 0.3208 | 0.3303 |

Table 7: Performance analysis of the TaxoCritic model with different path LSTM dimensions.

# E. Additional Experimental Results

Results on the evaluation set are reported in Table 8, while the edge training results are reported in Table 9.

| Model | Epochs | | | | |
|---|---|---|---|---|---|
| | **100** | **150** | **200** | **250** | **300** |
| TaxoRL | 0.063 | **0.066** | **0.069** | **0.068** | **0.065** |
| DTaxa* | **0.065** | 0.063 | 0.067 | **0.068** | 0.053 |
| TaxoCritic | 0.063 | 0.062 | 0.062 | 0.067 | 0.058 |

Table 8: The Edge-F1 score on the evaluation set performance of the of all three models, TaxoRL, DTaxa*, and TaxoCritic after given number of epochs.

# F. Robustness Results

## F.1. TaxoCritic Robustness

Figure 9 illustrates the example taxonomy. While Figure 10 shows the tree generated by TaxoCritic to evaluate the robustness.

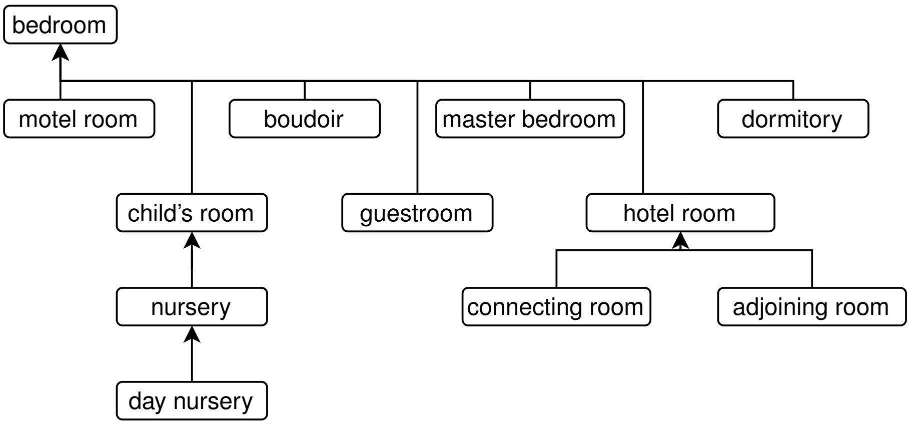


Figure 9: The example taxonomy.

| **Model** | **Edge Precision** | **Edge Recall** | **Edge F1** |
|---|---|---|---|
| TaxoRL | 0.23 | 0.427 | 0.299 |
| DTaxa* | **0.55** | **0.662** | **0.601** |
| TaxoCritic | 0.321 | 0.443 | 0.372 |

Table 9: Final precision, recall, and F1 scores after training of all three models, TaxoRL, DTaxa*, and TaxoCritic.

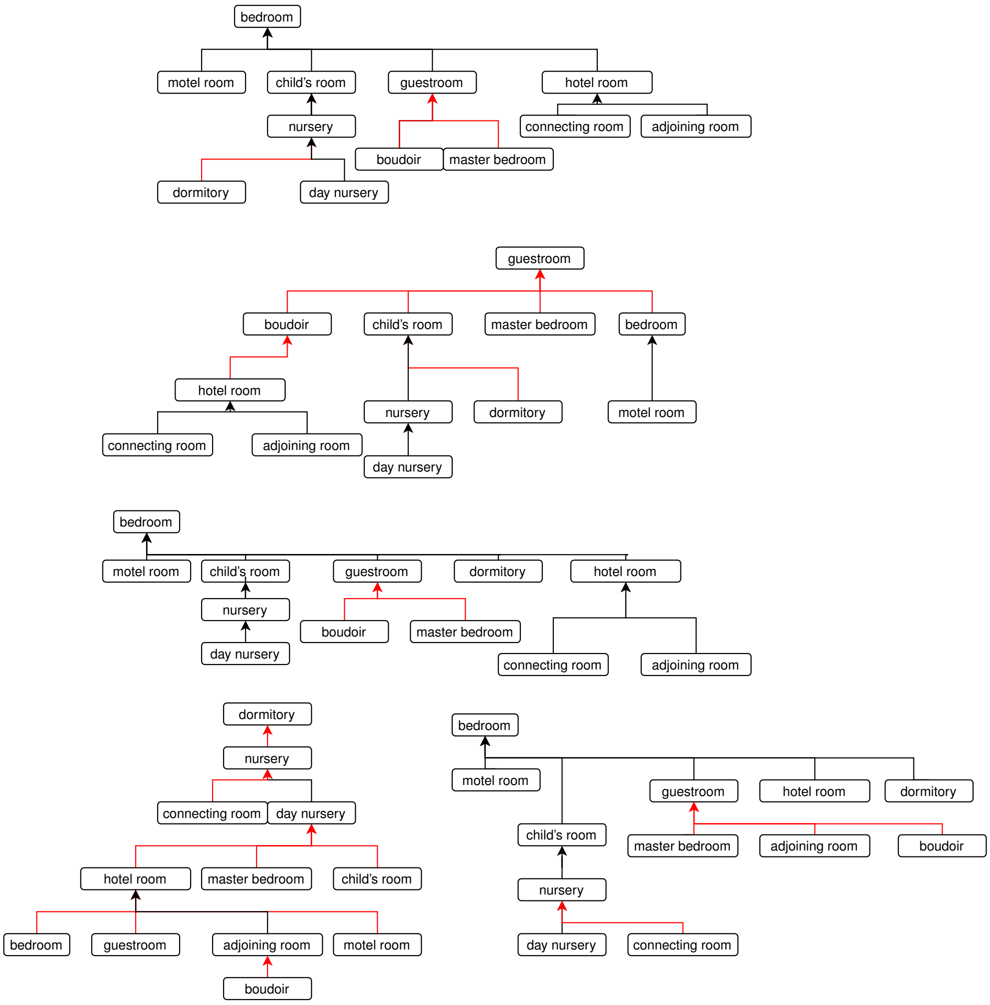


Figure 10: Trees generated by our model to evaluate TaxoCritic’s robustness. Red arrows indicate incorrect edges, while black edges represent correct ones.

## F.2. DTaxa* Robustness

Figure 11 shows the taxonomy trees generated by DTaxa* during the robustness analysis. Only three different trees were generated, two of the trees occurring twice each.

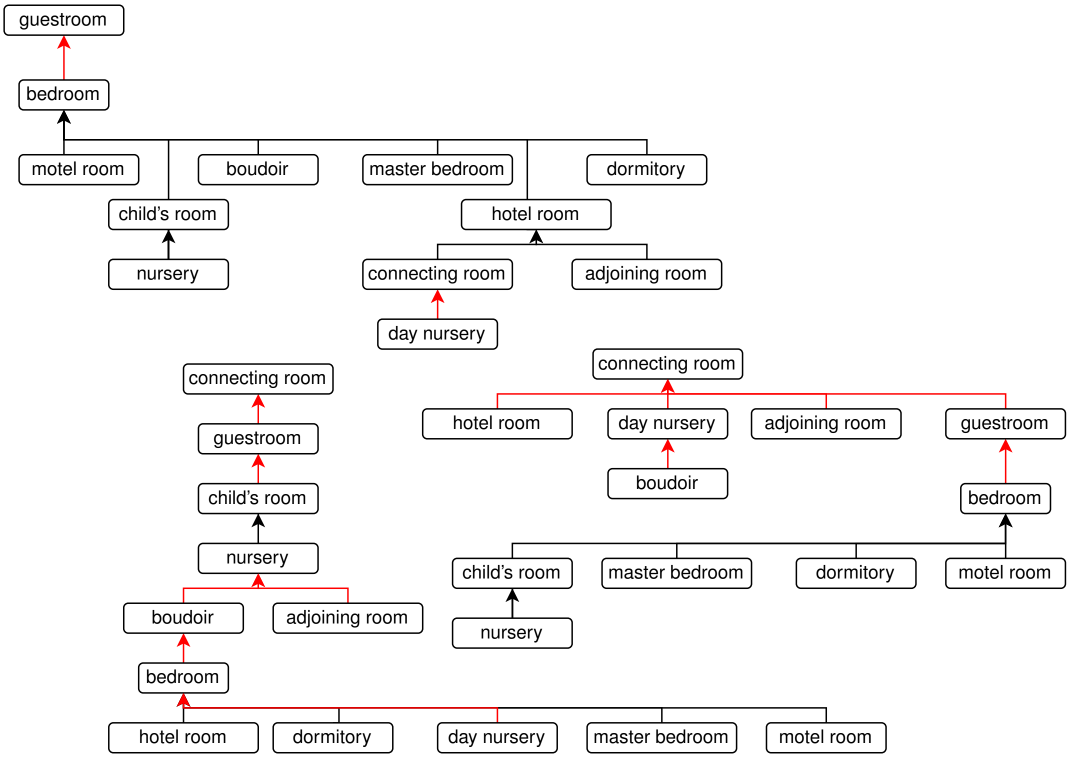


Figure 11: The trees generated by DTaxa* in five attempts, with two trees produced twice. Incorrect edges are indicated by red arrows, while the black edges represent correct identifications.

### F.3. TaxoRL Robustness

Figure 12 shows the taxonomy trees generated by TaxoRL during the robustness analysis. Only three different trees were generated, with the first one occurring three times.

## G. Credit Assignment Analysis

Referring to Figure 5, we notice that only selecting one of the blue nodes (*child’s room, hotel room*) as parents leads to a correct action, as none of the yellow terms can be a parent to any of the potential children. Thus, we anticipate the blue terms to have a higher action values than the yellow ones. Table 10 displays the values of the parent candidates. The average value of the incorrect parent candidates (yellow) is -6.92, while the average value of the correct parent choices (blue) is -4.04. Once again, we observe the same phenomenon as earlier: the sub-critic effectively prioritizes choices that are meaningful independently, even without specific information about the other component of the action (i.e., the choice of child in this case).

| | bedroom | boudoir | motel room | guestroom |
|---|---|---|---|---|
| **Action value** | -5.76 | -4.87 | -5.87 | -5.24 |

| | master bedroom | dormetry | child’s room | hotel room |
|---|---|---|---|---|
| **Action value** | -9.00 | -10.76 | -4.71 | -3.37 |

Table 10: This table shows the inverse actions values of choosing each of the nodes as the parent

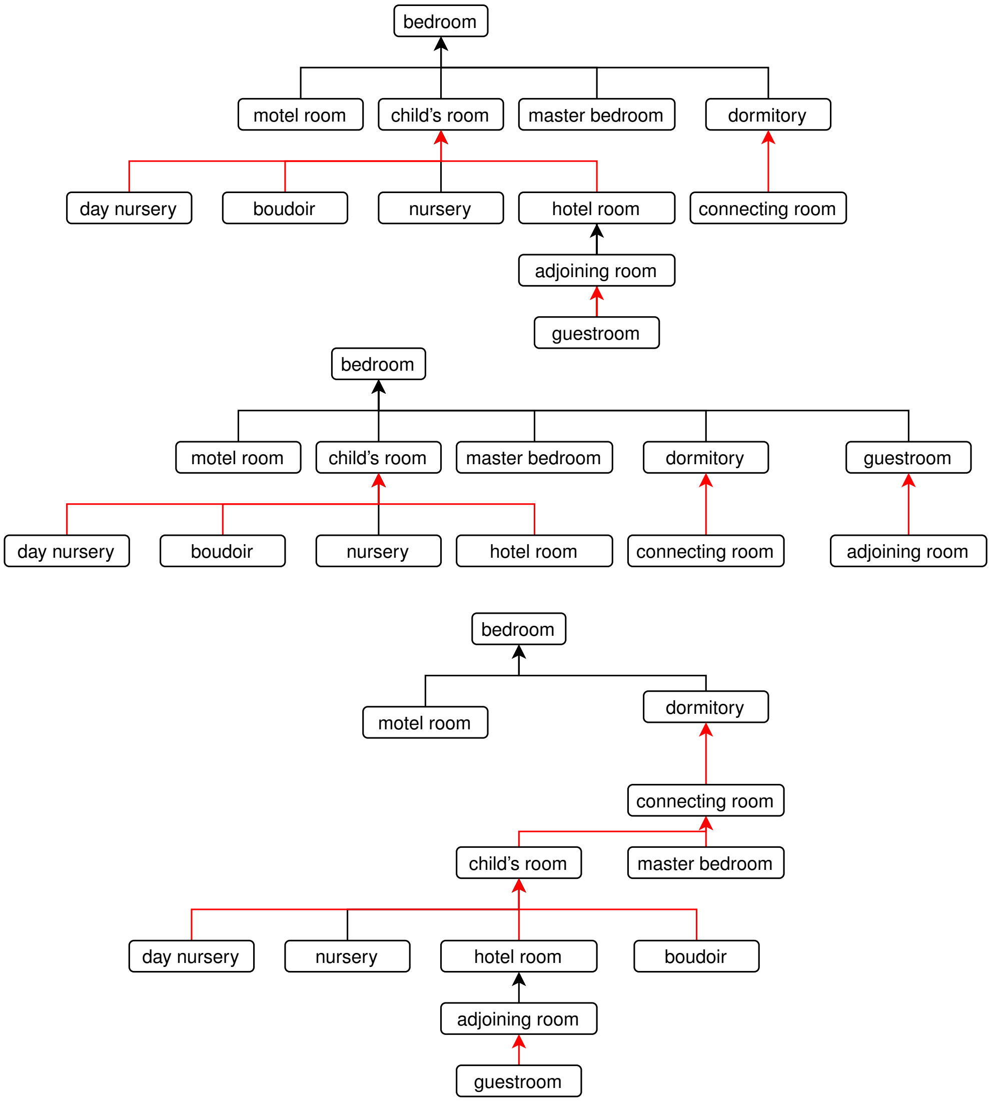


Figure 12: The trees generated by TaxoRL in five attempts, with the first tree occurring three times. Incorrect edges are highlighted by red arrows, while the black edges indicate correct identifications.

# Combining Deep learning Models and Lexical Linked Data: some insights from the development of a Multilingual News Named Entity Recognition and Linking dataset

**Emmanuel Cartier, Emile Peetermans**
European Commission, Joint Research Center
Via Enrico Fermi, 2749
21027 Ispra (VA), Italy
emmanuel.cartier@ec.europa.eu, peetermansemile@gmail.com

### Abstract

This paper presents the methodology and outcomes of a Named Entity Recognition and Linking multilingual news benchmark that leverages both Deep learning approaches by using a fine-tuned transformer model to detect mentions of persons, locations and organisations in text, and Linguistic Linked Open Data, through the use of Wikidata to disambiguate mentions and link them to ontology entries. It shows all the advantages of combining both approaches, not only for building the benchmark but also for fine-tuning detection models. We also insist on several perspectives of research to improve the accuracy of a combining system and go further on leveraging the complementary approaches.

**Keywords:** Named Entity Recognition and Linking, Transformers, Deep learning, Wikidata

## 1. Introduction

Named entity recognition (NER), disambiguation, and linking (abusively dubbed as Named Entity Linking or NEL) represent a trio of critical tasks within the field of natural language processing (NLP). These tasks are concerned with the extraction and classification of specific references from text, including but not limited to individuals, organizations, geographical locations, and other entity mentions such as dates and emails.

Following the progress in natural language processing, the current state-of-the-art systems are all based on deep learning systems, especially based on the Transformer architecture generating pre-trained models then fine-tuned for the task. But it appears that these systems, used alone, still struggle when context is sparse or noisy or far from the training data characteristics. For example, in the last Multilingual Complex Named Entity Recognition competition (SemEval 2023 (SemEval 2023 task 2: MultiCoNER II), the winning system does not only leverage these pre-trained contextual models, but also multilingual lexical knowledge bases, namely Wikipedia and Wikidata, especially to disambiguate and link mentions of named entities to knowledge bases entries. The combination consists in this case in creating sentence embeddings from Wikipedia instances linked to title entries, retrieve the most similar contexts to the one to annotate (semantic search) and then feed a Conditional Random Field (CRF) to generate the token annotation. In the same vein, current cutting edge systems (e.g. Wikineural) combine pre-trained models with fine-tuning from silver-annotated versions of Wikipedia mentions of Named Entities.

Additionally, and surprisingly provided that NER and NEL tasks are on the table for dozen of years, if several reputed benchmarks exist for NER evaluation, NEL evaluation benchmark are still far behind, even if a recent work has proposed a silver dataset (ie a dataset without human validation) from Wikipedia ((Kubeša and Straka, 2023a)). In this context, this paper will provide some insights of the advantages and current limitations of a combination of Deep Learning (DL) systems and Linked Open Data (LOD) knowledge bases, from an experiment aiming to design and compile a new NER and NEL benchmark, created specifically for the purpose of evaluating any system of NER/NEL on Multilingual news textual data.

The paper is further divided into three parts: the first part presents the existing systems for NER And NEL and give some key characteristics of these systems, insisting on the new trend of combining Deep learning systems and Linked Open Ontologies and Lexicons. The second part details the methodology and steps followed to construct this dataset as well as key figures on it. The third and conclusive part presents our insights from this experiment on the DL - LOD combination and some perspectives to push it even further.

## 2. Named Entity Recognition and Linking State-of-The-Art

This section presents the current SOTA models for NER and NEL, and then the most used knowledge bases available.

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 31–44
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

## 2.1. Deep learning language Models

In this section, we highlight some key characteristics of State-of-the-Art (SOTA) language models for NER and NEL, in the context of the Europe Media Monitor (EMM) we intend to apply them to. SOTA systems for NER/NEL - as all computational linguistic tasks - all rely on embedding representation and pre-trained language models (LM). Several competing LMs with desired features (multilinguality, open source, SOTA on relevant benchmarks) are available. To name the most prominent ones:

- XLM-Roberta-large (XLM-R) (Conneau et al., 2019a): this transformer-based masked language model is the base model for multilingual computational tasks. It was trained on one hundred languages, using more than two terabytes of filtered CommonCrawl data. This model outperformed one of the first multilingual model, multilingual BERT (mBERT) (Devlin et al., 2019). The XLM-R model is still present in leaderboards as a base (see e.g. XTREME benchmark, (Hu et al., 2020), and TNER list of fine-tuned Roberta models (TNER list of fine-tuned models)). These models have been fine-tuned on several datasets, as in WikiNEuRal (Tedeschi et al., 2021c), which combines a multilingual lexical knowledge base (i.e., BabelNet) and transformer-based architectures (i.e., mBERT) to produce high-quality annotations for multilingual NER. An mBERT model fine-tuned on this silver dataset reach an overall accuracy of 0.80.

- New SOTA multilingual models: these models exhibit strong performance on multilingual tasks and should be considered as swiss-knife pre-trained models. These next-generation transformer models add new tasks during the pre-training steps and evaluation leaderboards show that they enable to gain additional quality. Among them Turing ULR v6, (Patra et al., 2022a) adds a new task at the pre-training step, called cross-lingual contrast (XLCO). The goal of XLCO is to maximize mutual information between the representations of parallel sentences c1 and c2, i.e., I(c1, c2). It leverages this new task by creating multi bi-texts. VECO 2.0, (Zhang et al., 2023a), is the most recent SOTA model on the XTREME benchmark, going a step further by aligning not only sentence but also tokens of the bitexts.

- SOTA Models specific to NER/NEL: these specialised models are the winners of the most recent NER/NEL competitions (semEval 2022 task 11: and semEval 2023 task 2:): mLUKE, (Ri et al., 2021a) built on XML-RoBERTa, and additionally trained on 24 languages with entity representations taken from Wikipedia. The model consistently outperforms word-based pre-trained models in various crosslingual transfer tasks. KB-NER, (Wang et al., 2022a) multilingual knowledge base based on Wikipedia to provide related context information to the named entity recognition (NER) model. Given an input sentence, the system retrieves related contexts from the knowledge base. The original input sentences are then augmented with such context information, allowing significantly better contextualized token representations to be captured. Winner on 10 over 13 subtasks (semEval 2022 task 11). in the same vein and a similar architecture, U-RaNER won the semEval 2023 competition:Github repo.

As can be seen from on-going competition, especially to adapt the systems to more complex named entities, new domains and low-resourced languages, even if NER and NEL have now a long trail of research, there are still ways to improve the systems. The last winners of the Multiconer competition show that the main avenue to improve the current systems is to combine the pre-trained transformer models with external knowledge bases, in two main ways:

- by using directly available structured knowledge bases, i.e. Wikidata, especially the feature linking entries to their mention variants, directly at the recognition stage,

- by fine-tuning a secondary transformer model from a textual knowledge base (Wikipedia being the most used) and use it as a complementary resource if the pre-trained model needs additional context information to detect mentions.

## 2.2. Named Entity Linking Knowledge bases

We list here the main existing evaluation datasets again keeping in mind the multilingual and genre features.

- Mewsli-9 (Botha et al., 2020): this dataset contains manually labelled WikiNews articles in 9 different languages. New formulation for multilingual entity linking, where language-specific mentions resolve to a language-agnostic Knowledge Base. A dual encoder was trained in this new setting, building on prior work with improved feature representation, negative mining, and an auxiliary entity-pairing task, to obtain a single entity retrieval model that covers 100+ languages and 20 million entities. The model outperforms state-of-the-art results from a far more limited cross-lingual linking task. Rare entities and low-resource languages pose challenges at this large-scale.

- Mewsli-X (Ruder et al., 2021a): Mewsli-X is a multilingual dataset of entity mentions appearing in WikiNews and Wikipedia articles, that have been automatically linked to WikiData entries. The primary use case is to evaluate transfer-learning in the zero-shot cross-lingual setting of the XTREME-R benchmark suite: fine-tune a pre-trained model on English Wikipedia examples; evaluate on WikiNews in other languages — given an entity mention in a WikiNews article, retrieve the correct entity from the predefined candidate set by means of its textual description. Mewsli-X constitutes a doubly zero-shot task by construction: at test time, a model has to contend with different languages and a different set of entities from those observed during fine-tuning.

- DaMuEL (Kubeša and Straka, 2023b): a large Multilingual Dataset for Entity Linking containing data in 53 languages. DaMuEL consists of two components: a knowledge base that contains language-agnostic information about entities, including their claims from Wikidata and named entity types (PER, ORG, LOC, EVENT, BRAND, WORK-OF-ART, MANUFACTURED); and Wikipedia texts with entity mentions linked to the knowledge base, along with language-specific text from Wikidata such as labels, aliases, and descriptions, stored separately for each language.

As a matter of fact, even for multipurpose evaluation for LLMs, the WIKIANN benchmark remains the *de facto* standard for multilingual evaluation of core Named Entities, but it is exclusively built from Wikipedia and is known to contain a lot of errors. But, with Mewsli-9 and -X and DaMuEL, new silver standard datasets are built from a combination of DL models and existing Knowledge bases, namely Wikipedia, Wikinews and, as an aggregating KB, Wikidata. This combination of both approaches has the merit of enabling the building of large datasets that in turn can be used to fine-tune DL models. As an inspiration, we will use the DaMUEL dataset that has been built from Wikipedia, by applying a similar method to its news counterpart, Wikinews. We will detail the methodology after a presentation of State-of-The-Art models for NER and NEL. That will enable also to support the need of a new Multilingual News dataset.

# 3. WiNNL (WikiNews Named entity recognition and Linking)

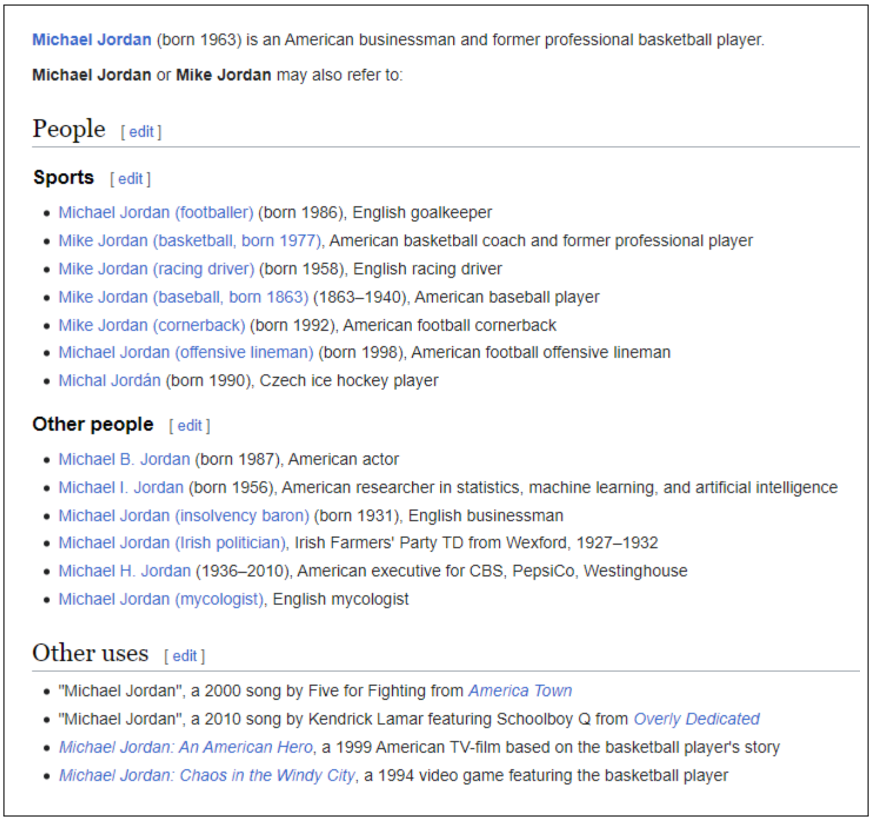
This section introduces WiNNL (WikiNews Named entity recognition and Linking), a new multilingual NER & NEL benchmark based on Wikinews articles. Wikinews, a free-content news source from Wikimedia Foundation, provides a rich and diverse environment for creating a realistic benchmark as it incorporates a wide range of topics and languages. Our benchmark, which for now encompasses 11 European languages, aims to provide a rigorous evaluation framework for multilingual NER/NEL systems. It also facilitates an understanding of how these models perform across different languages on the specific domain of news articles.

Our approach is inspired by the DaMuEL Wikipedia based benchmark (Kubeša and Straka, 2023b). In this work, the authors describe a pipeline to convert Wikipedia articles by detecting entity types using Wikidata and propagating mentions throughout the article.

## 3.1. Existing Benchmarks

Within the Joint research Center (JRC) Text and Data Mining Unit, we are facing the challenge of detecting and linking named entities within a live stream of retrieved news articles from more than 80 languages. Our main interest is to detect Persons, Locations and Organisations, as well as temporal information. The linguistic processing chain already includes a NER and NEL dictionary and rule-based system, setup and maintained for more than two decades, and we are in the phase of renewing it with more accurate systems based on Deep learning architectures and pre-trained language models. The first step is to evaluate such state-of-the-art models and the current system towards a benchmark tailored to our needs and constraints, as defined above.

In the core named entities recognition task, the WIKIANN dataset is the most used silver standard, especially in multilingual settings (see XTREME benchmark for example), but it does not correspond to news style and its low quality is often highlighted. Apart from the SlavNER dataset that enables to evaluate slavic languages, all the other datasets are more interesting for specific Named Entities or difficult cases (e.g. MultiCoNER 1 and 2). As a result, there is a strong need for developing a gold standard for multilingual news genre.



**Michael Jordan** (born 1963) is an American businessman and former professional basketball player.

**Michael Jordan** or **Mike Jordan** may also refer to:

People [ edit ]

**Sports** [ edit ]

- Michael Jordan (footballer) (born 1986), English goalkeeper
- Mike Jordan (basketball, born 1977), American basketball coach and former professional player
- Mike Jordan (racing driver) (born 1958), English racing driver
- Mike Jordan (baseball, born 1863) (1863–1940), American baseball player
- Mike Jordan (cornerback) (born 1992), American football cornerback
- Michael Jordan (offensive lineman) (born 1998), American football offensive lineman
- Michal Jordán (born 1990), Czech ice hockey player

**Other people** [ edit ]

- Michael B. Jordan (born 1987), American actor
- Michael I. Jordan (born 1956), American researcher in statistics, machine learning, and artificial intelligence
- Michael Jordan (insolvency baron) (born 1931), English businessman
- Michael Jordan (Irish politician), Irish Farmers' Party TD from Wexford, 1927–1932
- Michael H. Jordan (1936–2010), American executive for CBS, PepsiCo, Westinghouse
- Michael Jordan (mycologist), English mycologist

Other uses [ edit ]

- "Michael Jordan", a 2000 song by Five for Fighting from *America Town*
- "Michael Jordan", a 2010 song by Kendrick Lamar featuring Schoolboy Q from *Overly Dedicated*
- *Michael Jordan: An American Hero*, a 1999 American TV-film based on the basketball player's story
- *Michael Jordan: Chaos in the Windy City*, a 1994 video game featuring the basketball player

Figure 1: Named Entity Linking illustration (Wikipedia page)

## 3.2. Methodology to setup the Multilingual News dataset

### Dataset design

WiNNL's annotation scheme prioritises three core categories of entities: PER, ORG and LOC. These categories refer to person names, organisations and geographical locations, respectively. We opt for a word-level annotation scheme, where a word can be tagged as being the beginning of an entity, inside of an entity mention or outside of any annotation. This is indicated by Inside-Outside-Beginning (IOB) tags, where the prefix I- or B- is attached to the type (PER, ORG or LOC) of the entity for each word (Ramshaw and Marcus, 1999).

### Data collection process

The process of collecting and cleaning our multilingual NER/L dataset is initiated by downloading the HTML of articles from Wikinews. This source was chosen due to its extensive cross-linguistic coverage and the rich network of interlinked entities it contains. In Wikinews, authors of articles generally tag each first occurrence of a named entity with their respective Wikipedia page. These links are denoted as `<a>` tags with the class `extiw` in HTML. From the Wikipedia page we extract the unambiguous QID of the entity, that uniquely identifies the item in Wikidata across all languages.

The next step in the pipeline involves the classification of these entity QID's. This is achieved through a SPARQL query against a local instance of the Wikidata dataset, based on the simplified qEndpoint (Willerval et al., 2022). Entries classes in Wikidata are organised as a graph, where each instance belongs to one or more classes and each class has one or more superclasses. The query seeks to traverse the superclasses of a Wikidata instance until one of several predetermined base types is encountered, or until a defined depth limit is reached. This mechanism allows us to categorise and detect only entities of the types we are interested in.

Following the classification of entities, the system then maps all aliases of the entity that are found on Wikidata to the QID of the entity type. These aliases serve as additional textual representations of the entities and are crucial for detecting all possible mentions. These steps of resolving the type of entity based on a Wikipedia link are illustrated in Figure 2.

The final stage of the data collection pipeline involves propagating the entity links throughout the article, using the knowledge base generated by the Link resolver. This stage is illustrated with an example in Figure 3. The system scans through all n-grams of the article text and creates offset-based

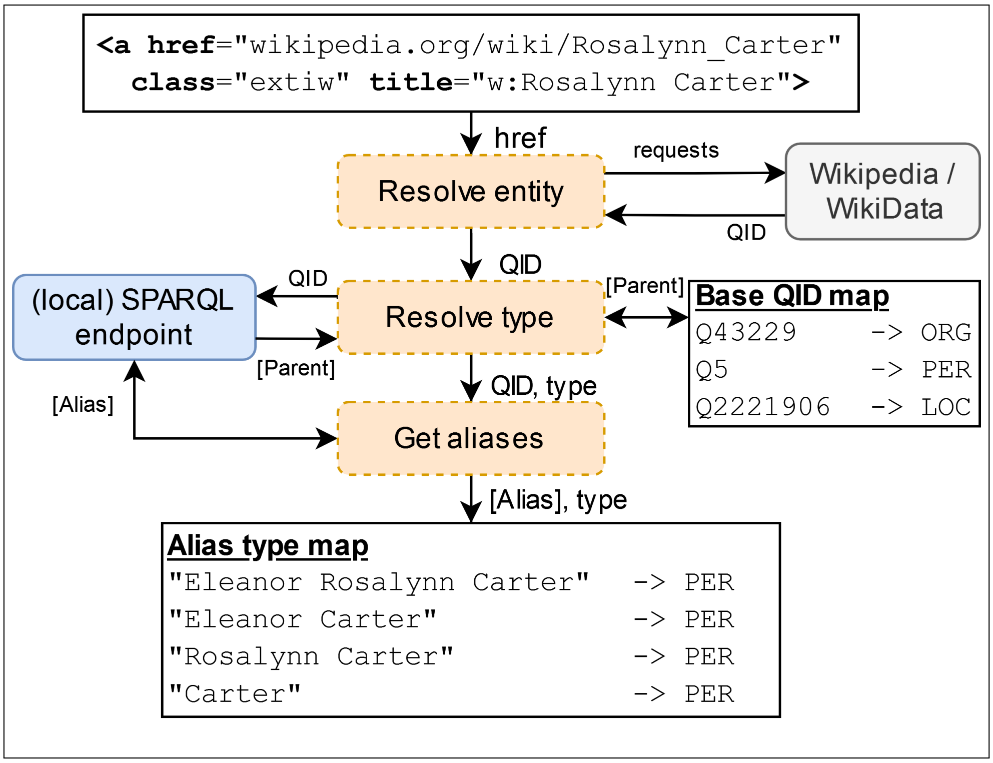


Figure 2: Example of how the link resolving pipeline builds or extends a mapping from aliases to entity types, in this case based on a link to a First Lady of the United States. Once the unambiguous QID is found, the resolver will iteratively go through the levels of parent entity classes until a parent is found that is present in the base QID mapping. In this case Q5 (human) is found in the parent QIDs and so the aliases of the lady are mapped to “PER”.

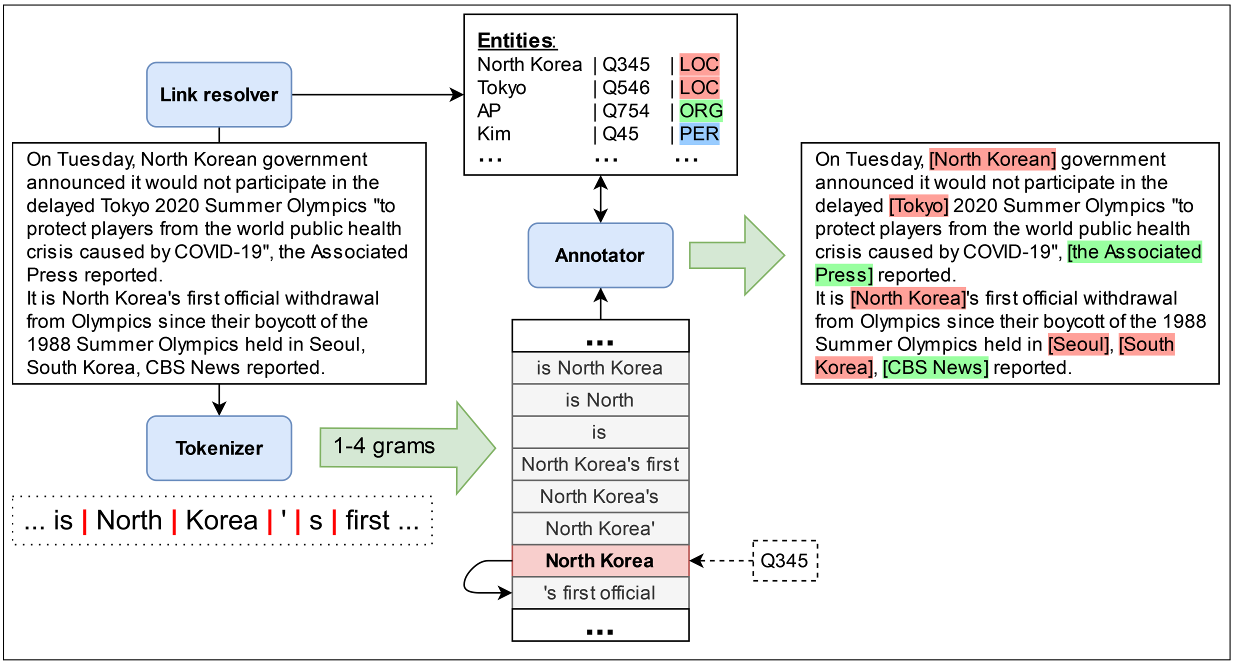
annotations for each combination of n-grams that matches one of the recognised aliases. This process aims to ensure that all potential mentions of the entities are captured and annotated with the correct type and QID.The combination of the article content stripped of HTML tags and the list of annotations is represented with the Article class.

### Post-processing

To render the scraped data suitable for evaluation, the articles must be segmented into sentences and annotated with Inside-Outside-Beginning (IOB) tags. Sentence termination is identified using the multilingual spaCy sentence model (Honnibal and Montani, 2017). To optimise the dataset’s size and enhance its usability, consecutive newline characters are compacted into a single white-space.

Subsequently, all sentences devoid of any named entity are eliminated. The remaining data is validated through a multilingual language model, specifically fine-tuned for Named Entity Recognition (NER). For this version of the dataset, we used distilbert-base-multilingual-cased-ner-hrl (Adelani, 2024). This NER finetuned version of distilbert has been trained on news data for several high resource languages. If the system-generated tags coincide with the model, the sentence is retained. By contrast, sentences for which our system yields fewer or different annotations than the model are discarded. Formally, for every sentence $x$ comprised of $n$ tokens $x_1, ..., x_n$, we evaluate the annotation (i.e., a named entity tag) $y_i$ produced by our method for each token $x_i$ against the one predicted by the auxiliary language model, $\hat{y}_i$. We retain the sentence if at least one annotation $y_i \neq O$ is present, and every $y_i \neq O$ possesses the same annotation as the corresponding $\hat{y}_i$. This procedure culminates in an enhanced precision of our annotations, as they are validated through this ensemble approach. These measures effectively reduce the volume of the collected data to approximately 0.4 to 2% of the initial scraped content, the percentage varying according to the language.

Each retained sentence is then tokenised, during which every token’s annotation type and Wikidata ID (QID) are encoded in IOB format. The final dataset comprises items with the following attributes: the original sentence, the sentence tokens, IOB-NER tags for each token, IOB-QID tags, the sentence language, and the complete URL of the source article. The steps of this pipeline are illustrated in Figure 4.



On Tuesday, North Korean government announced it would not participate in the delayed Tokyo 2020 Summer Olympics "to protect players from the world public health crisis caused by COVID-19", the Associated Press reported.
It is North Korea's first official withdrawal from Olympics since their boycott of the 1988 Summer Olympics held in Seoul, South Korea, CBS News reported.

On Tuesday, [North Korean] government announced it would not participate in the delayed [Tokyo] 2020 Summer Olympics "to protect players from the world public health crisis caused by COVID-19", [the Associated Press] reported.
It is [North Korea]'s first official withdrawal from Olympics since their boycott of the 1988 Summer Olympics held in [Seoul], [South Korea], [CBS News] reported.

Figure 3: Illustration of the mention detection algorithm of the automatic Annotator used in WINLL with a window size of 4.

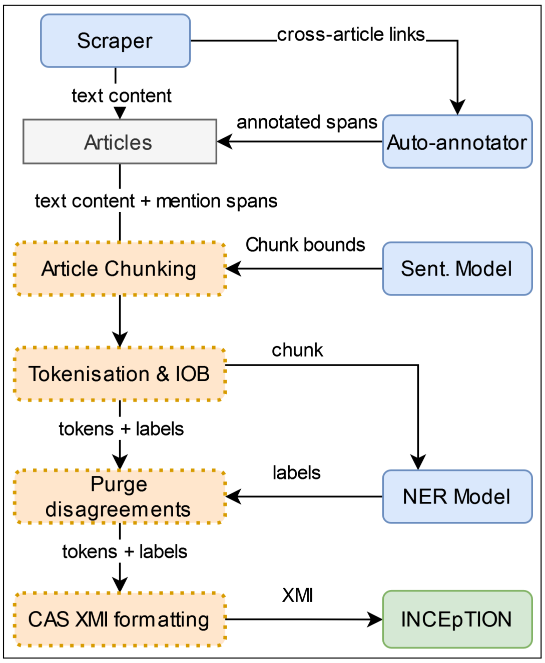


Figure 4: The post-processing pipeline used for WINLL.

### Human validation

In the final phase of data preparation, we instituted a rigorous quality control process to ensure the accuracy of our annotations. This process involved manual verification of annotated sentences by native speakers corresponding to each language in the study. These evaluators were selected from a pool of international research trainees at the European Joint Research Centre, who volunteered to help with the project. A locally hosted instance of the INCEpTION annotation tool was employed for this verification process (Klie et al., 2018). The evaluators were instructed to modify only those NER tags that were inaccurately assigned and to delete sentences with erroneous entity links entirely in order to streamline the verification process.

## 4. Results and Evaluation of DL models on the benchmark

In this section, we provide a comprehensive overview of the Multilingual Wikinews NER/L dataset, denoted as WiNNL v1. The current version of the dataset encompasses 11 predominantly European languages, namely Dutch, English, French, German, Swedish, Spanish, Portuguese, Italian, Greek, Polish, and Russian. Table 1 provides detailed statistics on the number of unique articles parsed, the count of sentences, tokens, and entities pertaining to each language.

Subsequently, we juxtaposed the outcomes of human validation against the original system-generated annotations. For each language, precision, recall, and F1 scores were computed, utilising the validated tags as the ground-truth. We use span-based metrics as opposed to token-based, as for our downstream purpose it is more useful to evaluate with metrics at a full named-entity level. This project makes use of the SemEval 2013-9.1 based evaluation library “nervaluate” (Batista and Upson, 2020). The results of this comparative anal-

| Language | Articles | Sentences | Entities | Tokens | Med. length |
|---|---|---|---|---|---|
| German (de) | 482 | 1000 | 1551 | 21222 | 119 |
| English (en) | 431 | 1000 | 1740 | 27037 | 126 |
| Dutch (nl) | 720 | 1500 | 2150 | 31584 | 112 |
| Polish (pl) | 1035 | 1606 | 2148 | 30615 | 101 |
| Italian (it) | 636 | 1155 | 1755 | 36669 | 150 |
| Spanish (es) | 319 | 615 | 1035 | 20503 | 160 |
| Portuguese (pt) | 342 | 539 | 695 | 14904 | 129 |
| French (fr) | 607 | 989 | 1661 | 34902 | 154 |
| Russian (ru) | 428 | 720 | 904 | 15371 | 104 |
| Swedish (sv) | 465 | 758 | 1024 | 139960 | 111 |
| Greek (el) | 348 | 520 | 716 | 13701 | 134 |

Table 1: Number of parsed articles, sentences, named entities, tokens and the median length in characters for the data in each language.

ysis are depicted in Table 2.

Finally, we evaluated the performance of three cutting-edge models on our dataset and compared the outcomes against other prevalent multilingual NER and NEL benchmarks. Our primary focus was to discern the impact of limiting the scope solely to sentences within the news domain on the performance of widely-used models. The findings of these named-entity level assessments are delineated in Table 3.

## 5. Conclusion and Perspectives

Based on the evaluations presented in Section 4, we conclude that the scraper pipeline has an average precision of .942, and an average recall of .917.

Although the ensemble system correctly identifies a significant portion of named entities, there are instances where it may fail to detect some entities. This shows that there is still need for a human correction step. For future iterations, we propose the use of more rigorous checking mechanisms by leveraging specific language models for each language.

Analysis of the human validation points to the system being most accurate for PER entities. A noteworthy observation by the human validators is the occasional tagging of common words that do not typically refer to named entities. This occurs in Wikinews articles when the context makes it clear what the common name refers to. An example is the term “the forest”, which could be linked to the Amazon Rainforest. To address this issue, we could implement stricter language model agreement checking. However, it is important to note that such a measure may also lead to a decrease in recall, as it might fail to identify some legitimate and linked named entities that the language model does not detect.

Therefore, the challenge lies in striking a balance between improving the precision of the NER system and maintaining, or potentially enhancing, its recall. This delicate balance will be our focus in the further development and refinement of the system.

Another consideration is the language support of Wikinews. Version 1 of WiNLL includes only 11 languages, with the main reason being the difficulty of scraping high quality tagged sentences for the other languages. For example, in the Russian language almost none of the named entities are tagged in articles. This increases the amount of articles the scraper must download, and in turn also the network overhead, to achieve a sufficiently large dataset.

In future work, the scraping pipeline could be adapted to work with other news sources, such as Voxeurop or any open sourced news website. This would involve the creation of a more elaborate interface between arbitrary HTML page sources and Wikidata objects. This will also imply to use a language model to detect the mentions, than feed the results to our pipeline and then validate the projection and linking. This will enable to see the added-value of the language model for entity mention detection and entity linking. In that scenario, another open questions arise and notably how to add new recognized entities to the Wikidata repository?

As a global conclusion, as has been shown here, to build our benchmark, we combined human annotation, Deep learning language models and Knowledge bases. The main outcome here is a benchmark that can be considered a quasi-gold standard, as it has been manually curated at the end of the process. In turn, the dataset can then be used to fine-tune a model for a specific genre (here news) and specific languages and thus create a SOTA model. As shown, the human validation is quite light, as it consists mainly in validating or invalidating the data already recognized by the KB and/or validated by the DL model.

| Language | Prec. | Rec. | F1 | LOC | ORG | PER |
|---|---|---|---|---|---|---|
| German (de) | .986 | .909 | .947 | .922 | .929 | .983 |
| English (en) | .956 | .938 | .947 | .931 | .951 | .956 |
| Dutch (nl) | .936 | .906 | .921 | .912 | .869 | .975 |
| Italian (it) | .986 | .907 | .945 | .882 | .945 | .968 |
| Spanish (es) | .944 | .944 | .943 | .884 | .941 | .975 |
| Portuguese (pt) | .965 | .936 | .950 | .958 | .927 | .965 |
| French (fr) | .879 | .870 | .875 | .813 | .826 | .944 |
| Greek (el) | .885 | .925 | .905 | .935 | .842 | .989 |

Table 2: Comparison of the accuracy of the system generated tags for each language, based on the human-validated samples. $F_1$ scores for each specific tag are given on the right. All metrics calculated on named-entity level. Mean $F_1$=.927.

| Dataset | Model | de | en | nl | pl | it | es | pt | fr | ru | sv | el |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WikiANN | XLM-Roberta | .354 | .373 | .325 | .272 | .275 | .273 | .317 | .345 | .048 | .223 | .024 |
| | wikineural | .715 | .554 | .716 | .758 | .696 | .671 | .628 | .688 | .361 | .733 | .661 |
| | distilbert | .657 | .521 | .653 | .694 | .584 | .589 | .549 | .542 | .331 | .715 | .505 |
| UNER | XLM-Roberta | .496 | .497 | - | - | - | - | .404 | - | .081 | .418 | - |
| | wikineural | .771 | .808 | - | - | - | - | .838 | - | .687 | .839 | - |
| | distilbert | .816 | .809 | - | - | - | - | .847 | - | .714 | .865 | - |
| | WiNNL-model | .762 | .772 | - | - | - | - | .804 | - | .670 | .821 | - |
| WiNNL | XLM-Roberta | .584 | .561 | .561 | .297* | .388 | .449 | .415 | .409 | .066* | .543* | .071 |
| | wikineural | .835 | .827 | .843 | .759* | .753 | .884 | .875 | .835 | .785* | .876* | .724 |
| | distilbert | .828 | .810 | .851 | .756* | .787 | .878 | .875 | .839 | .816* | .883* | .712 |

Table 3: Named-entity span level $F_1$ scores of wikineural, distilbert-cased and XLM-Roberta on the WikiANN, UNER and WiNNL NER benchmarks. Scores indicated with $\star$ are not validated by humans. We also evaluated the UNER benchmark with a multilingual distilbert model finetuned on our human validated WiNNL dataset (WiNNL-model).

In the next version of the benchmark-building system, for the remaining languages, as we don't have enough human annotated data sources, we will rely on a SOTA language model to first annotate mentions and then the propagation of mentions will be ensured by the KB mentions feature. That will open other questions, the way around, on the added-value of DL for updating KB.

## 6. Bibliographical References

1998. Message understanding conference-7: A research retrospective. In *Proceedings of a Workshop held at Fairfax Virginia USA: George Mason University Press*.

2016–2023. entity-fishing. `https://github.com/kermitt2/entity-fishing`.

David Ifeoluwa Adelani. 2024. distilbert-base-multilingual-cased-ner-hrl.

Kabir Ahuja, Rishav Hada, Millicent A. Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai. *ArXiv*, abs/2303.12528.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Tom Ayoola, Joseph Fisher, and Andrea Pierleoni. 2022a. Improving entity disambiguation by reasoning over a knowledge base. In *NAACL*.

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, and Christos Christodoulopoulos. 2022b. Refined: An efficient zero-shot-capable approach to end-to-end entity linking. *ArXiv*, abs/2207.04108.

David Batista and Matthew Antony Upson. 2020. nervaluate.

David S. Batista. 2024. Ner datasets.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Germeval-2014 named entity recognition shared task–guidelines and dataset. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*.

Priyanka Bose, Sriram Srinivasan, William C. Sleeman, Jatinder R. Palta, Rishabh Kapoor, and Preetam Ghosh. 2021. A survey on recent named entity recognition and relationship extraction techniques on clinical texts. *Applied Sciences*.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Nicola De Cao, Ledell Yu Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Sophie Chesney, Guillaume Jacquet, Ralf Steinberger, and Jakub Piskorski. 2017. Multi-word entity classification in a highly multilingual environment. In *MWE@EACL*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and M. Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *North American Chapter of the Association for Computational Linguistics*.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. 2021. Xlm-e: Cross-lingual language model pre-training via electra. In *Annual Meeting of the Association for Computational Linguistics*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. *CoRR*, abs/2105.07464.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A primer on pretrained multilingual language models. *ArXiv*, abs/2107.00676.

Henrique dos Santos Cardoso and José Gabriel Pereira Lopes Silva. 2018. Pereria: A portuguese named entity recognition dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC'18). European Language Resources Association (ELRA)*.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2023. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56:1 – 47.

Maud Ehrmann, Guillaume Jacquet, and Ralf Steinberger. 2016. Jrc-names: Multilingual entity name variants and titles as linked data. *Semantic Web*, 8:283–295.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named entity disambiguation for noisy text. In *Conference on Computational Natural Language Learning*.

Association for Computational Linguistics. Accessed: April 17, 2023. Named entity recognition (ner) bibliography. `https://www.aclweb.org/aclwiki/Named_entity_recognition_(NER)_bibliography`.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. Interpretable multi-dataset evaluation for named entity recognition. *ArXiv*, abs/2011.06854.

Jingru Gan, Jinchang Luo, Haiwei Wang, Shuhui Wang, W. He, and Qingming Huang. 2021. Multimodal entity linking: A new dataset and a baseline. *Proceedings of the 29th ACM International Conference on Multimedia*.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 466–471. Association for Computational Linguistics.

Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. 2014. Discovering emerging entities with ambiguous names. *Proceedings of the 23rd international conference on World wide web*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Conference on Empirical Methods in Natural Language Processing*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *ArXiv*, abs/2003.11080.

Guillaume Jacquet, Maud Ehrmann, and Ralf Steinberger. 2014. Clustering of multi-word named entity variants: Multilingual evaluation. In *International Conference on Language Resources and Evaluation*.

Guillaume Jacquet, Maud Ehrmann, Ralf Steinberger, and Jaakko J. Väyrynen. 2016. Cross-lingual linking of multi-word entities and their corresponding acronyms. In *International Conference on Language Resources and Evaluation*.

Guillaume Jacquet, Jakub Piskorski, and Sophie Chesney. 2019a. Out-of-context fine-grained multi-word entity classification: exploring token, character n-gram and nn-based models for multilingual entity classification. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*.

Guillaume Jacquet, Jakub Piskorski, Hristo Tanev, and Ralf Steinberger. 2019b. JRC TMA-CC: Slavic named entity recognition and linking. participation in the BSNLP-2019 shared task. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 100–104, Florence, Italy. Association for Computational Linguistics.

Piskorski Jakub. 2007. Express - extraction pattern recognition engine and specification suite.

Renato Stoffalette João. 2020. On the temporality of priors in entity linking. *Advances in Information Retrieval*, 12036:375 – 382.

Poonam Kashtriya, Pardeep Singh, and Parul Bansal. 2023. A review on clinical named entity recognition. *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)*, pages 1–6.

Hossein Keshavarz, Zografoula Vagena, Pigi Kouki, Ilias Fountalis, Mehdi Mabrouki, Aziz Belaweid, and Nikolaos Vasiloglou. 2022. Named entity recognition in long documents: An end-to-end case study in the legal domain. *2022 IEEE International Conference on Big Data (Big Data)*, pages 2024–2033.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *proceedings of the 27th international conference on computational linguistics: system demonstrations*, pages 5–9.

Veysel Kocaman and David Talby. 2020. Biomedical named entity recognition at scale. *CoRR*, abs/2011.06315.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vázquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Dong-Hong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, P. Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber Ahmad Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Joo young Choi, Karin M. Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, K. E. Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, C Ata, Tolga Can, Anabel Usie, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzábal, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7:S2 – S2.

David Kubeša and Milan Straka. 2023a. Damuel: A large multilingual dataset for entity linking.

David Kubeša and Milan Straka. 2023b. Damuel: A large multilingual dataset for entity linking. *ArXiv*, abs/2306.09288.

Dilek Küçük, Guillaume Jacquet, and Ralf Steinberger. 2014. Named entity recognition on turkish tweets. In *International Conference on Language Resources and Evaluation*.

Dilek Küçük and Ralf Steinberger. 2014. Experiments to improve named entity recognition on turkish tweets. *ArXiv*, abs/1410.8668.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Pei-Chi Lo and Ee-Peng Lim. 2020. Interactive entity linking using entity-word representations. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. Multiconer: A large-scale multilingual dataset for complex named entity recognition. In *International Conference on Computational Linguistics*.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. Semeval-2022 task 11: Multilingual complex named entity recognition (multiconer). In *International Workshop on Semantic Evaluation*.

Katja Markert and Malvina Nissim. 2007. Semeval-2007 task 08: Metonymy resolution at semeval-2007. In *International Workshop on Semantic Evaluation*.

Cedric Moller, Jens Lehmann, and Ricardo Usbeck. 2021. Survey on english entity linking on wikidata. *ArXiv*, abs/2112.01989.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Joel Nothman, James R Curran, and Tara Murphy. 2008. Transforming wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132. Citeseer.

Damien Nouvel. 2024. Corpora for named entities.

Italo Lopes Oliveira, Renato Fileto, René Speck, René Speck, Luís Paulo F. Garcia, Diego Moussallem, Diego Moussallem, Jens Lehmann, and Jens Lehmann. 2021. Towards holistic entity linking: Survey and directions. *Inf. Syst.*, 95:101624.

Yasumasa Onoe and Greg Durrett. 2019. Fine-grained entity typing for domain independent entity linking. *ArXiv*, abs/1909.05780.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017a. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017b. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of ACL 2017*, pages 1946–1958.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, Heng Ji, Diana Maynard, Yuan Yao, Zhijing Zhao, Yaqing Guan, et al. 2020. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022a. Beyond english-centric bitexts for better multilingual language representation learning. In *Annual Meeting of the Association for Computational Linguistics*.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022b. Beyond english-centric bitexts for better multilingual language representation learning. In *Annual Meeting of the Association for Computational Linguistics*.

Jakub Piskorski. 2008. Corleone - core linguistic entity online extraction. Technical Report EUR 23393 EN, OPOCE, Luxembourg (Luxembourg). JRC45952.

Jakub Piskorski, Laska Laskova, Michał Marcińczuk, Lidia Pivovarova, Pavel Přibáň, Josef Steinberger, and Roman Yangarber. 2019. The second cross-lingual challenge on recognition, normalization, classification, and linking of named entities across Slavic languages. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 63–74, Florence, Italy. Association for Computational Linguistics.

Jakub Piskorski, Karol Wieloch, and Marcin Sydow. 2009. On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. *Information Retrieval*, 12(3):275–299.

Mikhail Plekhanov, Nora Kassner, Kashyap Popat, Louis Martin, Simone Merello, Borislav M. Kozlovskii, Frédéric A. Dreyer, and Nicola Cancedda. 2023. Multilingual end to end entity linking. *ArXiv*, abs/2306.08896.

Marco Polignano, Marco Degemmis, and Giovanni Semeraro. 2021. Comparing transformer-based ner approaches for analysing textual medical diagnoses. In *Conference and Labs of the Evaluation Forum*.

Bruno Pouliquen. 2008. Similarity of names across scripts: Edit distance using learned costs of n-grams. In *GoTAL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *ArXiv*, abs/2302.06476.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Steinberger Ralf, Guillaume Jacquet, and Della Rocca Leonida. 2014. Creation and use of multilingual named entity variant dictionaries.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

L. Ratinov and D. Roth. 2011. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, pages 147–155.

L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Annual Meeting of the Association for Computational Linguistics*.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2021a. mluke: The power of entity representations in multilingual pretrained language models. *CoRR*, abs/2110.08151.

Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2021b. mluke: The power of entity representations in multilingual pretrained language models. In *Annual Meeting of the Association for Computational Linguistics*.

Juan Diego Rodriguez. 2024. Entity recognition datasets.

Henry Rosales-Méndez, Aidan Hogan, and Bárbara Poblete. 2018. Voxel: A benchmark dataset for multilingual entity linking. In *International Workshop on the Semantic Web*.

Arya Roy. 2021. Recent trends in named entity recognition (NER). *CoRR*, abs/2101.11420.

Sebastian Ruder. 2022. The State of Multilingual AI. `http://ruder.io/state-of-multilingual-ai/`.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021a. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan A. Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021b. Xtreme-r: Towards more challenging and nuanced multilingual evaluation. *ArXiv*, abs/2104.07412.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Conference on Computational Natural Language Learning*.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *International Conference on Language Resources and Evaluation*.

Ozge Sevgili, Artem Shelmanov, Mikhail V. Arkhipov, Alexander Panchenko, and Christian

Biemann. 2020. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13:527–570.

Akshaya Kesarimangalam Srinivasan and Sowmya Vajjala. 2023. A multilingual evaluation of ner robustness to adversarial inputs. *ArXiv*, abs/2305.18933.

Ralf Steinberger and Bruno Pouliquen. 2007. Cross-lingual named entity recognition. *Lingvisticæ Investigationes*, 30(1):135–162.

Ralf Steinberger, Bruno Pouliquen, Mijail Alexandrov Kabadjov, Jenya Belyaeva, and Erik Van der Goot. 2011. Jrc-names: A freely available, highly multilingual named entity resource. *ArXiv*, abs/1309.6162.

Stephanie Strassel and Jennifer Tracey. 2016. LORELEI language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3273–3280, Portorož, Slovenia. European Language Resources Association (ELRA).

Zeqi Tan, Shen Huang, Zixia Jia, Jiong Cai, Yinghui Li, Weiming Lu, Yueting Zhuang, Kewei Tu, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. Damo-nlp at semeval-2023 task 2: A unified retrieval-augmented system for multilingual named entity recognition.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021a. Named entity recognition for entity linking: What works and what's next. In *Conference on Empirical Methods in Natural Language Processing*.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021b. Named Entity Recognition for Entity Linking: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2584–2596, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021c. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021d. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812, Seattle, United States. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003a. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003b. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Robbert van der Goot, Floris-Jan van der Klis, and Gertjan van Noord. 2014. The gmbc corpus: A web-based corpus of dutch text with rich annotation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM.

C. J. van Rijsbergen. 1979. *Information Retrieval*. London: Butterworths.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Denny Vrandeić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57:78–85.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and

Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *ArXiv*, abs/2304.10428.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *ArXiv*, abs/2010.05006.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yue Ting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022a. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. *ArXiv*, abs/2203.00545.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yue Ting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022b. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. *ArXiv*, abs/2203.00545.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, C. Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *bioRxiv*.

Xuwu Wang, Junfeng Tian, Min Gui, Zhixu Li, Rui Wang, Ming Yan, Lihan Chen, and Yanghua Xiao. 2022c. Wikidiverse: A multimodal entity linking dataset with diversified contextual topics and entity types. In *Annual Meeting of the Association for Computational Linguistics*.

Antoine Willerval, Dennis Diefenbach, and Pierre Maret. 2022. Easily setting up a local wikidata sparql endpoint using the qendpoint.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Bo Xu, Shizhou Huang, Chaofeng Sha, and Hongya Wang. 2022. Maf: A general matching and alignment framework for multimodal named entity recognition. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.

Roman Yangarber, Jakub Piskorski, Anna Dmitrieva, Michal Marcinczuk, Pavel Přibáň, Piotr Rybak, and Josef Steinberger. 2023. Slav-ner: the 4th cross-lingual challenge on recognition, normalization, classification, and linking of named entities across slavic languages. *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*.

Hanming Zhai, Xiaojun Lv, Zhiwen Hou, Xin Tong, and Fanliang Bu. 2023. Mlnet: a multi-level multimodal named entity recognition architecture. *Frontiers in Neurorobotics*, 17.

Zhenru Zhang, Chuanqi Tan, Songfang Huang, and Fei Huang. 2023a. Veco 2.0: Cross-lingual language model pre-training with multi-granularity contrastive learning. *ArXiv*, abs/2304.08205.

Zhenru Zhang, Chuanqi Tan, Songfang Huang, and Fei Huang. 2023b. Veco 2.0: Cross-lingual language model pre-training with multi-granularity contrastive learning. *ArXiv*, abs/2304.08205.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480.

## 7. Language Resource References

WiNNL code and link to the dataset can be found here: `https://gitlab.jrc.ec.europa.eu/jrc-projects/emm/emm-wikinews-scraper`.

# Deductive Verification of LLM generated SPARQL queries

**Alexandre Rademaker**[1,2], **Guilherme Lima**[1], **Sandro R. Fiorini**[1], **Viviane T. da Silva**[1]
IBM Research[1] and School of Applied Mathematics FGV[2]
Rio de Janeiro, Brazil
{alexrad,vivianet}@br.ibm.com, {guilherme.lima,srfiorini}@ibm.com

## Abstract

Considering the increasing applications of Large Language Models (LLMs) to many natural language tasks, this paper presents preliminary findings on developing a verification component for detecting hallucinations of an LLM that produces SPARQL queries from natural language questions. We suggest a logic-based deductive verification of the generated SPARQL query by checking if the original NL question's deep semantic representation entails the SPARQL's semantic representation.

**Keywords:** SPARQL, LLM, hallucination, HOL, higher-order logic, Z3, theorem prover

## 1. Introduction

This paper reports the preliminary results of developing a verification component of a chat-like interface for chemists interested in retrieving information about chemical compounds from a knowledge graph (KG) like Wikidata or PubChem.[1]

In Section 2, we briefly presented our pipeline and how questions formulated in English as the Example (1-a) are translated to the SPARQL queries as the one presented in Listing 1 using an LLM like GPT-4 (OpenAI, 2023) or LLAMA-2-70b [2]. However, our focus in this paper is not on the SPARQL generation nor the correctness of the answer provided by the Knowledge Graph for the SPARQL query. Instead, we focus on validating the SPARQL query obtained from the LLM. In other words, if the SPARQL 'makes sense' given the original question formulated in English. Preventing hallucinations has been a hot topic in the literature recently (Wang et al., 2023; Cao, 2023; Ling et al., 2023; Dhuliawala et al., 2023). In our application to assist chemists asking for properties about chemical compounds in a chat, a user relying only on the LLM's final answer can be misguided if he can't read the intermediary SPARQL query produced to retrieve the facts from the Knowledge Graph.

(1) a. What is the mass of benzene?
b. Give me the benzene's toxicity.
c. What chemical compounds have less than 0.07 g/kg of solubility?
d. What is the electric dipole moment of the allyl alcohol?
e. What is the mass of the compound with InChIKey UHOVQNZJYSORNB-UHFFFAOYSA-N?

In Example (1), we present some variants of questions about chemical compounds and their properties that a chemist can submit to our chat interface. We are restricting our focus to factoid questions, answerable by simple SPARQL queries involving only simple triple patterns; there are many challenges to dealing with such questions. The first and most obvious is that the syntactic structure can vary greatly. The second challenge is that properties are hardly mentioned by their labels in KGs. For instance, in our context, toxicity should be interpreted as the substance's median lethal dose (LD50),[3] but we could not be sure in a more general context. Third, the property values are usually measured in complex units. Solubility is measured in 'grams per kilogram,' and LD50 is expressed as the mass of substance administered per unit mass of the test subject, typically as milligrams of a substance per kilogram of body mass. Moreover, Lethal dosage often varies depending on the method of administration; many substances are less toxic when administered orally than when intravenously administered. To sum up, the toxicity of a compound is usually expressed as a complex unit like 'LD50 Rat oral 3530 mg/kg'. Fourth, chemicals can be identified in various ways. For example, 'allyl alcohol' has 91 synonyms ranging from IUPAC names to identifiers in different standards proposed by the scientific communities.[4]

Figure 1 presents two logical formulas expressed in higher-order logic, particularly in ULKB Logic (Lima et al., 2023). ULKB is an open-source framework written in Python for logical reasoning over knowledge graphs. The first formula, Formula 1, is the logical semantics of Example (1-a)

---

[1] We will restrict our examples to Wikidata KG, but the techniques can be used with any KG.

[2] https://ai.meta.com/llama/

[3] The reader doesn't have to understand the chemical terms mentioned in this paragraph; we are only exemplifying the particularities on processing English questions on a technical domain.

[4] https://pubchem.ncbi.nlm.nih.gov/compound/Allyl-alcohol

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 45–52
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?v WHERE {
    wd:Q2270 wdt:P2067 ?v .
}
```

Listing 1: SPARQL query

obtained with MRS Logic (Rademaker et al., 2023), a library to translate English sentences into logical formulas built on top of ULKB and 'deep' linguistic specialized tools. The second formula, Formula 2, is the translation of the SPARQL to the ULKB logic, a functionality also presented in the ULKB Library. Provided a proper map between the predicates «`wdt:P2067`» [5] and _mass_n_off and also between «`wdt:Q2270`» and _benzene_n_1, we can conclude that Formula 1 entails Formula 2, certifying that the query is indeed related to the English questions. Section 3 presents the tools we rely on, and Section 4 elaborates on our process for validating the SPARQL query.

To summarize, our contribution is a deductive approach to prevent LLM hallucinations in translating English questions to SPARQL queries. The validation process reported here is a component of a chat-based assistant for chemists interested in obtaining information about chemical compounds from a large and complex KG without having to construct a SPARQL query manually. In other words, we are focusing not on ordinary questions presented in datasets like (Trivedi et al., 2017) but on questions made by chemists about chemical compounds, a deep technical domain with plenty of technical terms. On the other hand, our method is not restricted to any particular Knowledge Graph Question Answering (KGQA) approach; it can be adapted for different domains and systems such as (Zhou et al., 2021). Before describing the SPARQL validation method based on the semantic parsing of the NL utterances, we present in Section 2 the overview of our text to SPARQL conversion pipeline based on LLM. Our pipeline is one of the components of ChemChat, a conversational expert assistant in material science (Erdmann et al., 2024).

## 2. The architecture of our system

Using prompt engineering, we used 'few-short learning' (Brown et al., 2020). We implemented an LLM pipeline to translate English questions to SPARQL queries. Figure 2 presented the pipeline and the workflow to process the Example (1-a). To create the prompts, with the help of some chemists, we collected examples of English questions and manually annotated the technical chemical terms on them and their related SPARQL queries.

Let us first describe the step-by-step process of constructing a SPARQL from an English sentence. The pipeline (blue boxes) starts by sending the input question through an LLM with examples to suggest our goal of extracting a table with the relevant terms (usually adjective and noun phrases) and their classification as either 'property' or 'entity.' We construct 'Prompt 1' from the set of examples we have collected. Next, we parse the table received from the LLM to disambiguate the terms (that is, grounding them to Wikidata identifiers) using a full-text search on a database populated with relevant Wikidata items and properties with their labels. Utilizing the fact we are building a specialized interface for chemists, this database (an Elastic Search [6] index) is constructed from Wikidata's chemical taxonomy using items like 'type of chemical entity' (Q113145171) and the 'WikiProject Chemistry' (Q8487234) as seeds and exploring their descendants and related concepts and properties. We use the query results to construct a disambiguation table mapping each term obtained from the completion of Prompt 1 to its best-matching Wikidata identifier. Once we have the Wikidata identifiers, we can produce the second prompt (Prompt 2), now using examples of pairs of English questions and SPARQL queries together with their identifiers. We submit Prompt 2 to a second LLM to generate the final SPARQL query. Sometimes, the LLM fails to produce a valid SPARQL query; to deal with that, we repeat the process a few times and use the most frequent answer as the expected solution.

As stated in the introduction, this article does not aim to describe the LLM pipeline completely nor discuss its performance, precision, or recall. These topics will be the subject of another article. Our focus here is on one specific component: the method to validate the SPARQL obtained for each question in English, presented in the salmon boxes of Figure 2. First, the same English question submitted by the chemist is parsed with a computational grammar for English, and a semantic representation of the sentence is produced. This semantic representation is translated to a sentence (Sentence A) in higher-order logic using the MRS Logic library.

---

[5] We are adopting the notation «. . .» as a simplified way to reference the fully qualified URI of an item from the Wikidata data schema.

[6] https://www.elastic.co

$$\exists\, x_{13},\ \_benzene\_n\_1\ x_{13}\ \wedge\ (\exists\, x_8,\ \_mass\_n\_of\ x_8\ x_{13}\ \wedge\ (\exists\, x_3,\ thing\ x_3\ \wedge\ (\exists\, e_2,\ \_be\_v\_id\ e_2\ x_3\ x_8))) \quad (1)$$

$$\exists\, value,\ \text{«wdt:P2067»}\ \text{«wdt:Q2270»}\ value \quad (2)$$

Figure 1: The formal semantics of the sentence in Example (1-a) and the translation to a logical formula of the SPARQL query from Listing 1. «wdt:P2067» is a binary predicate (associated with an RDF edge), and «wdt:Q2270» is a constant (associated with an RDF node). In ULKB, the «...» indicates that the function or predicate is associated with a URI. Applying a function or predicate to its arguments is usually written as $f(x, y)$ in many formal languages. Still, in ULKB, we write $f\ x\ y$, omitting the parenthesis and commas.

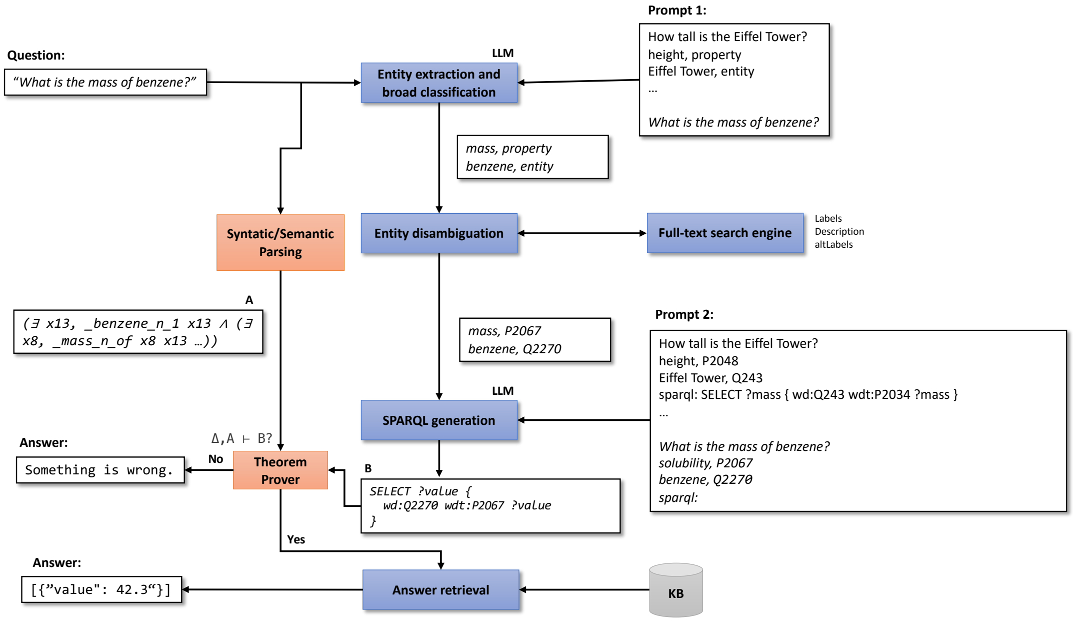


Figure 2: In our pipeline, the natural language question goes through one initial LLM prompted to extract relevant question terms. These terms are disambiguated using a full-text search over a subset of relevant Wikidata items. The search result helps a second LLM to generate SPARQL queries with the correct Wikidata identifiers.

The SPARQL query obtained from the LLM is also translated to Sentence B in higher-order logic using the ULKB library SPARQL to HOL translation. Using the ULKB wrapper to theorem provers, we check if sentence A (together with some additional axioms in the set of formulas Δ) entails sentence B. The following section describes the natural language processing tools we used for parsing and constructing logical formulas.

## 3. Background

MRS Logic (Rademaker et al., 2023) is a Python library to convert NL utterances into higher-order logic formulas. It is built on top of many other components that we describe below.

The main component of MRS Logic (Rademaker et al., 2023) is the English Resource Grammar (ERG) (Flickinger, 2000; Flickinger et al., 2000; Copestake and Flickinger, 2000). The English Resource Grammar is a broad-coverage, linguistically precise, general-purpose computational grammar. It is implemented in the theoretical framework of Head-driven Phrase Structure Grammar (Pollard and Sag, 1994) where both morphosyntactic and semantic properties of English are expressed in a declarative format. Combined with specialized processing tools, it can map running English text to highly normalized representations of meaning called Minimal Recursion Semantics (MRS) (Copestake et al., 2005). ERG is developed as part of the international Deep Linguistic Processing with HPSG Initiative (DELPH-IN). It can be processed by several parsing and realization systems, including ACE (Crysmann and Packard, 2012).[7]. MRS Logic uses PyDelphin (Goodman, 2019) library to communicate with ACE.

MRS structures directly interface with syntax and can be underspecified in various aspects, such as word senses and quantifier scopes. This underspecification enables a single MRS to encompass multiple interpretations. Figure 3 shows one among the nine possible MRSs for Example (1-a). It consists of a multiset of relations called elementary predications (EPs). An EP usually corresponds to a single lexeme but can represent grammatical features (e.g., `thing` and `udef_q`, called abstract predicates). Each EP has a label or handle, a predicate symbol, which, in the case of lexical predicates, encodes information about lemma, part-of-speech, and coarse-grained sense distinctions, and a list of numbered arguments: `ARG0`, `ARG1`, etc. The value of an argument can be either a scopal variable (a hole representing the places where alternative labels could fill) or a non-scopal variable (events, states, or entities). The `ARG0` argument has the EP's distinguished variable. This variable denotes an event, state, or referential or abstract entity ($e_i$ or $x_i$, respectively). Each non-quantifier EP has its unique distinguished variable. Finally, an MRS has a set of handle constraints describing how the EPs' scopal arguments can be nested with EP labels. A constraint $h_i =_q h_j$ denotes equality modulo quantifier insertion. In addition to the indirect linking through handle constraints, EPs are directly linked by sharing the same variable as argument values, capturing the predicate-argument structure of the sentence. Finally, MRS also records properties on variables indicating morpho-syntactic marks of person, number, tense, aspect, etc. The topmost relation in Figure 3 is _be_v_id, which has the non-empty arguments $x_5$ and $x_9$. The $x_5$ is the distinguished variable of the relation `thing`. A handle constraint equates the sentential variable $h_2$ with $h_1$, the top handle. The rest of the EPs can be explained similarly. Note that $h_7$ does not appear in the handle constraints, suggesting that we have more than one possible way to equate this hole with the available labels.

For solving the underspecification of the scopes of quantifiers in an MRS, MRS Logic employs the Utool scope resolution Java Library (Koller and Thater, 2005, 2006, 2010). From a single MRS, Utool can produce many possible scope trees, fully scoped resolved trees, reflecting the different possible order of quantifiers in the final logical formula. For instance, the MRS of Figure 3 has an alternative reading for the order of quantifiers in Formula 1, e.g., $\exists\, x_8, \ldots \exists\, x_{13}, \ldots$, but in this case, the two are semantically equivalent.

The scope trees are not yet a concrete logical expression in any logical language. The literature has many proposals for representing NL utterance semantics. One of the most fundamental issues about which logic to use is whether one assumes any structure on the individuals. Other issues are the complexity, decidability, and tools for reasoning in a particular logic. Type theories are widely used in formal theories of the semantics of natural languages (Chatzikyriakidis and Luo, 2020; Ranta, 1994; Winter, 2016). A subset of that, simple type theory, also called higher-order logic (HOL), is a natural extension of first-order logic, which is elegant, highly expressive, and practical (Farmer, 2008).

The ULKB Logic (Lima et al., 2023) implements HOL in Python for logical reasoning over knowledge graphs. The formulas presented in Figure 1 are HOL formulas encoded in ULKB Logic. ULKB provides an interactive theorem prover-like environment that can interact with external provers such as E prover (Schulz et al., 2019) and Z3 SMT solver (de Moura and Björner, 2008). In (Lima et al., 2023), the authors present the logical foundations and implementation of ULKB Logic and its interfaces for fetching statements from knowledge graphs and calling external provers. These interfaces are vital for achieving ULKB Logic's primary goal, which is twofold: (i) provide a common language and interactive theorem prover-like environment for representing commonsense and linguistic knowledge, and (ii) facilitate the use of state-of-the-art computational logic tools to reason over the knowledge available in knowledge graphs. For (ii), ULKB uses SPARQL (W3C SPARQL Working Group, 2013), the standard query language of the Semantic Web, and allows users to use logic formulas as queries, parse SPARQL queries into logic formulas and submit SPARQL queries to KG endpoints.

Finally, consider the possible senses for the word 'mass.' ERG only distinguishes senses that are morphosyntactically marked. Since further sense distinctions could never be disambiguated based on grammatical structure alone, the ERG predicate symbol `_mass_n_of` intended to be an underspecified representation of all the specific word senses. For instance, Wordnet 3.1 (Miller, 1995) contains eleven possible nominal senses for this word. We use UKB (Agirre and Soroa, 2009) for Word Sense Disambiguation (WSD), the ERG predicates. UKB performs graph-based disambiguation using any pre-existing knowledge base, provided the structure of the graph (nodes and edges) and the dictionary of words or multi-word expressions associated with each node.

To summarize, MRS Logic takes an NL utterance and calls ACE to obtain all possible MRSs. Given an MRS, it is transformed into a scope tree using Utool and passed to UKB to disambiguate the ERG predicates, linking them to nodes in a reference

---

[7] `http://sweaglesw.org/linguistics/ace/`

$$\begin{array}{l}
\langle\, h_1, e_3\{\text{SF } \mathit{ques}, \text{TENSE } \mathit{pres}, \text{MOOD } \mathit{indicative}, \text{PROG -}, \text{PERF -}\}, \\
\quad \left| \begin{array}{l}
h_4\text{:thing}\langle 0{:}4\rangle(\text{ARG0 } x_5\{\text{PERS } 3, \text{NUM } \mathit{sg}\}), \\
h_6\text{:which\_q}\langle 0{:}4\rangle(\text{ARG0 } x_5, \text{ RSTR } h_8, \text{ BODY } h_7), \\
h_2\text{:\_be\_v\_id}\langle 5{:}7\rangle(\text{ARG0 } e_3, \text{ ARG1 } x_5, \text{ ARG2 } x_9\{\text{PERS } 3, \text{NUM } \mathit{sg}\}), \\
h_{10}\text{:\_the\_q}\langle 8{:}11\rangle(\text{ARG0 } x_9, \text{ RSTR } h_{12}, \text{ BODY } h_{11}), \\
h_{13}\text{:\_mass\_n\_of}\langle 12{:}16\rangle(\text{ARG0 } x_9, \text{ ARG1 } x_{14}\{\text{PERS } 3, \text{NUM } \mathit{sg}\}), \\
h_{15}\text{:udef\_q}\langle 20{:}28\rangle(\text{ARG0 } x_{14}, \text{ RSTR } h_{17}, \text{ BODY } h_{16}), \\
h_{18}\text{:\_benzene\_n\_1}\langle 20{:}27\rangle(\text{ARG0 } x_{14})
\end{array} \right| \\
\quad \{\, h_1 =_q h_2,\ h_8 =_q h_4,\ h_{12} =_q h_{13},\ h_{17} =_q h_{18} \,\}\,\rangle
\end{array}$$

Figure 3: The first MRS return by ERG for the Example (1-a).

KG.[8] Finally, the MRS is translated into ULKB formulas. MRS Logic integrates all the technologies described above. At the high level, the translation starts from the topmost node of the scope tree, the handle in the higher position, usually a quantifier. The translation is fully explained in (Rademaker et al., 2023).

## 4. Validating an SPARQL

Our main problem can be defined as the logical entailment test in Equation 3.

$$\Delta, T(\alpha) \models G(\alpha) \tag{3}$$

where $\alpha$ is an English question, $T(\alpha)$ is one of the possible higher-order logic formulas obtained from the English question $\alpha$ by the MRS Logic (Section 3), e.g., Formula 1, and $G(\alpha)$ is a first-order logic formula obtained from the translation of the SPARQL query, in our case, produced as the translation of the same English question to SPARQL by an LLM, e.g., the Formula 2 (Section 1). Finally, $\Delta$ is a set of axioms to support the entailment. This logic theory connects the symbols from the ERG grammar presented in the MRS to those obtained from the SPARQL query, the KG identifiers.

Consider again the natural language (English) question from Example (1-a). From the MRS in Figure 3, the UKB disambiguation step, using Wordnet 3.1 as KG disambiguates, produces the mapping of $e_2$ to the sense "have the quality of being; copula, used with an adjective or a predicate noun" (Synset `02610777-v`), variable $x_8$ to "the property of a body that causes it to have a weight in a gravitational field" (Synset `05031420-n`. Synset `05031420-n` is associated with the Wikidata item `Q11423` by the 'WordNet 3.1 Synset ID' (P8814) Wikidata property. This item has 'Wikidata property' linking it to the property `P2067`). The variable $x_{13}$ is disambiguated to "a colorless liquid hydrocarbon" (Synset id `14798860-n`, associated to the Wikidata item `Q2270` by the same P8814 Wikidata property. In other words, from the disambiguation produced by UKB, we can follow the links to the Wikidata items and properties. This process allow us to properly associating the ERG predicates _benzene_n_1 to `Q2270` and _mass_n_of to `P2067`. The Wikidata item `Q2270` does not have a value for 'Wikidata property,' which means it is not an item used as a property of something. This disambiguation process will be revised to better use the Wikidata Lexeme data, which can have more flexible mappings to the items and properties of Wikidata.

From the last paragraph, we have enough information to instantiate some axioms in the ULKB theory, the $\Delta$ above. Axiom 4 tells us that something that is the argument of the ERG unary predicate _benzene_n_1 is the «wsd:benzene» item in Wikidata. Axiom 5 tells us that the ERG binary predicate _mass_n_of can be translated to the Wikidata property «wsd:mass». A Python function in ULKB can construct both axioms; these functions are actually macros in HOL. The function gets the ERG predicates and the mappings from UKB. From the mapping and the type (and arity) of the Wikidata entities (item or property), the function can instantiate the axioms from a set of templates.

$$\forall\, x, \mathit{\_benzene\_n\_1}\ x\ \ \rightarrow x = \text{«wsd:benzene»} \tag{4}$$

$$\forall\, x\ y, \mathit{\_mass\_n\_of}\ x\ y\ \ \rightarrow \exists\, v, \text{«wsd:mass»}\ y\ v \tag{5}$$

Example (1-b) would instantiate another axiom, the word 'toxicity' evokes an ERG unary predicate _toxicity_n_1 and not a binary predicate as the word 'mass.' The connection between the words 'benzene' and 'toxicity' is mediated by the abstract predicate poss (possessive) from ERG. The template that handles this case would produce the Axiom 6.

$$\begin{gathered}
\forall\, x\ y\ z,\ \mathit{\_toxicity\_n\_1}\ y \wedge \\
(\mathit{\_of\_p}\ z\ y\ x\ \vee\ \mathit{poss}\ z\ y\ x\ \vee\ \mathit{compound}\ z\ y\ x) \\
\rightarrow \exists\, v, \text{«LD50»}\ x\ v
\end{gathered} \tag{6}$$

Provided the axioms 4 and 5 above, Z3 SMT Solver (de Moura and Björner, 2008) can easily

---

[8] This process can later be refined to use the Wikidata Lexemes.

prove the entailment $\Delta, T(\alpha) \models G(\alpha)$ certifying that the SPARQL query is indeed entailment by the HOL formula, the semantics of the original NL question.

We admit that Example (1-a) discussed above is quite simple. We have not addressed the more complicated cases with properties and entities expressed by more than one word (Sag et al., 2002) and complex expressions of units of measurement. The literature is vast on possible methods for linking entities and their use in domain-specific cases (Zhou et al., 2023). However, it is worth highlighting that (1) It seems that few templates deal with the most common cases of variants of syntactic constructions used in English questions we are considering, that is, questions in a technical domain such as chemistry; (2) any entity detection and entity disambiguation (also called entity linking) method can be equally employed in our framework; and (3) Since 2018, Wikidata has also stored linguistic data such as words, phrases, and sentences. This information is stored in new types of entities called Lexemes (L), Forms (F), and Senses (S). These entities can be linked appropriately to Q items and properties, facilitating the disambiguation process during the semantic parsing of the sentence and constructing the axioms above by demand.

## 5. Conclusions and Future Work

In conclusion, we have presented a logic-based approach to validate SPARQL queries derived from translations of natural language (NL) questions. Our focus on addressing the well-documented risks of hallucinations in KGQA amidst the widespread utilization of Large Language Models (LLMs) positions our work as a neuro-symbolic endeavor toward ensuring 'Safe AI.' While much previous research has also explored the use of semantic parsing for question-answering (Gu et al., 2022; Berant et al., 2013) – mainly using machine learning methods for semantic parsing and producing representations like AMR (Banarescu et al., 2013) – and evaluated LLMs in this context (Faria et al., 2023), the novelty of our approach lies in the use of Minimal Recursion Semantics (MRS) produced by ERG, a high-precision computational grammar, and the translation of MRS to higher-order logic (HOL) to represent the semantics of English sentences and the further compositional and deterministic translation of HOL formulas to SPARQL (query) and from SPARQL (to validate).

Central to our methodology is MRS Logic, a Python Library built upon 'deep' linguistic processing technologies from the DELPH-IN Consortium. By extending DELPH-IN tools to translate MRS to HOL formulas and employing ULKB to reason with these formulas and query KGs, our approach bridges linguistic and statistical processing methods for semantic understanding. To the best of our knowledge, our work is the first comprehensive report on the translation of MRS to a higher-order logic language, the subsequent translation of SPARQL to/from HOL, and the use of these methodologies for comparing English questions with SPARQL queries.

While a preliminary evaluation of the MRS Logic capability of translating NL utterances to HOL statements has been conducted using text entailment tests (Rademaker et al., 2023) in the SICK dataset (Marelli et al., 2014), we recognize the necessity of further evaluating our SPARQL validation procedure, mainly as we aim to tackle more challenging questions in domains like chemistry, leveraging insights from existing KGQA systems (Zhou et al., 2023). The translation from HOL to SPARQL is compositional and deterministic, but still, many nuances of NL utterances may not be captured adequately by our current implementation. At this stage, there is no dataset of NL queries in the chemistry domain associated with SPARQL with and without hallucinations to test our approach. Note that we focus on technical domains rather than on general-purpose common sense datasets like LC-QUAD (Trivedi et al., 2017). We are not dealing with unrestricted entities and their properties (people, places, events, etc.) that make entity recognition and entity and word sense disambiguation almost a guess without a reasonable context. The LLM pipeline for SPARQL generation was used precisely for its coverage and robustness, and it is unclear if the symbolic processing will capture few or many of the actual possible queries that a domain expert may submit. A quantitative evaluation of our approach will undoubtedly be necessary in subsequent work. It is worth mentioning the complexity of constructing a domain-specific QA dataset with questions that need to be relevant (not toy examples) and with different levels of complexity.

As part of our future endeavors, we aspire to reimplement our approach using Lean (Moura and Ullrich, 2021), a programming language and interactive theorem prover, thus transitioning from HOL to dependent types. Dependent type theory has been widely acknowledged as a formal tool for understanding natural language (Ranta, 1994; Chatzikyriakidis and Luo, 2020), and exploring this avenue could further enhance the robustness and applicability of our methodology.

## 6. Bibliographical References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In

*The 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece. ACL.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lang Cao. 2023. Enhancing reasoning capabilities of large language models: A graph-based verification approach. *arXiv preprint arXiv:2308.09267*.

Stergios Chatzikyriakidis and Zhaohui Luo. 2020. *Formal Semantics in Modern Type Theories*. Wiley.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage english grammar using hpsg. In *The Second Linguistic Resources and Evaluation Conference*, pages 591–600, Athens, Greece.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3:281–332.

Berthold Crysmann and Woodley Packard. 2012. Towards efficient HPSG generation for German, a non-configurational language. In *COLING*, pages 695–710.

Leonardo de Moura and Nikolaj Björner. 2008. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin. Springer.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Tim Erdmann, Stefan Zecevic, Sarath Swaminathan, Brandi Ransom, Krystelle Lionti, Dmitry Zubarev, Siya Kunde, Stephanie Houde, James Hedrick, Nathaniel Park, and Kristin Schmidt. 2024. Chemchat: Conversational expert assistant in material science and data visualization. In *American Chemical Society (ACS) Spring Meeting*.

Bruno Faria, Dylan Perdigão, and Hugo Gonçalo Oliveira. 2023. Question Answering over Linked Data with GPT-3. In *12th Symposium on Languages, Applications and Technologies (SLATE 2023)*, volume 113 of *Open Access Series in Informatics (OASIcs)*, pages 1:1–1:15, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

William M. Farmer. 2008. The seven virtues of simple type theory. *Journal of Applied Logic*, 6(3):267–286.

D. Flickinger, A. Copestake, and I. A. Sag. 2000. Hpsg analysis of english. In *Verbmobil: Foundations of speech-to-speech translation*, pages 321–330. Springer, Berlin, Germany.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Michael Wayne Goodman. 2019. A python library for deep linguistic resources. In *2019 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)*, Singapore.

Yu Gu, Vardaan Pahuja, Gong Cheng, and Yu Su. 2022. Knowledge base question answering: A semantic parsing perspective. *arXiv preprint arXiv:2209.04994*.

Alexander Koller and Stefan Thater. 2005. Efficient solving and exploration of scope ambiguities. In *The ACL Interactive Poster and Demonstration Sessions*, pages 9–12, Ann Arbor, Michigan. ACL.

Alexander Koller and Stefan Thater. 2006. An improved redundancy elimination algorithm for underspecified representations. In *The 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 409–416, Sydney, Australia. ACL.

Alexander Koller and Stefan Thater. 2010. Computing weakest readings. In *The 48th Annual Meeting of the ACL*, pages 30–39, Uppsala, Sweden. ACL.

Guilherme Lima, Alexandre Rademaker, and Rosario Uceda-Sosa. 2023. Ulkb logic: A hol-based framework for reasoning over knowledge graphs. In *Proceedings of the 26th Brazilian Symposium on Formal Methods*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *The Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. ELRA.

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

Leonardo de Moura and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language. In *The 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*, pages 625–635. Springer.

OpenAI. 2023. Gpt-4 technical report.

C. Pollard and I. A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Alexandre Rademaker, Guilherme Lima, and Renato Cerqueira. 2023. Extracting higher-order logic formulas from english sentences. Under evaluation.

Aarne Ranta. 1994. *Type-theoretical grammar*. Oxford University Press.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Computational Linguistics and Intelligent Text Processing*, pages 1–15.

Stephan Schulz, Simon Cruanes, and Petar Vukmirović. 2019. Faster, higher, stronger: E 2.3. In *Automated Deduction – CADE 27*, pages 495–507. Springer.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QUAD: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

W3C SPARQL Working Group. 2013. SPARQL 1.1 overview. W3C recommendation, W3C. `http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Yoad Winter. 2016. *Elements of Formal Semantics: An Introduction to the Mathematical Theory of Meaning in Natural Language*. Edinburgh University Press.

Xiaochi Zhou, Daniel Nurkowski, Sebastian Mosbach, Jethro Akroyd, and Markus Kraft. 2021. Question answering system for chemistry. *Journal of Chemical Information and Modeling*, 61(8):3868–3880.

Xiaochi Zhou, Shaocong Zhang, Mehal Agarwal, Jethro Akroyd, Sebastian Mosbach, and Markus Kraft. 2023. Marie and bert – a knowledge graph embedding based question answering system for chemistry. *ACS omega*, 8(36):33039–33057.

# How to Turn Card Catalogs into LLM Fodder

**Mary Ann Tan, Shufan Jiang, Harald Sack**
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Eggenstein-Leopoldshafen, Germany
Karlsruhe Institute of Technology, Karlsruhe, Germany
{ann.tan, shufan.jiang, harald.sack}@fiz-kalrsruhe.de

## Abstract

Bibliographical metadata collections describing pre-modern objects suffer from incompleteness and inaccuracies. This hampers the identification of literary works. In addition, titles often contain voluminous descriptive texts that do not adhere to contemporary title conventions. This paper explores several NLP approaches where greater textual length in titles is leveraged to enhance descriptive information.

**Keywords:** NLP, named entity recognition, question-answering, large language model, digital libraries

## 1. Introduction

Cultural heritage (CH) institutions have been spending considerable resources digitizing their vast collections resulting in an overwhelming volume of digitized objects and their metadata. A large proportion of these are organized as linked data. Notable examples include the Rijksmuseum (Alani et al., 2018), WarSampo (Hyvönen et al., 2016), and Europeana (Purday, 2009).

Recently, the European Parliament identified the challenges facing cultural heritage institutions in the context of the emergence of Artificial Intelligence (AI) solutions. One of these challenges is uneven metadata quality (Pasikowska-Schnass and Lim, 2023). Metadata consists of a set of information that describes and provides context to resources.

As the German national aggregator to the Europeana, the *Deutsche Digitale Bibliothek* (DDB) collects metadata from other cultural heritage institutions all over Germany. Its metadata collection has been published on the web[1] and has been made accessible through an API[2].

More than a quarter of the DDB's entire holdings is composed of 13.5 million[3] digitized texts from the libraries. Part of the digitization process and the subsequent creation of these metadata involved taking information from both existing physical catalog cards and digital sources. The fact that the age of these objects spans several millennia leads to a high level of uncertainty.

Due to the evolution of cataloging standards [4] and the age of some of the objects, author attribution, creation date, and subject heading classifications are missing (See Section 2, Figure 1). The absence of this information, which facilitates item identification in a contemporary library, makes search and retrieval a laborious process. These challenges also makes content exploration and recommendations unfeasible.

Using Semantic Web Technologies (SWT), the metadata collection of the DDB is currently encoded as linked open data and stored in a knowledge graph (KG) (Tan et al., 2021b).

Section 2 provides a thorough description of the metadata collection. Section 3 provides a review of related literature, while Section 4 and 5 describe in detail the main contributions of this paper:

- How different NLP tasks and models can be leveraged to address the challenges of metadata incompleteness and inaccuracy.
- How the results of the experiments can help librarians improve their metadata.

Finally, section 6 presents the conclusion and future work.

## 2. The DDB Collection

The metadata collection of the DDB conforms to the information exchange and description standard specified by the Resource Description Framework (RDF). The metadata is represented using an extension of the Europeana Data Model (EDM) [5]. In accordance with the EDM standards, the DCMI Metadata Element Set [6] (Dublin Core or DC) and the DCMI Metadata Terms [7] (DC Terms or DCT) properties are used to describe a resource.

---

[1] DDB, `https://www.deutsche-digitale-bibliothek.de`

[2] DDB Rest API, `https://labs.deutsche-digitale-bibliothek.de/app/ddbapi/`

[3] As of March 2024

[4] The provenance of catalogs used as source of the metadata is not available to the DDB. This assumption is primarily based on the time span covered by the collection.

[5] EDM, `https://pro.europeana.eu/page/edm-documentation`

[6] Dublin Core, `https://www.dublincore.org/specifications/dublin-core/dces/`

[7] DC Terms, `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/`

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 53–65
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

The DDB dataset is divided into seven (7) sectors, each corresponding to the type of institution from which the metadata originates, namely, archives, libraries, historical preservation, research, media libraries, museums, and the rest. Participating institutions are numbered in the hundreds. This paper is focused on the metadata provided by libraries.

The flexibility afforded by the EDM in the cataloging process and the large number of contributing institutions lead to the uneven quality of the metadata collection, since only `dc:title` is indicated as mandatory.

In the DDB, a single book may be composed of several digitized objects, such as the front cover, *Ex Libris* page, table of contents, a chapter, a section, or a page showing an illustration. Each digitized object is equivalent to a single metadata record, which is then defined as an instance of the class *edm:ProvidedCHO*. To distinguish these digitized objects from each other, the data property *ddb:hierarchyType* is used.

In addition, an object can either be a *primary* or *secondary* object. This is indicated by the object property *dcterms:isPartOf*. The primary object of a book is the cover page, while the other components are the secondary objects.

Due to the heterogeneous, hierarchical and highly-granular nature of the bibliographic collection, the metadata is aligned to another data model that reflects the standards defined by the Functional Requirements for Bibliographic Records (FRBR) (Tillet, 2004). The main classes in FRBR correspond to the four (4) conceptual entities: *frbr:**Work***, *frbr:**Expression***, *frbr:**Manifestation***, and *frbr:**Item*** or "*WEMI*".

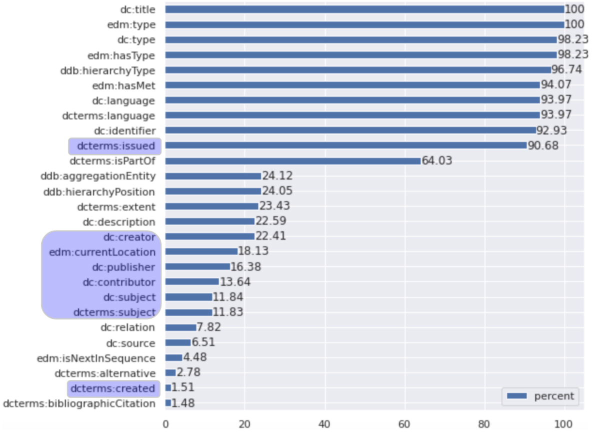


**Figure 1:** List of properties in the DDB

Tan et al. (2021a) map instances of *edm:ProvidedCHO* to their respective entities in FaBiO or FRBR-Aligned Bibliographic Ontology. Ideally, a primary object such as a cover page can be mapped to its corresponding *Work* entity using properties that distinguish one literary work from another. The title, author, creation date, and subject headings are required to properly identify a literary work. This mapping is necessary since users are more likely to search for higher level representations, *Work* and *Expression* levels, rather than *Manifestation* and *Item* levels.

In a contemporary library, these properties are readily available, often written on catalog cards. However, as can be seen in Figure 1, the property corresponding to the author (*dc:creator*) exists only 22% of the time, while creation date (*dcterms:created*) is specified 1.5% of the time. Moreover, the codificaton of card cataloging rules had not been established prior to the French Revolution; it was only in 1791 when the French Cataloging Code was established (Hopkins, 1992), it is highly likely that inaccurate or incomplete information from old card catalogs, created from the times before then, were carried over during the digitization process.

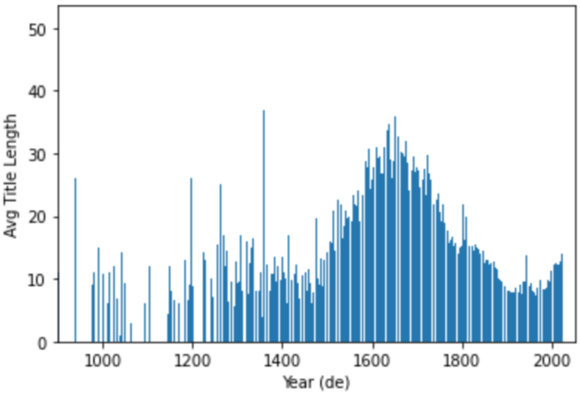


**Figure 2:** Distribution of title lengths.

A notable example of this phenomenon is the range of values encoded in the data property *dc:title*. As can be seen in Figure 2, the average length of German titles by number of tokens is considerably longer before the French Revolution than after. These titles often contain voluminous descriptive texts that do not adhere to contemporary title conventions.

All example titles from hereon have to be redacted (`<...>`) due to space constraints. Appendix A lists these examples including full titles, their translations, URLs, and metadata.

As can be seen in Example 1, the title contains the author, creation date and location, subject heading, and a short description of the content.

> *Die Letzte Predigt, **Doctoris Martini Lutheri**, heiliger Gedechtnis: So er gethan hat zu **Wittemberg** ... **den 17. Januarij, im 1546. Jar** :Darinnen wir für falschen Lehrern gewarnet ... werden*

**Example 1:** Martin Luther's last sermon.

Taking into account the very features that would be considered a disadvantage by present day cataloging standards, this paper explores several NLP approaches where greater textual length and more information contained in the titles might be advantageous.

Moreover, the name seen in this example is "*Doctoris Martini Lutheri*", which is the genitive case of Martin Luther's latinized name. The names found in the titles are not normalized: they can be misspelled or in the wrong language, they can contain professional titles ("*Doctoris*"), honorifics, and/or official designations. These naming variations require a more forgiving matching criterion during evaluation.

Another notable example is a dedication written by Lorenz Pscherer for King Gustav II Adolph of Sweden (r. 1611-1632) (Example 2). The metadata attributes authorship (`<dc:creator>`) to "`Horky, Martin *ca. 17. Jh.*`", while the role of "`Pscherer, Lorenz`" is labeled as one of the `<dc:contributor>`'s. Moreover, the dedicatee, King Gustav II Adolph of Sweden, is described as a contributor rather than a subject heading.

> *Ein frölicher Triumph Wagen/ Von der Göttlichen* `[...]` *Gottfürchtige und gelerte Mann* ***Laurentius Pscherer*** *zu* ***Nürnberg*** *gehabt/ und nu mehr dem* ***7. Septembris Anno 1631****. sich* `[...]`

**Example 2:** The title containing Lorenz Pscherrer's name.

These inaccuracies add another layer of complexity in the automatic construction of an evaluation dataset.

## 3. Related Work

The popularity of KGs arose from their ability to encode real-world information using nodes and vertices: the nodes to represent entities or individuals, and the vertices to represent the relationships that exist between these entities.

However, due to the open world assumption, KGs are in practice incomplete, or worse, incorrect. To mitigate the issue of incompleteness, KG completion approaches such as Link Prediction (Rossi et al., 2021) and Entity Alignment (Zeng et al., 2021) became the *de facto* solutions. Both approaches harness the approximation power of KG Embeddings (KGEs) by adding missing information into the KG.

Research into KG construction benefited from advances in Information Extraction (IE), an important branch of NLP. IE provides a scalable solution to KG construction by automatically turning unstructured data, such as texts, into structured or semi-structured data. IE pipelines are often composed of several modules, including, but not limited to, the following: Named Entity Recognition (NER), Entity Linking (EL) and Relation Extraction.

It is possible to enhance the DDB metadata collection by identifying pertinent information from lengthy titles using an IE pipeline composed of fine-grained NER and EL. In this work, we focus on the identification and classification of bibliographic entities.

A recent survey (Ehrmann et al., 2023) summarizes the challenges of NER in historical documents by pointing to the variety of historical document types, topics and domains, noisy input derived from optical character recognition (OCR), handwritten text recognition (HTR), dynamics of language and lack of resources. The use of pre-trained language models in transfer learning leverages knowledge from unlabeled historical corpora. It captures historical language idiosyncrasies during the pre-training phase before adapting the models to a specific NER task in the fine-tuning phrase. The pre-training-fine-tuning paradigm requires task-specific model architecture and storage; it also needs a certain amount of expert annotation. We found the following labeled datasets for German historical and bibliographic named entities:

- AjMC dataset (Romanello et al., 2021; Romanello and Najem-Meyer, 2022) consists of NE-annotated multilingual 19 $^{th}$ century classical commentaries and contains 3,500 mentions of German names, of which 356 are classified as authors.
- CLEF-HIPE 2020 (Ehrmann et al., 2020), a multilingual historical news corpus covering a time span of 200 years, contains 660 mentions of organizations, 58 of which are classified as press agencies.
- NewsEye (Hamdi et al., 2021) consists of annotated multilingual historical newspaper materials published between 1850 and 1950, containing 3,500 German names, of which 30 are classified as article authors.

These labeled datasets are relatively small; covering short time spans, a narrow range of topics and limited materials; this necessitated the creation of our own ground truth data.

The Question Answer (QA) task is another possible solution to leverage the potential of existing data in Language Models (LMs) for IE in a low resource setup. Given a passage and a question, the goal is to provide an answer to be extracted from a given passage. Best performing approaches use the SQUAD (Rajpurkar et al., 2016) dataset and its extensions for training and fine-tuning. This dataset contains handcrafted, general questions and answers drawn from excerpts of top Wikipedia pages. Depending on the passage and the type of question, the expected answer may be simple or complex. For the current use case, the title is used as the passage. The questions are formulated such that the expected answers are simple and explicit (i.e. author names, creation date, etc.)

The recent proliferation of Large Language Models (LLMs) spurred intense research activity due to the generalization, language understanding and generation ability of LLMs (Wei et al., 2022). Several notable studies provided an in-depth analysis of pre-trained language models on how well they can recall factual knowledge using a series of probing questions (Petroni et al., 2019; Poerner et al., 2020). A fact is formulated as a triple consisting of a subject, a relation, and an object. An LLM is said to "know" a fact, if it can fill in the masked relation in a cloze statement, i.e. Dante [MASK] Florence, where [MASK] is the relation *birthplace*. Petroni et al. (2020) concluded that providing relevant context to the LLM improves fact retrieval performance.

Fine-tuning LLMs for the purpose of this study was not possible due to limited access to computational power, neither was it possible to consult experts for manual annotation of the current dataset.

Regarding the application of LLMs for Historical IE, (De Toni et al., 2022) explores the zero-shot abilities of the `T0` model for coarse-grained NER over the CLEF-HIPE 2020 dataset (Ehrmann et al., 2020) with a naive prompt-based approach; it showed the `T0`-like models' potential to probe for language tags and publication dates.

## 4. Methodology

This section describes dataset construction (Section 4.1), the evaluation procedure and metrics (Section 4.2), and NLP models used for experimentation, and the experimental setup (Section 4.3).

### 4.1. Dataset

In order to construct our dataset for experimentation and evaluation of the aforementioned approaches, the entire DDB bibliographic metadata collection has been filtered down to a manageable representative sample.

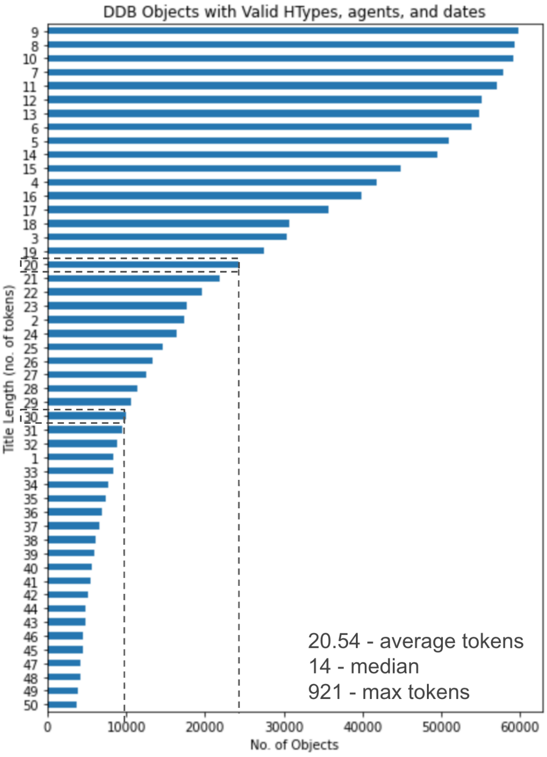


**Figure 3:** Title token distribution of German titles in the DDB.

The DDB has objects in more than 200 languages. The scope of this study is limited to digitized textual objects tagged as “`ger`" for German and “`zxx`" for unknown or no language tag. The Python library *langid* (Lui and Baldwin, 2012) is used to confirm that the titles of these objects are indeed in German, since there are objects where the language of the title is not the same as the value indicated in the metadata[8]. There is a considerable number of objects with Latin titles that have been tagged as German. In addition, since the collection grew across a long period of time, the German language has had time to evolve. Hence, there are titles written in different versions of the German language from Middle High German (see Example 1) to Standard High German.

Using the *ddb:hierarchyType* mentioned in Section 2, the metadata describing textual objects[9] is further reduced to approximately 30% of its original size. The hierarchy types selected are Monograph, Chapter, Essay, Volume, Manuscript, Letter and Multi-Volume Work, since these are likely to have identifiable titles.

In order to have a ground truth, the representative objects are described with agents (`<dc:creator>`, `<dc:publisher>`, `<dc:contributor>`) and dates (`<dcterms:issued>`, `<dcterms:created>`). Moreover, since the goal is to leverage lengthy titles, the representative objects should have more tokens than the average length of 20.54 (Figure 3). Choosing 30 tokens to be the cut-off still leaves a little over 100k objects after the final pruning criterion. The remaining objects are pruned for the final time to only contain books (`<dc:type> = "Monografie"`), since this object type suggests a physical manifestation in the context of FRBR and can possibly be aligned to their respective higher-level entities in FaBiO.

---

[8] `https://www.ddb.de/item/DNEBFCMMEO52LAQWGT5JHULKPXBU2QYG`

[9] `<edm:hasType> == 'TEXT'`

Table 1 provides some statistics about the pruned dataset. Figure 4 in Appendix C shows the distribution of title lengths with respect to age of the objects after pruning.

| Characteristic | Value |
|---|---:|
| No. of Objects | 108,827 |
| Average Title Length | 55.49 Tokens |
| Median Title Length | 47 Tokens |
| Longest Title Length | 364 Tokens |

**Table 1:** Characteristics of the dataset.

## 4.2. Evaluation Guidelines

The goal of this study is to find out how well a particular approach can retrieve identifying information included in the title, such as dates and agents.

Dates are trivial to compare. The dates stored in the evaluation dataset only include the year element in ’`YYYY`’ format. For metadata values that include month and day, a regular expression is applied to retrieve the year. As in the example in section 2, “1956" is compared against the values of either `<dcterms:created>` or `<dcterms:issued>`, while ignoring “*den 17. Januarij, im...*"

Agents, in the bibliographic domain, refer to the persons responsible, in any capacity, for the creation of the object. Author, editor, and publisher are the roles often attributed to these agents. The properties `<dc:creator>`, `<dc:contributor>`, and `<dc:publisher>` store names of persons in the format of “`last name, first name`" following the German version of the name, without title, honorific, or official designation.

Exact name matching is non-trivial, as mentioned in Section 2 and illustrated in Example 1. To facilitate approximate name matching, an extension of the Python package *sqlite-spellfix* [10] is used.

*Spellfix* is implemented as a virtual table that stores all the vocabulary terms and uses Levenshtein distance (Levenshtein, 1966) to compute edit distance in order to gauge the lexical similarity between the vocabulary terms and the search string.

The agents’ names found in the ground truth are collected and stored as vocabulary terms. The reference table of the *Spellfix* virtual table is composed of two (2) columns: the person’s name normalized in the format of “`firstname lastname`"; and the object ID, a 32-character unique identifier of an object, associated with the agent. To illustrate, if a model is able to extract “*Martini Lutheri*" from the text, this string is used to lookup the most similar names found in the *Spellfix* virtual table and the corresponding object ID linked to these names as defined in the primary table. If the ID of the object currently being evaluated is found in the list of the object IDs resulting from the *Spellfix* lookup results, then it is considered a match.

It is important to note that some objects are annotated with agents that cannot be found in the title. Authors are more likely mentioned in the title, like “Schiller’s Robbers"[11], while editors and publishers rarely are. However, the latter roles can still be associated with the properties *dc:contributor* and *dc:publisher*. This will not affect the evaluation results, since we test the results on the list of all agents and role-specific agents (“Ground Truth" column in Tables 6 and 8). In addition, a more forgiving *Precision@n* metric is used, where *n* varies depending on the number of agents associated with an object. If there are 2 agents associated with an object, and only a single name gets a match, *Precision@n* will be equal to 1 for this specific object. Appendix B shows some of the matches related to the LLM experiments.

The applied metric deviates from the customary precision, recall, and $F_1$ score combination for IE, due to the nature of the ground truth, and the variety of name formats found in the text. On the other hand, this metric is similar to the *Top1Acc* measure for the extractive QA task meant for closed-domain evaluation, where 1 point is attributed if the predicted answer has a single word overlap with the labeled answer. As for the task involving LLM, the model is instructed to only provide names without justifications. Therefore, the same metric is used during evaluation.

## 4.3. NLP Tasks and Models

This subsection describes how the use case of the DDB is recast into the three chosen NLP tasks: (1) **NER**, (2) Extractive **QA** and (3) Open Generative QA using an **LLM**. Moving forward, Task 2 will simply be referred to as **QA** while task 3 as **LLM**

**NER.** Since the goal is to extract the people, dates, and possibly, subject headings, from a lengthy title, it is appropriate to adopt an IE pipeline. The current state-of-the-art, general-purpose, open source, and off-the-shelf model is the FLAIR English NER Large Model (FLERT[12]) (Schweter and Akbik, 2020). Despite being classified as an English model, its pre-trained Language Model (PLM) is based on XLM-R (Conneau et al., 2020). Choosing FLERT is motivated by its multilingual representations capability and its ability to identify 18

---

[10] sqlite-spellfix,https://pypi.org/project/sqlite-spellfix

[11] Schiller’s Räuber, https://www.ddb.de/item/FXHCBDNJAAHI7PSMOYBMKZS5I47NX36J

[12] FLERT, https://huggingface.co/flair/ner-english-ontonotes-large

different entity types, including dates (`DATE`) and works (`WORK_OF_ART`).

Further classification of PERSON entities according to specific bibliographic roles, whether author, editor, or publisher, calls for a fine-grained NER approach. Such a requirement necessitates an expert-annotated dataset that can be used for fine-tuning (Radford and Narasimhan, 2018; Peters et al., 2019) to produce a domain-specific, fine-grained NER model as in LegalNER (Leitner et al., 2019; Akbik et al., 2018). This currently exceeds the scope of this study and is being considered for future work.

For this task, the model is expected to find entity mentions and to classify them given a title. In example 2, the FLERT model recognizes 3 highlighted entities: "Laurentius Pscherer" as `PERSON`, "Nürnberg" as `GPE` and "7. Septembris Anno 1631" as `DATE`. For this specific use case, only the `PERSON` and `DATE` entities are scrutinized.

Following the evaluation procedure and metric described in section 4.2, FLERT's prediction results in a single matching point for each person and each date.

**QA.** Existing "Extractive" QA models can be retrofitted for the purpose of the DDB. Using the title as the passage, below is the list of simple questions posed to the models:

1. Who is the author? ("*Wer ist der Autor?*")
2. Who wrote the text? ("*Wer hat den Text geschrieben?*")
3. Who is the publisher? ("*Wer ist der Herausgeber?*")

The best German QA models available are fine-tuned using the German equivalent of the Wikipedia articles used in SQUAD, aptly named GermanQUAD (Möller et al., 2021). Using GELECTRA (Chan et al., 2020) as the PLM, these models are fine-tuned with the goal of *extracting* relevant parts of the passage with dense representation to be most similar to the corresponding dense representation of the question. Since the goal is to retrieve the names of the persons from the passage whose specific role is indicated in the question, and the German QA models adopted are not trained on unanswerable QA pairs, it is necessary to ensure that only titles with names are included in the test by using those identified by the NER model to have PERSON entities.

To find out how well the German QA models compare to one of the top 3[13] English QA models, experiments also used the `roberta-large-squad2` model[14] published by Deepset. This model is fine-tuned using Squad 2.0 (Rajpurkar et al., 2018). Squad 2.0 is an extension of SQUAD that includes an additional set of 50,000 handcrafted, adversarial questions that have no answers but are very similar to existing answerable questions.

In order to do so, the German titles are translated into English using the DE-EN machine translation model submitted by Facebook's FAIR for the WMT19News Translation Task[15], which boasts a SacreBLEU score of 40.8 (Ng et al., 2019). The translations of the titles used in the examples are listed in Appendix A

Table 2 shows an example of the answers of the different QA models: both the `GELECTRA`-based models are provided with the original title in German as context, while the `RoBERTa`-based (Liu et al., 2019) model is fed with the English machine-translated title. Despite being given a translated text produced by a moderately performing machine translation model, the confidence score of the English QA model is still considerably higher than the German QA models. Nevertheless, these scores are not taken into account since only the answers matter during evaluation.

| MODEL | ANSWER | SCORE |
|---|---|---|
| `gelectra-base-germanquad` | Doctoris Martini Lutheri | 0.0539 |
| `gelectra-large-germanquad` | Doctoris Martini Lutheri | 0.0115 |
| `roberta-large-squad2` | Doctoris Martini Lutheri | 0.9425 |

**Table 2:** Answers of different QA models when asked about the author of Example 1.

**LLM.** With the optimal mix of instructions, LLMs trained as conversational agents are known to generate impressively coherent and sometimes factual texts. The prompts used for the experiments are patterned after the guidelines provided by Bsharat et al. (2024). Specifically, the following principles are incorporated into the prompts:

- P16: Assign a role to the large language models.
- P8: Use line breaks to separate instructions.
- P25: Clearly state the requirements.

The series of instructions used to test the chosen LLM is provided below. Lines 3-5 are explicitly specified to suit the evaluation procedure described in Section 4.2. Line 6 varies depending on the question that needs to be asked (author, publisher, etc.) Line 7 contains the full title. Given the title as the context, this task is categorized as a *Open* Generative QA task.

1. `You are a librarian doing cataloging work.`

---

[13] As of March 2024, https://paperswithcode.com/sota/question-answering-on-squad-v2

[14] https://huggingface.co/deepset/roberta-large-squad2

[15] https://huggingface.co/facebook/wmt19-de-en

2. `Respond with "I don't know" when uncertain.`
3. `Enumerate your answers with numbers.`
4. `Only answer with the name of the persons.`
5. `Do not provide justifications.`
6. `Who is/are the publisher/s of this text?`
7. `"The Last Sermon, Doctoris Martini Lutheri, Sacred..."`

`Mistral-7B-Instruct-v0.2`[16] is an open source LLM developed by Mistral (Jiang et al., 2023). The `Mistral-7B` PLM is fine-tuned on an instruction dataset developed by HuggingFace. This dataset contains "high-quality, diverse, human-written instructions with demonstrations"[17]. Since the model is trained and fine-tuned with English datasets, the machine-translated titles are used for the succeeding experiments. The choice of `Mistral-7B-Instruct-v0.2` is motivated by its availability (open source) and published performance besting state-of-the-art open source LLMs at the time of this writing.

```
1.  Martin Luther
2.  Wittenberg:  Druck von
Paulus Berckmann
Or:
1.  Paulus Berckmann (printed
by)
```

**Example 3:** An Example Response from `Mistral-7B-Instruct-v0.2`.

Example 3 shows the response of `Mistral-7B-Instruct-v0.2` given the aforementioned series of instructions concerning the publisher and provided with the machine-translated title in Example 1. This tests the model's ability to respond with "I don't know". However, the model ignored the instruction, and instead "hallucinated" at least 2 names. The publisher was never in the title and the value of `<dc:publisher>` in the metadata of Example 1 is "Bergen".

## 5. Experiments and Results

The goals of the experiments are to find out the following:

- To what extent can coarse-grained, general purpose NER models be used in filling missing metadata descriptions?
- How can a NER model be leveraged to further refine the evaluation dataset for QA and LLM tasks?
- How well can QA models identify the different agent roles?
- How does an LLM-based chat model compare to a QA model in identifying different agent roles?

The configurations and parameters used for the succeeding experiments are kept according to the published default settings.

### 5.1. NER

Using the dataset constructed in Section 4.1, all 108,827 records are processed with FLERT. Since `FLERT` is a general-purpose NER model, it is not able to distinguish between the different agent roles. For this task, the ground truth is composed of the values of the 3 agent-related properties: `<dc:creator>`, `<dc:contributor>`, and `<dc:publisher>`.

| | **PERSON** | **DATE** |
|---|---|---|
| **Exact Match** | 9.61% | 27.68% |
| **Approx. Match** | 8.42% | N/A |

**Table 3:** `FLERT`'s Precision@n results.

The disappointing results of Table 3 can be partially explained. Although only lengthy titles are considered, it is possible that despite the large number of tokens, a title might not contain any entity mentions of `PERSON` or `DATE`. Table 4 shows the number of proportion of objects from the dataset where PERSON, DATE or both entity types are detected by `FLERT`.

| **PERSON** | **DATE** | **Both** |
|---|---|---|
| 59.07% | 49.12% | 32.93% |

**Table 4:** Proportion of objects with PERSON and DATE entity mentions detected by `FLERT`.

This specific case is shown in Example 4. The NER model correctly identifies "*Das Hohe Lied des Königes Salomons*" (The Song of Songs of King Solomon) as a `WORK_OF_ART`. A 4-tag model such as Flair's German NER (Large)[18] predicts "Salomon" as a `PERSON`, which is only partly correct.

***Das Hohe Lied des Königes Salomons***
*: Wie es/ Zu der aus Gott wieder-geboren-*
`[...]` *... ausgefärtiget hat*

**Example 4:** The Song of Songs, a lengthy title without `PERSON` or `DATE` entities.

Although the results of `FLERT` cannot differentiate between an author and a publisher, this step can already identify possible objects for metadata enhancement, just by identifying the very existence

---

[16] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[17] `https://huggingface.co/datasets/HuggingFaceH4/instruction-dataset`

[18] `https://huggingface.co/flair/ner-german-large`

of the entity mentions. Moreover, by further pruning objects without `PERSON` entities, the dataset can be further improved for the succeeding tasks, particularly for the extractive QA task where the answer is expected to be present in the context.

## 5.2. QA

For this task, the evaluation dataset is reduced to only those records that yielded agent matches according to the previous NER model. This step is necessary to ensure that the QA models are provided only with questions where the answers exist in the passage. This cuts down the original dataset by 84% to 17,084 titles.

| **Ground Truth with all Agents** | `gelectra-base-germanquad` | `gelectra-large-germanquad` | `roberta-large-squad2` |
|---|---|---|---|
| Context | Title (DE) | Title (DE) | Title (EN) |
| "Who is the author?" | 62.94% | **66.23%** | 63.07% |
| "Who wrote" "the text?" | 58.12% | 60.83% | 58.36% |

**Table 5:** QA results against ground truth containing all names.

Table 5 shows that the result of the best performing model, `gelectra-large-germanquad`, is consistent with its published Exact Match (EM) results of 68.6%. The differences between Middle High German and Standard High German do not seem to matter as much. The results also show that asking direct questions yielded some improvement (i.e. by providing specific roles).

To check whether the models understand the difference between author and publisher, the list of names in the ground truth is made more specific according to the question, such that only values described under `dc:creator` are included in the reference list when asked about the author, and respectively, when asked about the publisher.

|  | **Ground Truth** | `gelectra-large-germanquad` | `roberta-large-squad2` |
|---|---|---|---|
| "Who is the author?" | `<dc:creator>` | **32.19%** | 31.16% |
| "Who is" "the publisher?" | `<dc:publisher>` | 0.85% | 0.78% |

**Table 6:** QA results against ground truth containing all names.

The results in Table 6 are inconclusive, because publishers are rarely mentioned in the title. However, looking closely at the title in Example 5, the two names mentioned have two distinct roles:

- `<dc:creator>`: “Ignatz"[19]
- `<dc:contributor>`:“Johann Jacob Ferber"[20]

> *Des Hrn.* ***Ignatz****, Edl. von Born, Ritters, K.K. Berg-Raths,* [...] *Gesellschaft zu Padua Mitglieds [et]c. Briefe über* [...] *und Nieder-Hungarn, an den Herausgeber derselben,* ***Johann Jacob Ferber****, Mitglied der Königl.* [...] *zu Florenz, geschrieben*

**Example 5:** The agent roles of Ignaz von Born and Johann Jacob Ferber.

Table 7 shows the responses of `gelectra-large-germanquad` and `roberta-large-squad2`, incorrect answers are highlighted.

| **Question** | `gelectra-large-germanquad` | `roberta-large-squad2` |
|---|---|---|
| ...Author<->Autor? | **Johann Jacob Ferber** | **Johann Jacob Ferber** |
| ...Editor<->Redakteur? | Johann Jacob Ferber | Johann Jacob Ferber |
| ...Herausgeber? | Johann Jacob Ferber | - |
| ...Verfasser? | **Johann Jacob Ferber** | - |
| ...Publisher<->Verleger? | **Johann Jacob Ferber** | **Mr. Ignatz, Edl. von Born** |

**Table 7:** QA models not being able to tell the different agent roles.

Depending on the historical context of the object, the translation of the German term *Herausgeber* can either be editor or publisher. *Redakteur* is almost always the direct translation of editor, while *Verfasser* means author, and *Verleger* means publisher. Despite providing the passage in the language native to the respective QA models, these models have difficulty distinguishing agent roles. This limitation could be due to the fact that the titles are fragmented texts and the roles being asked do not explicitly appear with the names mentioned.

## 5.3. LLM

Using the instructions described in Section 4.3, Table 8 shows that the LLM is less precise when asked about the author, but performs better compared to the QA model in all other experiments conducted.

| **Question:** **"Who is the ...?** | **Ground Truth** | **LLM** `mistral-7b instruct-v0.2` | **QA** `gelectra-large-germanquad` |
|---|---|---|---|
| **Author** | all agents | 51.60% | **66.23%** |
|  | `<dc:creator>` | **37.60%** | 32.19% |
| **Publisher** | `<dc:publisher>` | **2.70%** | 0.85% |

**Table 8:** LLM vs QA results.

When inspecting the responses closely, `Mistral7BInstructv0.2` occasionally makes up names (See Example 3), is not following instructions with regard to formatting (Appendix B #5) and still provides justifications (Appendix B #6), despite being told not to do so.

---

[19] `https://d-nb.info/gnd/118661868`

[20] `https://d-nb.info/gnd/118686690`

### 5.4. Discussions

Revising tens of millions of metadata records is a daunting task. With the help of these NLP models, it is possible to identify candidate objects for refinement. Concretely, an object lacking in descriptive information, but with a lengthy title from which an NER model may be able to extract pertinent entities, can automatically signal further attention from librarians. Even when the extracted entities are not entirely accurate, the results can be used as suggestions in a post-ingestion editing workflow. The level of post-processing required for each of the objects can also be automatically determined. For instance, those objects whose titles do not yield any results when fed into an NER model will require more work than others.

Since there is currently no gold standard dataset, both QA models and LLMs are meant to gauge their efficacy in determining fine-grained agents. In this setting, objects identified by these models as having authors, but without matching values against `<dc:creator>` indicate the need for manual intervention. In this scenario, the models' results can be leveraged for possible refinements. For example, an extracted agent may already be indicated as a `<dc:contributor>`, in which case, the metadata can be made more accurate by defining this agent as a `<dc:creator>`. Another possibility would be to fill out missing `<dc:subject>`.

The disparity of the adapted models in terms of their published performance and the results shown in this paper can be attributed to several factors.

Primarily, the titles used for the experiments contain fragmented texts in older versions of the German language where spelling and naming conventions changed over time. In contrast, these off-the-shelf models were pre-trained and fine-tuned on contemporary, general purpose texts. This limitation calls for future work in adapting models trained on texts whose age and domain overlap with the DDB dataset. In addition, the absence of a gold standard evaluation dataset limits the validity of the results. This limitation will be the first to be addressed in the next iteration of our work.

Although the experiments conducted with `gelectra-base-germanquad` and `gelectra-large-germanquad` lacked adversarial questions, this limitation was partly mitigated by comparing their results with `roberta-large-squad2`, an English QA model trained on the Squad 2.0 dataset, which includes unanswerable questions. Nonetheless, this calls for further experiments that include titles without entity mentions.

The last limitation concerns the unpredictability of the LLMs and the difficulty of formulating the most optimal prompts. This affects the reproducibility of the experiments conducted in section 5.3.

## 6. Conclusion

The challenge of incomplete and inaccurate bibliographical metadata collection, the linked data source of DDB-KG, can be addressed using a combination of NLP tasks. The results show that NER, QA and LLMs can, to some extent, be used to extract some bibliographic properties from lengthy titles of historical objects. A domain-specific dataset is currently being prepared for a fine-grained NER model capable of determining literary work title, agent roles, dates, and subject headings.

The experiments make use of an evaluation dataset where the agent roles encoded in the metadata are not entirely accurate. While domain experts are necessary in the preparation of a more precise dataset for future DDB-KG enhancement initiatives leveraging AI models, domain experts can also benefit from the rapid approximation capabilities of AI models. In particular, the list of objects with `PERSON` entities that are not matching any answers provided by either QA models or LLMs may be used as an initial list of objects to undergo expert scrutiny for a possible revision.

Further experiments are planned to compare NLP models that are relevant to the DDB dataset. It is worthwhile to test the efficacy of models trained on $19^{\text{th}}$-$20^{\text{th}}$ historical German text (Ehrmann et al., 2023). Moreover, once the aforementioned gold standard dataset is available, further experiments will be conducted using state-of-the-art commercial LLMs.

It is the ultimate goal to develop a collaborative tool for metadata providers where inputs from both domain experts and AI models can be combined to provide better results in search, retrieval and exploration of cultural heritage.

## 7. Acknowledgements

We would like to express our gratitude to Frank Pöhlmann who dedicated his time and editing experience to provide valuable feedback and constructive criticism on this paper.

## 8. Ethics Statement

The authors of this paper are affiliated with FIZ Karlsruhe and Karlsruhe Institute of Technology. Funding was provided solely by FIZ Karlsruhe. The work done on this paper has been conducted ethically. No part of this paper was suggested, generated, improved, or corrected using generative AI models.

## 9. Bibliographical References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Harith Alani, Chris Dijkshoorn, Lizzy Jongma, Lora Aroyo, Jacco van Ossenbruggen, Guus Schreiber, Wesley ter Weele, and Jan Wielemaker. 2018. The rijksmuseum collection as linked data. *Semant. Web*, 9(2):221–230.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Francesco De Toni, Christopher Akiki, Javier De La Rosa, Clémentine Fourrier, Enrique Manjavacas, Stefan Schweter, and Daniel Van Strien. 2022. Entities, Dates, and Languages: Zero-Shot on Historical Texts with T0. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 75–83, virtual+Dublin. Association for Computational Linguistics.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2023. Named entity recognition and classification in historical documents: A survey. *ACM Comput. Surv.*, 56(2).

Maud Ehrmann, Matteo Romanello, Alex Flückiger, and Simon Clematide. 2020. Extended overview of clef hipe 2020: named entity processing on historical newspapers. In *CLEF 2020 Working Notes. Conference and Labs of the Evaluation Forum*, volume 2696. CEUR-WS.

Ahmed Hamdi, Elvys Linhares Pontes, Emanuela Boros, Thi Tuyet Hai Nguyen, Günter Hackl, Jose G. Moreno, and Antoine Doucet. 2021. A Multilingual Dataset for Named Entity Recognition, Entity Linking and Stance Detection in Historical Newspapers. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2328–2334, Virtual Event, Canada. ACM.

Carla Hayden. 2017. *The Card Catalog: Books, Cards and Literary Treasures*, 1st edition. The Library of Congress, Chronicle Books, San Francisco, CA.

Judith Hopkins. 1992. The 1791 french cataloging code and the origins of the card catalog. *Libraries & Culture*, 27(4):378–404.

Eero Hyvönen, Erkki Heino, Petri Leskinen, Esko Ikkala, Mikko Koho, Minna Tamper, Jouni Tuominen, and Eetu Mäkelä. 2016. Warsampo data service and semantic portal for publishing linked open data about the second world war history. In *The Semantic Web. Latest Advances and New Domains*, pages 758–773, Cham. Springer International Publishing.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained Named Entity Recognition in Legal Documents. In *Semantic Systems. The Power of AI and Knowledge Graphs. Proceedings of the 15th International Conference (SEMANTiCS 2019)*, pages 272–287.

Vladimir I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Magdalena Pasikowska-Schnass and Young-Shin Lim. 2023. Artificial intelligence in the context of cultural heritage and museums: Complex challenges and new opportunities. Technical Report PE 747.120, European Parliamentary Research Service, Brussels.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Jon Purday. 2009. Think culture: Europeana.eu from concept to construction. *Bibliothek Forschung und Praxis*, 33(2):170–180.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Matteo Romanello and Sven Najem-Meyer. 2022. Guidelines for the annotation of named entities in the domain of classics, march 2022. *DOI: https://doi. org/10.5281/zenodo*, 6368101.

Matteo Romanello, Sven Najem-Meyer, and Bruce Robertson. 2021. Optical character recognition of 19th century classical commentaries: the current state of affairs. In *Proceedings of the 6th International Workshop on Historical Document Imaging and Processing*, HIP '21, page 1–6, New York, NY, USA. Association for Computing Machinery.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, 15(2).

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition.

Mary Ann Tan, Tabea Tietz, Oleksandra Bruns, Jonas Oppenlaender, Danilo Dessì, and Harald Sack. 2021a. DDB-EDM to FaBiO: The Case of the German Digital Library. *The Semantic Web – ISWC 2021*.

Mary Ann Tan, Tabea Tietz, Oleksandra Bruns, Jonas Oppenlaender, Danilo Dessì, and Harald Sack. 2021b. DDB-KG: The German Bibliographic Heritage in a Knowledge Graph. In *6th International Workshop on Computational History at JCDL – Histoinformatics*, volume 2981. CEUR-WS.org.

Barbara Tillet. 2004. What is FRBR?: A Conceptual Model for the Bibliographic Universe.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Kaisheng Zeng, Chengjiang Li, Lei Hou, Juanzi Li, and Ling Feng. 2021. A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, 2:1–13.

# A. Appendix A

**Titles and their Details**

## A.1. Example 1: Martin Luther's Last Sermon.

- **Title:** *Die Letzte Predigt, Doctoris Martini Lutheri, heiliger Gedechtnis: So er gethan hat zu Wittemberg ... den 17. Januarij, im 1546. Jar :Darinnen wir für falschen Lehrern gewarnet ... werden*
- **Google Translate:** The last sermon, Doctoris Martini Lutheri, holy memory: What he did in Wittemberg... January 17th, 1546: In which we are warned... for [sic] false teachers
- **WMT19:** The Last Sermon, Doctoris Martini Lutheri, Sacred Memory: If he did at Wittemberg... January 17, 1546. Yar: In which we are warned for false teachers...
- **URL:** https://ddb.de/item/6563H62JUWEVSVTH3T7TJWCPK2NOMLK7
- **Metadata:** https://ddb.de/item/xml/6563H62JUWEVSVTH3T7TJWCPK2NOMLK7

## A.2. Example 2: The title containing Lorenz Pscherer's name.

- **Title:** *Ein frölicher Triumph Wagen/ Von der Göttlichen Offenbarung/ so durch den Engel Gottes/ der Gottfürchtige und gelerte Mann* ***Laurentius Pscherer*** *zu* ***Nürnberg*** *gehabt/ und nu mehr dem* ***7. Septembris Anno 1631****. sich glücklichen angefangen*
- **Google Translate:** A happy triumph chariot/ From the Divine Revelation/ so through the angel of God/ the God-fearing and learned man Laurentius Pscherer had at Nuremberg/ and now the 7th of September 1631. began to be happy
- **WMT19:** A devout triumph chariot / From the Divine Revelation / so had by the Angel of God / the God-fearing and learned man Laurentius Pscherer of Nuremberg / and now more the 7th of September in 1631.
- **URL:** https://ddb.de/item/WECO4OXGK3FXONM57VUUDZDOHACE4VCK
- **Metadata:** https://www.ddb.de/item/xml/WECO4OXGK3FXONM57VUUDZDOHACE4VCK

## A.3. Example 4: The Song of Songs, a lengthy title without PERSON or DATE entities.

- **Title:** *Das Hohe Lied des Königes Salomons : Wie es/ Zu der aus Gott wieder-geboren- und/ durch die Betrachtung himmlischer Dinge/ in Gott verliebten Seelen Geist-feuriger Liebes-üb- und Külung/ nach der Ordnung des Textes/ schrifftmässig erkläret gesungen; und/ mit anmutigen Kupffer- und Sinnen-Bildern ... ausgefärtiget hat*
- **Google Translate:** The Song of Songs of King Solomon: As it is sung/ To the souls who are reborn from God and/ through the contemplation of heavenly things/ in love with God, spirit-fiery love and cultivation/ according to the order of the text/ sung in scriptural terms; and/ with graceful copper and sensual images...
- **WMT19:** The Song of Solomon: As it / To the souls born again of God and / through the contemplation of heavenly things / fallen in love with God, the spirit of fiery exercise of love and cooling / as explained in writing according to the order of the text; and / with graceful copper and sensual images...
- **URL:** https://www.ddb.de/item/6PQAFR3SSP6F5OZPKSIYCRTSFWXP5CAO
- **Metadata:** https://www.ddb.de/item/xml/6PQAFR3SSP6F5OZPKSIYCRTSFWXP5CAO

## A.4. Example 5: The agent roles of Ignaz von Born and Johann Jacob Ferber.

- **Title:** *Des Hrn. Ignatz, Edl. von Born, Ritters, K.K. Berg-Raths, der Königl. Akademie der Wissenschaften zu Stockholm, der Großherzogl. zu Siena, u. der Georg. gelehrt. Gesellschaft zu Padua Mitglieds [et]c. Briefe über Mineralogische Gegenstände, auf seiner Reise durch das Temeswarer Bannat, Siebenbürgen, Ober- und Nieder-Hungarn, an den Herausgeber derselben, Johann Jacob Ferber, Mitglied der Königl. Grßherzogl. Akademie der*

*Wissenschaften zu Siena, und der Ackerbau-Gesellschaft zu Vicenza und zu Florenz, geschrieben*

- **Google Translate:** Of Mr. Ignatz, Edl. by Born, Ritters, K.K. Berg-Raths, the king. Academy of Sciences in Stockholm, the Grand Duke. to Siena, and the George. taught. Society of Padua members [et]c. Letters about mineralogical objects, on his journey through the Timisoara Bannat, Transylvania, Upper and Lower Hungary, to the editor of the same, Johann Jacob Ferber, member of the Royal. Grand Duke Academy of Sciences in Siena, and the Agricultural Society in Vicenza and Florence
- **WMT19:** Mr. Ignatz, Edl. von Born, Ritter, K.K. Berg-Raths, the Royal Academy of Sciences of Stockholm, the Grand Duchy of Siena, and Georg. erudite. Society of Padua member [et] c. Letters on mineralogical objects, on his journey through the Timisoara Bannat, Transylvania, Upper and Lower Hungary, to the editor of the same, Johann Jacob Ferber, member of the Royal Grand Ducal Academy of Sciences of Siena, and the Agricultural Society of Vicenza and Florence, written
- **URL:** https://www.ddb.de/item/H72GA6ENNNPAPYL5EYR4O2WHCJIOCJQP
- **Metadata:** https://www.ddb.de/item/xml/H72GA6ENNNPAPYL5EYR4O2WHCJIOCJQP

## B. Appendix B

### Sample approximate name matches from section 5.3

1. <ddb:2WQOAYCEL4TN4W5FB2ODYCUEX4KTSKC5> <dc:creator>: **Franz Xaver Brean**
   LLM Response:
   1. Augustine Baarfüssern
   2. Franciscus Xaverius Brean

2. <ddb:X4DLZTZUIPLDGGJURXV5EAFHJIW7YX2K> <dc:creator>: **Hendrik van Deventer**
   LLM Response:
   1. Henrici à Deventer
   2. Henrici à Deventer, Med. Doct.

3. <ddb:UT7GE4Y6URQ64QGDW7WA4HIMTUXI2OPN> <dc:creator>: **Georg Augustin Holle**
   LLM Response:
   1. Huber, Petrum
   2. Holler

4. <ddb:NJXCFR6YQKNOSYURGZ4TAZNFQ2KG7F4V> <dc:creator>: **Edward Wortley Montagu**
   LLM Response:
   1. E. W. Montagu

5. <ddb:ZJMLI4LH5R73SXZEH544TXLQ42FSP3CA> <dc:creator>: **Felix Loë**
   LLM Response:
   I. Deputy Baron Felix von Loë
   (Authors of the stenographic reports) Prussian House of Representatives reporters.)

6. <ddb:JOVL2DKWUICA6464DMGLMHHFI6VRZAL7> <dc:creator>: **Johann Hinrich Lehmann**
   LLM Response:
   1. Johann Hinrich Lehmann
   2. Nicol
   (Note: The given text seems to be written in old German, it might be a part of an obituary or a will, mentioning Johann Hinrich Lehmann and Nicol as authors)

## C. Appendix C

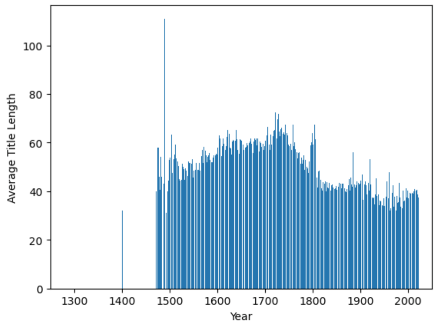


**Figure 4:** Distribution of title lengths after pruning.

# Evaluating Large Language Models for Linguistic Linked Data Generation

**Maria Pia di Buono**$^1$, **Blerina Spahiu**$^2$, **Verginica Barbu Mititelu**$^3$
$^1$University of Naples "L'Orientale", Naples, Italy
$^2$University of Milan-Bicocca, Italy,
$^3$Romanian Academy Research Institute for Artificial Intelligence, Bucharest, Romania

### Abstract

Large language models (LLMs) have revolutionized human-machine interaction with their ability to converse and perform various language tasks. This study investigates the potential of LLMs for knowledge formalization using well-defined vocabularies, specifically focusing on OntoLex-Lemon. As a preliminary exploration, we test four languages (English, Italian, Albanian, Romanian) and analyze the formalization quality of nine words with varying characteristics applying a multidimensional evaluation approach. While manual validation provided initial insights, it highlights the need for developing scalable evaluation methods for future large-scale experiments. This research aims to initiate a discussion on the potential and challenges of utilizing LLMs for knowledge formalization within the Semantic Web framework.

**Keywords:** Large Language Models, Knowledge Formalisation, Linguistic Data, Semantic Web

## 1. Introduction

The recent advancements in large language models (LLMs) like GPT-3 (Brown et al., 2020) and GPT-4 (Achiam et al., 2023), PaLM (Chowdhery et al., 2023), LLaMA (Touvron et al., 2023), etc., have highlighted the potential of deep learning techniques to facilitate natural language conversations between humans and artificial agents. Additionally, such language models are advancing quickly and they have been proven to be useful for different language-related tasks, such as question answering (Kim et al., 2023), information extraction (Dunn et al., 2022), code generation (Liu et al., 2024), etc. Most importantly, their current performance is reaching surprisingly beyond the state-of-the-art results.

However, LLMs are not without limitations. Issues like hallucination (Tonmoy et al., 2024), reliability (Huang et al., 2023), sensitivity to prompts (Qi et al., 2023), and limited context windows (Li et al., 2023), especially in free-tier models, bottleneck truly satisfactory generative tasks. To identify areas for improvement and explain these limitations, robust evaluation of LLMs is crucial, as evidenced by the growing body of research in this area. Evaluating current generative results comprehensively challenges traditional testing methods for such models.

This paper delves into whether and how effectively LLMs perform in knowledge formalization of language resources using well-defined vocabularies. The adoption of best practices and principles to describe language resources entails advantages for conveying useful linguistic information about them, allowing linking among resources, interoperability across datasets and systems, as well as their federation (Chiarcos et al., 2020).

Despite this, Linguistic Linked Data (LLD) best practices and principles seem to be far from being widely adopted. Such a situation can be related to some challenges in the creation, reusing, and exposing of LLD (Mititelu et al., 2023). Leveraging LLMs to generate formalized language resources could support the adoption of LLD principles and best practices. For this reason, we specifically focus on OntoLex-Lemon, a standard ontology for representing lexical knowledge.

In this context, the research questions we want to address are the following:

- How will this new paradigm of human-machine interaction impact established knowledge representation formalisms?
- Are LLMs ready to contribute to knowledge formalization using well-defined ontologies?
- Do these models perform consistently across different languages?

To address these questions, we conducted a preliminary study aimed at providing initial insights on the application of LLMs for generating LLD. We tested four languages: English (EN), Albanian (AL), Italian (IT), and Romanian (RO). To assess the quality of the Resource Description Framework (RDF)[1] formalization generated by LLMs, we employ a multidimensional evaluation approach. We examined nine words with diverse characteristics, including

---

[1] `https://www.w3.org/RDF/`

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 66–75
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

single words, multiword expressions, affixes, lexical entries with multiple forms, orthographic variants, conjugations, ambiguous words (polysemy), and lexicographic resources serving as both nouns and adjectives. To gain initial insights, we manually validated the LLM outputs. This approach underscores the need for developing more scalable evaluation methods for future experiments, suitable to assess both the presence of hallucinations in the general LLM outputs and the quality of the generated RDF (Section 4).

The paper is organized as follows: Section 2 delves into existing research on validating LLM outputs, providing context for our approach. Section 3 outlines the specific methodology employed to answer our research questions. Section 4 details the quality dimensions established and the corresponding metrics used to assess the quality of the generated RDF formalizations. Following this, Section 5.2 presents a thorough analysis of the obtained results. Finally, Section 6 discusses our conclusions based on the findings and outlines potential directions for future research.

## 2. Related Work

The work most relevant to ours is reported by Armaselu et al. (2023) who present preliminary results from experiments with LLMs, linked data, and semantic change in multilingual diachronic contexts. Similarly to our work, for the experiments the authors utilized the OpenAI platform for interacting with the GPT conversational agent via a user accounts. Qualitative evaluations of GPT's responses were performed, focusing on tracing semantic evolution of words like 'revolution' across different periods and languages, and providing citations when prompted. Furthermore, the model's ability to generate code based on specific word relations using OntoLex-Lemon was evaluated. Initial findings showed GPT's proficiency in generating OntoLex, but its responses related to OntoLex-FrAC, while sounding meaningful, were incorrect, likely due to insufficient training data in that formalism.

It is important to note that there are relatively few similar works in the current state-of-the-art literature. However, the rest of this section provides various methodologies for evaluating the output of LLMs. It is important to clarify that, while the generated output in our study pertains to formalizing words in OntoLex-Lemon across different languages, we draw on relevant approaches to assess the effectiveness and accuracy of the model's outputs.

Vaithilingam et al. (2022) evaluates the usability of GitHub Copilot a code generation tool empowered by LLMs through a user study with 24 participants. Participants performed programming tasks using Copilot and Intellisense, with Copilot generating code based on context and user prompts. Despite the results showing that Copilot did not consistently improve task completion time, participants preferred it for providing a starting point for tasks. Some of the results of this experiment shed light and highlighted the importance of understanding and debugging the code generated by Copilot.

Liu et al. (2024) introduces EvalPlus, a comprehensive framework designed to assess the functional correctness of code produced by LLMs. Recognizing the lack of existing frameworks for evaluating generated code, the authors put forth EvalPlus as a solution. By integrating both LLM- and mutation-based approaches, EvalPlus generates a diverse set of test inputs essential for evaluating the accuracy of code synthesized by LLMs. The evaluation involved analysing pass rates (assessing the accuracy and reliability) of LLM-generated code across various tasks and datasets.

Poesia et al. (2022) propose a framework for improving automatic code generation, which outperforms GPT-3 and Codex. The framework, called SYNCHROMESH, retrieves few-shot examples from a training bank and identifies those that are similar to the required task to be fed to the pre-trained language model. The result (the automatically generated code from natural language description) is constrained to follow the syntax of the programming language and is better than the results obtained without the use of this framework.

In the domain of automatic code generation, Perez et al. (2021) explore the possibility of automatically completing a function from initial lines of code using documentation in natural language. The used model is GPT-2, which is tuned on a corpus of Python code freely available and the reported results show that the model learns quite quickly. The authors conclude that GPT-2 treats programming languages in a manner similar to domain-specific languages.

Bareiß et al. (2022) show that few-shot learning with LLM proves effective for completing a code example or generating code snippets from scratch, sometimes even outperforming traditionally built tools. The model used is Codex, which is trained on a GitHub projects. They show that the better the prompts' design, the better the results obtained and that the descriptions of the task in natural language is also useful.

## 3. Methodology

As we want to test the possibility of leveraging LLMs in real-case scenarios, in this preliminary work we take into account the use of an easily accessible and well-known model, that is ChatGPT.

**Data Selection and Gold Standard Creation** As testing requires a gold standard to compare the ChatGPT generated answers with, we harvest several English examples from the W3C specifications page[2].

With respect to the linguistic phenomena to be investigated, we select: single word entries, multiword expressions, affixes, lexical entries with two forms (e.g., irregular plural forms), orthographic variants, conjugation, ambiguous words (i.e., polysemous words and homonyms), and lexicographic resources. For each of the aforementioned phenomena, the OntoLex-Lemon specifications provide examples of RDF formalization. The examples are extracted to have a list of linguistic realizations for prompting the model and to create a gold standard (GS) to compare the results. In total, we select eight English examples and a Latin one (the latter used for conjugation): *cat*, *African Swine Fever*, *anti-*, *child/children*, *color/colour*, *amare* (LA), *bank*, *troll*, and *animal*.

In order to create a multilingual GS suitable for a cross-language evaluation, the examples extracted from the W3C specification for the OntoLex-Lemon model are translated into Albanian, Italian, and Romanian. In some cases, adjustments (or different word choices) are required to respect the linguistic characteristics present in the original example (e.g., ambiguous words distinct in part-of-speech, gender, inflected forms or etymology). Table 1 shows the entries selected to create the GS and to input the zero-shot prompt for each of the languages.

**Prompts** For each of the entries we initially define a set of different EN prompt types and then translate these into each of the languages selected for the experiment.

The prompt types are run using the Web UI of ChatGPT, which means that the transformer is GPT-3.5.

- **Zero-shot prompt (ZSP1)** The zero-shot prompt is defined as a direct request of formalizing one of the entries from the GS word list, using the OntoLex-Lemon model.
  For AL and RO we formulate the prompt as a polite request (i.e., Could you formalize the entry [WORD] using the OntoLex-Lemon model?), as it follows:
  **AL**: *A mund të formalizoni hyrjen* [WORD] *duke përdorur modelin ontolex-lemon?*
  **RO**: *Poți formaliza intrarea* [WORD] *folosind modelul OntoLex-Lemon?*.
  For the EN and IT prompts we had to rephrase the request due to the fact that the polite question did not produce the required RDF output (see Section 5.2). Thus, for EN and IT we use an imperative clause to give the command[4], e.g., "Formalize the entry [WORD] using the OntoLex-Lemon model".
- **Zero-shot prompt with specification (ZSP2)** This type of zero-shot prompt is still a direct request of formalization without providing any example, but specifying the type of linguistic phenomenon we would like to formalize for the specific entry. For instance, for the entry African Swine Fever, we prompt the sentence "Formalize the entry American Swine fever specifying its components" to account for the subelements forming the multiword expression.
- **Few-shot prompt (FSP)** We also test the model using a few-shot prompt. In such setting, the model is provided with one example, i.e., a formalized entry from the GS, and asked to formalize a new entry. The new entries in each language, reported in Table 2, are selected on the basis of the linguistic phenomenon represented in the ones from the GS. Thus, for instance, in the few-shot setting we provide the IT GS example *uomo/uomini* (man/men) and ask to formalize the entry *bue/buoi* (ox/oxen), which present an irregular plural form.

## 4. Quality Evaluation

In evaluating the results, we adopt a multidimensional approach which takes into account the outputs from each of the prompts to assess both the general output and the RDF output quality.

**General Output** Given that the interaction with the LLM is done in a natural language, it executes the request, but also provides some commentaries (called here general output). We do not force the model to return only the RDF output, thus there is the chance that the answer contains such additional text. Indeed, we notice that in most of its answers, besides the RDF output, the model supplies an explanation of its formalization choices, which could help a user unknowledgeable of the syntax and semantics of OntoLex-Lemon to understand the use of classes and the syntax of data representation.

For monolingual outputs, when additional text is present, we evaluate some dimensions pertaining

---

[2] https://www.w3.org/2016/05/ontolex/

[4] It is worth noticing that while the direct EN prompt produces the desired RDF outcome independently of the word order, the IT prompt requires a precise word order to produce the RDF output, that is "Formalizza in OntoLex-Lemon the entry [WORD]" (Formalize in OntoLex-Lemon the entry [WORD]).

| ID | EN | AL | IT | RO |
|---|---|---|---|---|
| 1 | cat | mace | gatto | pisică |
| 2 | African Swine fever | murtaja afrikane e derrave | peste suina africana | pestă porcină africană |
| 3 | anti- | anti- | anti- | anti- |
| 4 | child/children | zot/zotërinj* | uomo/uomini* | om/oameni* |
| 5 | color/colour | sanduiç/sandwich* | skyphos/scifo* | sendviș/sandvici* |
| 6 | amare (LA) | dashuroj | amare | –[3] |
| 7 | bank | bankë | potere | sare |
| 8 | troll | akrep* | troll | trol |
| 9 | animal | kafshë | animale | animal |

Table 1: Gold Standard entries used in the zero-shot prompting. Entries marked with * do not represent the translation of EN entries, nevertheless they are representative of the same linguistic phenomenon.

| ID | EN | AL | IT | RO |
|---|---|---|---|---|
| 1 | dog | qen | cane | câine |
| 2 | prepaid credit card | kartë krediti e paguar | carta di credito prepagata | card de credit preplătit |
| 3 | pre- | para- | pre- | pre- |
| 4 | man/men | lumë/lumenj | bue/buoi* | piuă/pive* |
| 5 | center/centre |  | giovane/giovine* | cearceaf/cearșaf* |
| 6 | videre (LA) | shoh | vedere | – |
| 7 | travel | udhëtim | calcare* | vin* |
| 8 | pen | verë | botte* | limbă* |
| 8 | square | lis | rosa* | pătrat |

Table 2: Entries for the few-shot prompting. Entries marked with * do not represent the translation of EN entries, nevertheless they are representative of the same linguistic phenomenon.

to the information in the narrative part of each answer, that are: (i) completeness; (ii) correctness; (iii) consistency; (iv) interference.

- *Completeness* refers to the presence of a complete explanation for each of the formalized aspects and the relative classes/properties selected to represent them.
- *Correctness* evaluates whether the provided explanations are correct in describing the formalisation.
- *Consistency* concerns two aspects, namely (i) the extent to which the provided output adheres to what is required in the prompt and (ii) the capability of the model to be consistent across prompts and entries in the provided explanations.
- *Interference* pertains to the possibility that the output is written in more than one language. To some extent, this can be the results of some hallucinations or language bias, as well as of the way in which the model is prompted.

**RDF output** As the output of the LLM for the formalisation is in Turtle format[5], to evaluate the quality of the generated formalisation we adopted the quality metrics from Zaveri et al. (2016). Herein we list only the quality dimensions and the respective metrics that we applied in this experimental setting. The definition and the dimensions are borrowed from Zaveri et al. (2016).

- *Syntactic Validity*: the extent to which an RDF document adheres to the specifications outlined for its serialization format. The metric used for this dimension is *no malformed datatype literals*. Detecting ill-typed literals involves identifying instances where values do not adhere to the lexical syntax specified for their respective data types. This can happen if a value is either malformed or belongs to an incompatible data type.
- *Semantic Accuracy*: the extent to which data values accurately represent real-world facts. The metrics used for this dimension are (i) *no inaccurate annotations, labellings or classifications*, and (ii) *no inaccurate values*. For both metrics we manually evaluate if the classification or labelling of the entries and their values were inaccurate.
- *Interference*: the extent to which the RDF produced by the LLM mixes elements from multiple languages. This mixing (or interference) can potentially hinder the clarity and accuracy of the generated knowledge formalization. It specifically assesses the presence or absence of different languages within the same output, when the model is prompted with a question in a single language.

---

[5] `https://www.w3.org/TR/turtle/`

- *Understandability*: the clarity and absence of ambiguity in data, enabling easy comprehension and utilization by human information consumers. For this dimension, we use three metrics: (i) *human-readable labelling of classes, properties and entities as well as the presence of metadata*, (ii) *indication of one or more exemplary URIs*, and (iii) *indication of the vocabularies used in the dataset*. The first metric regards the detection of human-readable labeling of classes, properties, and entities, as well as indicating metadata (such as name, description, website) of a dataset. The second metric considers the detection of whether the pattern of the URIs is provided. Finally, the indication of the vocabularies used in the dataset can be measured by checking whether a list of vocabularies used in the formalisation is provided.
- *Interoperability* refers to the extent to which the format and structure of information conform to previously provided data as well as data from external sources. Two metrics are used for this dimension: (i) *re-use of existing terms* and (ii) *re-use of existing vocabularies*. The first metric refers to the detection of whether existing terms from all pertinent vocabularies in that specific domain have been utilized while the second evaluates the utilization of pertinent vocabularies specific to the domain in question.
- *Interpretability* concerns the technical aspects of data, encompassing whether information is represented using suitable notation and whether the data can be processed effectively by machines. For this metric we use only the *invalid usage of undefined classes and properties* metric. This metric detects the improper use of undefined classes and properties (i.e., those lacking formal definitions).

## 5. Result Analysis

In this section, we provide a result analysis for both the general output and the RDF output. Although they pertain to the data under study here and generalizations cannot be made based on these few examples, not even for the languages under study, they show what ChatGPT is able to do, as well as some of its (current) shortcomings.

### 5.1. General Output

The general output and its quality differ across languages. English and Italian do not present errors, while in some cases Albanian and Romanian sentences present some grammatical errors, mainly in the value of `rdfs:comment` and `skos:definition`. To ensure a comprehensive analysis, we firstly evaluated the *completeness* of the natural language explanations for the model's output in Albanian, Italian, and English for ZSP1 prompts. These explanations on the use of URIs, lexical entries, senses, and other relevant aspects are provided in natural language. However, for Romanian, these explanations are provided inconsistently across different entries and prompts. The natural language explanations accompanying the formalizations contain some errors in terms of *correctness*, which is observable across languages. For instance, in the IT output to the ZSP1 prompt for the entry *skyphos/scifo*, which represents two otrographical variants of the same concept, the model states that two senses have been defined. Considering the RDF output, this is correct, as two `lemon:sense`[6] have been formalized, nevertheless, the proposed senses refer to the same meaning in different languages, that are Italian and English. The provided explanation could be misleading due to the fact that it can be interpreted as a formalization of a polysemous word. For instance, the EN output to the ZSP1 for the entry *cat* contains a clarification on the use of a URI to represent a `lexicalEntry`, the way in which the canonical form and its part-of-speech are represented, and how the sense is formalized. In this case, the model does not provide information about the role of `ontolex:writtenRep`, so we consider the explanation incomplete.

As far as *consistency* is concerned, we observe that ZSP2 prompts are usually not satisfied in their specific request of formalization, mainly for some types of linguistic phenomena. This is the case when we explicitly ask to formalize a word as a lexicographic entry and the model output does not contain any lexicographic reference.

As further described in the language-specific paragraph, we also notice that language *interference* happens with Romanian explanations, which are mixed up with some Albanian words, even though the prompts for each of the languages were run at two different times, using the option 'new chat'.

Our analysis revealed several interesting patterns regarding the LLM performance on various word types used for formalization. *Single words* were generally formalized more accurately than *multiword expressions*. However, for latter, the model often struggles to identify their tag. Instead of classifying them as such, it sometimes generates irrelevant and non-existent classes.

---

[6] In evaluating this dimension in the general output, we do not assess the validity of classes/properties usage, which is evaluated according to the interpretability and semantic accuracy dimension in the RDF output evaluation.

Formalizing loan words also presents a challenge. Despite specifying the language for formalization, the model frequently defaults to English. However, the model performs well with lexical entries having both singular and plural forms, especially when prompted with some specification, as in the ZSP2 setting. This positive trend holds true across all tested languages. Similarly, the formalisation of *lexical entries with two forms* in singular and plural seems to be more accurate for the zero shot prompt with some specifications. Also this is observed across all tested languages.

*Homonyms* (i.e., words with the same spelling but different meanings) presented the most significant challenges, even though the model performs better on English. In general, it fails to distinguish between parts-of-speech, gender, inflected forms, or etymology for these entries. With *polysemous* words (having multiple meanings), the few-shot prompts lead to ambiguous formalizations. The model often misses some of the word meanings in the specific language context. Interestingly, the few-shot prompt appears to be more effective when formalizing lexical entries like nouns or adjectives.

## 5.2. RDF Output

In this subsection, we evaluate the results for each of the languages considered in our experiment. Table 3 gives an overview of evaluating the formalizations for each prompt in each language, the evaluation being made according to the criteria described in Section 4.

Some phenomena are consistent across languages and entries: e.g., the use of Lemon classes instead of OntoLex, as in `lemon:LexicalSense` instead of `ontolex:LexicalSense` and the use of some unspecified classes. Also some elements are used incorrectly, e.g., `lexinfo:Noun` and `lexinfo:Prefix`, that are defined as classes in the LexInfo ontology; however, they are written with syntax errors as if they were properties.

Furthermore, in all languages, when `rdfs:comment` and `skos:definition` are provided for an entry, they both report the same value, usually the definition of the entry.

When the request for the formalization of a word is made, it seems that there is a tendency to offer it only for one sense of the respective word, irrespective of how many it has: e.g., the word *pisică* has more meanings in Romanian (the domestic animal, as well as any of the representatives of the family Felidae). However, the formalization is presented only for the most frequent of this word's meanings, i.e., the former. Only when the request specifically mentions the polysemy of a word (see ZSP2 in Section 3) does ChatGPT offer a formalization including several senses of the respective word.

For the *anti-* entry, the LLM interprets it as a prefix for the ZSP1, while it provides a more specific type for the FSP, classifying it as affix, even though the example provided in the prompt is classified as a prefix.

**English** The analysis of the English results revealed that the LLM model struggles with assigning labels and categories accurately in all the three settings. For instance, in ZSP2, it could not distinguish between US and UK English (enUS and enGB) for words like "*color/colour*" and "*centre/center*". In other cases, there is an interference with the Italian language, that probably happens because we do not specify any information about the language of the entry that can belong to more than one language, i.e. *amare*[7], but also with an EN entry as *African Swine fever*. With reference to this type of error, we note one case, i.e., *travel*, which is affected by an interference with the German language, even though we did not run any prompt in German or use any German entry.

In ZSP1, the formalization of the verb *amare*, whose `writtenRep` is tagged as `@IT`, presents language interference as the provided definition for the `skos:definition` predicate is in English and not in Italian. In the ZSP2 results, the entry is recognized as a Latin word, nevertheless, the `writtenRep` predicate value is incorrect, i.e., *am* and *am-* instead of *amare*. Furthermore, the output contains also other incorrect information about the verb tense, mood, person, and number, that are represented respectively as present, infinitive, third person, and singular. The model performs well with the Latin verb "*videre*" (to see) in the FSP, formalizing it correctly.

Another interesting aspect pertains to the entry *travel* that presents the reference to the language specification through the use of `dct:language` and URIs for the ISO language codes. While the provided URIs are correct for English, the reference to the German language presents unresolvable URIs[8].

**Albanian** We observe that for the Albanian language, for all entries in the ZSP1, the properties used for formalisation are the same. This is not observed with the entries for the other prompts.

Another interesting pattern is that the model seems to work better with formalising singular and plural. In fact, for the ZP1, it assumes *Zot* (Gentleman) and *Zoterinj* (Gentlemen) as two distinct

---

[7] The first time we run the ZSP1, the model recognized this as a Latin word.

[8] `http://id.loc.gov/vocabulary/iso639-2/de`, `http://lexvo.org/id/iso639-1/de`

| Quality Dimension | Metrics | Prompt | EN | AL | IT | RO |
|---|---|---|---|---|---|---|
| Syntactic Validity | no malformed datatype literals | ZSP1 | 1 | 1 | 1 | 1 |
| | | ZSP2 | 1 | 1 | 1 | 1 |
| | | FSP | 1 | 1 | 1 | 0.85 |
| Semantic Accuracy | no inaccurate annotations, labellings or classifications | ZSP1 | 0.77 | 1 | 0.55 | 0.62 |
| | | ZSP2 | 0.75 | 0 | 0.62 | 0.66 |
| | | FSP | 0.88 | 0.85 | 1 | 0.62 |
| | no inaccurate values | ZSP1 | 1 | 0.55 | 1 | 0.62 |
| | | ZSP2 | 0.87 | 0.85 | 1 | 0.5 |
| | | FSP | 1 | 0.85 | 1 | 0.85 |
| Interference | no languages interference | ZSP1 | 0.77 | 0.89 | 0.44 | 0.87 |
| | | ZSP2 | 1 | 0.87 | 0.62 | 0.71 |
| | | FSP | 0.88 | 1 | 1 | 0.85 |
| Understandability | indication of one or more exemplary URIs | ZSP1 | 1 | 1 | 1 | 1 |
| | | ZSP2 | 1 | 0 | 0 | 0 |
| | | FSP | 0.22 | 0.14 | 0.14 | 0.14 |
| | indication of the vocabularies used in the dataset | ZSP1 | 1 | 1 | 1 | 1 |
| | | ZSP2 | 1 | 1 | 0 | 0 |
| | | FSP | 0 | 0.85 | 0 | 0 |
| Interoperability | re-use of existing terms | ZSP1 | 0 | 0 | 0 | 0 |
| | | ZSP2 | 0 | 0 | 0 | 0 |
| | | FSP | 0 | 0 | 0 | 0 |
| | re-use of existing vocabularies | ZSP1 | 1 | 1 | 1 | 1 |
| | | ZSP2 | 1 | 1 | 1 | 1 |
| | | FSP | 1 | 1 | 1 | 1 |
| Interpretability | invalid usage of undefined classes and properties | ZSP1 | 1 | 1 | 1 | 1 |
| | | ZSP2 | 1 | 0 | 1 | 1 |
| | | FSP | 0 | 0.85 | 0 | 1 |

Table 3: Quality Evaluation of the RDF output for each language

lexical entries, while in the ZSP2, it actually taggs these entries with their singular of plural form.

The model hallucinates more than with the other languages, for classes and predicates, e.g., lexinfo:WordMeaning, ontoloex:isA, lexinfo:FinancialInstitutionMeaning, ArthropodMeaning, etc. It also is hallucinating URIs for resources in DBpedia[9]. Moreover, especially for under-resourced languages, the model seems to do more grammatical errors. It does not follow masculine and feminine cases, singular and plural forms of adjectives, e.g., "*Një sëmundje virale e përhapur shumë e cila prek derrat e rritur dhe të egra...*", "*Një ushqim i përbërë nga një cope buke me një materiale mbushës..*".

The formalisation of the verbs also has some attributes worth to be discussed. For the verb "dashuroj" (love), ZSP1 formalises only the POS tag as a verb and provides a value for the `rdfs:comment` predicate. While the ZSP2 provides a part from the POS, and also the formalisation for its two inflections. However, these two inflections are described with the same predicates and classes, without making any distinctions with respect to the person, mood and tense of the verb. Similarly, for the entry *lis* (oak) used as a noun for the tree and as an adjective for somebody to express his/her height, for the FSP it does not follow the example provided in the prompt, but it also formalises the entry as an adjective apart from noun.

**Italian** As stated previously, in Italian, to obtain the RDF output we have to phrase the prompt as an imperative clause, as the polite form of prompting produces a narrative result without any RDF output[10], in which the model describes the linguistic characteristics that could be formalized in the OntoLex-Lemon model for that specific entry[11], e.g. the syntactic category, morphological information, etc.

With reference to the RDF output evaluation, there are not malformed datatype literals and inaccurate values, and the existing vocabularies are always re-used.

As far as the semantic accuracy of annotations, labellings or classifications is concerned, the FSP is the only one that does not present any error. In the ZSP1 and ZSP2 results, in most of the cases, the LLM fails in assigning the right language tag, in

---

[9] http://dbpedia.org/resource/VerÃń_ (pÃńrdorim_i_ndryshÃńm)

[10] The complete output is not shown due to the lack of space.

[11] It is worth stressing that also this type of results presents some errors: for instance, the model suggests to formalize both the part-of-speech and the syntactic category, which overlaps in their values.

that it applies the EN one, or it does not assign a language tag at all. This inaccuracy is probably related to some language interference between English and Italian that happens mainly in the case of loan words (e.g., coming from Greek, such as *anti* and *skyphos*). In other cases, the language interference with English has been retrieved in `rdfs:comment` and `skos:definition` in ZSP1, as well as in `writtenRep` in ZSP2. For instance, in formalizing the entry *troll* the ZSP1 result presents both the `writtenRep` and the comment/definition in English, while, for the same entry, the ZSP2 produces an Italian `writtenRep` and an English comment/definition.

We also notice that ChatGPT specifies the namespaces to indicate the vocabularies used only in the ZSP1 setting for the Italian language.

Other observations are related to the use of undefined or deprecated classes, such as in *gatto* (cat) and *peste suina africana* (African Swine fever), when `semnet` is used as reference for the entries.

In the formalization of verbs, i.e., *amare* (love) and *vedere* (see), in the ZSP1 the output does not contain any information about the conjugation or the morphological pattern, but in the FSP setting the LLM provides the correct conjugation. Nevertheless, for the verb *amare* in ZSP1 the output contains a formalization of a morphological pattern using a regular expression, that is `lemon:pattern [ lemon:regexPattern "am[a-z]*re" ]`, in which the root of the verb (*am*) is correctly recognized, while the inflectional morpheme is decomposed into one or more characters followed by *re*. This accounts for the presence of a theme vowel at the end of the stem, and also for a possible tense/mood/aspect morpheme followed by an ending that represents the morphological covariance.

**Romanian** The list of namespaces is presented only for ZSP1, never with ZSP2 or FSP, for all Romanian entries tested. When present, namespaces are never explained to the user in the general output.

Identity of form with an English word (e.g., the Romanian form *animal* is spelt identically to the English equivalent) leads to the formalization of the English word, instead of the Romanian one, in spite of formulating the request in Romanian.

With respect to the general output, interference of languages (Romanian and Albanian) happens only for FSPs, for most of them, though not for all. Even if Albanian and Romanian are languages from different language families, while Italian and Romanian are both Romance ones, it is difficult to explain why there is no interference between Romanian and Italian, but only between Romanian and Albanian. Here is an example: the boldfaced text is in Romanian, while the italic is in Albanian[12]:

**Sigur,** *po,* **poți formaliza intrarea pentru "câine" folosind modelul ontolex-lemon. Iată o formalizare posibilă:**

*Këtu*, ‘`:lex_câine`‘ *është hyrja leksikale për* "**câine**", *ndërsa* ‘`:form_câine`‘ *është forma kanonike e tij. Duke përdorur këtë formalizim, specifikoni që* "**câine**" *është një fjalë dhe specifikoni formën e shkruar të saj në gjuhën rumune*.

## 6. Conclusions

This paper provides preliminary results of the capabilities of LLMs (more specifically, ChatGPT3.5) to formalise linguistics resources using OntoLex-Lemon for four different languages. We selected 9 words from each language and asked the model to formalise it with three different prompts.

When prompted with the ZSP1, the model used the same set of properties for all entries. This could be due to overfitting on a limited training dataset or a bias towards a specific formalization style.

Another interesting result is observed in the way the model handles singular and plural forms. ZSP1 recognized them as distinct entries, while ZSP2, interestingly, attempted to capture the singular/plural information within the same entry. Additionally, the model invented fake URIs for resources within DBpedia, and new and undefined classes and properties. This "hallucination" tendency poses a serious challenge to the trustworthiness and reliability of the generated knowledge formalizations. The performance of the model in under-resourced languages (e.g., Albanian) reveals grammatical accuracy limitations, especially for noun/adjective case handling.

Despite the aforementioned limitations, the application of LLMs for generating LLD seems quite promising under the assumption of adopting specific strategies of prompting to ensure the result robustness. In the future, we plan to implement a post-generation filtering system that performs some sanity checks and adaptive prompting to improve the quality of the LLM output by identifying and correcting errors, leading to more reliable results.

## Acknowledgments

This work has been carried out within the COST Action CA 18209 European network for Web-centred linguistic data science (Nexus Linguarum).

Maria Pia di Buono has been supported by Fondo FSE/REACT-EU - Progetti DM 1062 del 10/08/2021 "Ricercatori a Tempo Determinato di tipo A) (RTDA)". Azione IV.4 - Dottorati e contratti di ricerca su tematiche dell'innovazione/Azione IV.6 - Contratti di ricerca su tematiche Green.

---

[12] We omitted the formalization for space constraints.

## 7. Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Florentina Armaselu, Christian Chiarcos, Barbara Mcgillivray, Anas Fahad Khan, Ciprian-Octavian Truică, Giedrė Valūnaitė-Oleškevičienė, Chaya Liebeskind, Elena-Simona Apostol, and Andrius Utka. 2023. Towards a conversational web? a benchmark for analysing semantic change with conversational bots and linked open data. In *LDK 2023 Conference*. NOVA CLUNL.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Patrick Bareiß, Beatriz Souza, Marcelo d'Amorim, and Michael Pradel. 2022. Code generation tools (almost) for free? a study of few-shot, pretrained language models on code. *arXiv preprint arXiv:2206.01335*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christian Chiarcos, Bettina Klimek, Christian Fäth, Thierry Declerck, and John Philip McCrae. 2020. On the Linguistic Linked Open Data infrastructure. In *Proceedings of the 1st International Workshop on Language Technology Platforms*, pages 8–15.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Trevor Cohn, Yulan He, and Yang Liu. 2020. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Alexander Dunn, John Dagdelen, Nicholas Walker, Sanghoon Lee, Andrew S Rosen, Gerbrand Ceder, Kristin Persson, and Anubhav Jain. 2022. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238*.

Association for Computing Machinery. 1983. Association for Computing Machinery. *Computing Reviews*, 24(11):503–512.

Margherita Hack. 2011. *Libera scienza in libero Stato*. Bur.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Dan Jurafsky and Christopher Manning. 2012. Natural language processing. *Instructor*, 212(998):3482.

Daniel Jurafsky and James H Martin. 2019. Speech and language processing 3rd edition draft.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2023. Sure: Improving open-domain question answering of llms via summarized retrieval. In *The Twelfth International Conference on Learning Representations*.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2024. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.

Verginica Mititelu, Maria Pia Di Buono, Hugo Gonçalo Oliveira, Blerina Spahiu, and Giedrė Valūnaitė-Oleškevičienė. 2023. Adopting linguistic linked data principles: Insights on users' experience. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 347–357.

Miguel Ortega-Martín, Óscar García-Sierra, Alfonso Ardoiz, Jorge Álvarez, Juan Carlos Armenteros, and Adrián Alonso. 2023. Linguistic ambiguity analysis in chatgpt. *arXiv preprint arXiv:2302.06426*.

Luis Perez, Lizi Ottens, and Sudharshan Viswanathan. 2021. Automatic code generation using pre-trained language models. *arXiv preprint arXiv:2102.10535*.

Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable code generation from pre-trained language models. *arXiv preprint arXiv:2201.11227*.

Shuhan Qi, Zhengying Cao, Jun Rao, Lei Wang, Jing Xiao, and Xuan Wang. 2023. What is the limitation of multimodal llms? a deeper look into multimodal llms through prompt probing. *Information Processing & Management*, 60(6):103510.

Norman C Stageberg. 1968. Structural ambiguity in the noun phrase. *TESOL Quarterly*, 2(4):232–239.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*, pages 1–7.

Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web*, 7(1):63–93.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B Tenenbaum, and Chuang Gan. 2023. Planning with large language models for code generation. *arXiv preprint arXiv:2303.05510*.

# Towards Automated Evaluation of Knowledge Encoded in Large Language Models

**Bruno Carlos Luís Ferreira, Catarina Silva, Hugo Gonçalo Oliveira**
University of Coimbra, DEI, CISUC/LASI
Coimbra, Portugal
{brunof,catarina,hroliv}@dei.uc.pt

## Abstract

Large Language Models (LLMs) have a significant user base and are gaining increasing interest and impact across various domains. Given their expanding influence, it is crucial to implement appropriate guardrails or controls to ensure ethical and responsible use. In this paper, we propose to automate the evaluation of the knowledge stored in LLMs. This is achieved by generating datasets tailored for this specific purpose, in any selected domain. Our approach consists of four major steps: (i) extraction of relevant entities; (ii) gathering of domain properties; (iii) dataset generation; and (iv) model evaluation. In order to materialize this vision, tools and resources were experimented for entity linking, knowledge acquisition, classification and prompt generation, yielding valuable insights and lessons. The generation of datasets for domain specific model evaluation has successfully proved that the approach can be a future tool for evaluating and moving LLMs “black-boxes” to human-interpretable knowledge bases.

**Keywords:** Natural Language Processing, Large Language Models, knowledge Base, Explainable Artificial Intelligence

## 1. Introduction

Nowadays, even those with minimal computer proficiency and a basic understanding of current technologies are likely aware and taking advantage of Large Language Models (LLMs). On the one hand, there are many upsides to these technologies, such as, efficiency, automation, and versatility (Strasser, 2023). On the other hand, there are many sectors of society that have reported downsides to their usage, including education (students, professors, researchers), companies (administrative work), and others (Fecher et al., 2023). Understanding the fact that humans will not go backwards, we need to address the current and future problems of such technologies.

Due to the rapid advancements and widespread acceptance of LLMs, numerous drawbacks of these technologies emerged. The well-known examples of some shortcomings are: factual errors (Wang et al., 2024), hallucinations (Ye et al., 2023), inconsistency (Elazar et al., 2021), and not being human-interpretable, i.e., “black-boxes” (Sun et al., 2022). These issues do not align with the principles of Responsible Artificial Intelligence. Also, LLMs are trained on large quantities of data that is not always easy to track, represented through opaque methods and not directly accessible. Therefore, we may add that, to some extent, LLMs do not adhere to the Findable, Accessible, Interoperable, Reusable (FAIR) principles (Wilkinson et al., 2016). Nevertheless, researchers are working to understand how to adapt FAIR guiding principles to FAIR AI models (Ravi et al., 2022).

Our objective is to contribute to more transparent and human-interpretable LLMs. Towards that vision, we propose an approach for automating the evaluation of knowledge in LLMs, across diverse domains. This process is key in our world because, as mentioned earlier, we are witnessing the widespread application of LLMs across various software, professions, and as auxiliary aid in a broad range of tasks.

The main contributions of this work are summarized as follows:

- The proposal of an end-to-end solution for automating domain-specific generation of evaluation datasets, applicable to any LLM and domain;
- An instantiation of the proposed approach with its application to two critical domains, finance and medicine.
- The evaluation of a broad range of masked language models in the previous domains, where we confirm the feasibility of the proposed approach and reveal limitations of such models when it comes to zero-shot domain knowledge acquisition.

In the remainder of this paper, we present the starting points and inspirations of our work, by describing the related work in Section 2. We proceed by detailing our general approach in Section 3. In

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 76–85
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Section 4, we instantiate the approach for two domains: financial and medical. Obtained results are further analysed, as well as challenges and limitations of the implementation. We conclude the paper in Section 6, with important takeaways and plans for future work.

## 2. Related Work

Since their early instantiations, researchers have explored Transformer Language Modes (LMs) as sources of knowledge (Petroni et al., 2019) and assessed them in tasks like relation completion, in a broad range of domains. Such an evaluation is typically supported by specifically-tailored datasets, such as LAnguage Model Analysis (LAMA), created semi-automatically from knowledge sources like Wikidata (Vrandečić and Krötzsch, 2014) and ConceptNet (Speer et al., 2017).

In the following years, various contributions adopted a similar approach (Bouraoui et al., 2020; Mickus et al., 2023; Gromann et al., 2024), i.e., probe LLMs and evaluate them on a set of knowledge sources comprised of a set of facts. Besides LAMA, other datasets were used. For instance, despite originally created for assessing static word embeddings in analogy solving, BATS (Gladkova et al., 2016) has also been used for evaluating Transformer LMs. BATS, created for English but translated to other languages (Mickus et al., 2023; Gromann et al., 2024), covers four groups of relations: inflexion morphology, derivational morphology, lexicographic semantics, encyclopedic semantics.

The goal of relation completion is to, given a subject and a predicate (relation), obtain suitable values for the object. It may resort to prompting a Transformer LMs, including BERT-based masked language models (Petroni et al., 2019; Mickus et al., 2023; Gromann et al., 2024), where the object is masked; or generative based models, including GPT-3 (Gonçalo Oliveira and Rodrigues, 2023) or BLOOM (Gromann et al., 2024), where the object is generated.

Knowledge acquisition from LMs has also raised interest from the Semantic Web community, which is confirmed by the challenge on Knowledge Base Construction from Pre-trained Language Models (LM-KBC) (Singhania et al., 2022; Kalo et al., 2023). Evaluation was based on a datasets comprising diverse world-knowledge relations (e.g., BandHasMember, FootballerPlaysPosition, PersonCauseOfDeath), each including a set of subjects and a list of ground-truth objects per subject-relation-pair.

Despite the existence of datasets like those used in the previous works, essential for evaluating LLMs, they are inherently limited. Some were created manually (Singhania et al., 2022), while others, despite involving some automatic procedure (Petroni et al., 2019), required specific planning (e.g., in the selection of relations and definition of inclusion criteria). They are created with a specific goal and, once created, remain static.

The automation of data collection from specific domains and the dynamic generation of datasets represents a promising avenue. Therefore, we propose a methodology for the automation of the creation of datasets for multiple domains that can be used for evaluating LLMs.

## 3. Proposed Approach

We see knowledge acquisition from LLM as a way towards more transparent models. While it is impossible to represent everything in a model as a single Knowledge Graph (KG), in theory, a smaller KG can be extracted on specific domains. The models will perform differently for different domains.

So, our vision involves the possibility of evaluating any model in any domain of interest. This requires specific methods for turning user-provided seeds (e.g., domain data) into relevant domain knowledge (e.g., entities, relations), and for assessing to what extent such knowledge can be obtained from the model.

We propose an approach for this vision, depicted in Figure 1. It is based on the automatic generation of datasets given a collection of textual documents on the target domain. Datasets will contain knowledge on the target domain, guided by the input collection (seeds), but effectively extracted from a human-created KG. Briefly, the proposed approach encompasses four main steps:

1. Extraction of relevant entities from domain data;
2. Gathering of domain-related entity properties from a human-created knowledge graph;
3. Generation of an evaluation dataset on domain knowledge;
4. Automation of LLMs evaluation.

More formally, from a set of entities `E` extracted from the input collection, a subset of domain-relevant `E'`$\subset$ `E` is gathered. For each entity `e` $\in$ `E'`, a set of domain properties `P` is then obtained from a knowledge graph `G`. With entities `e` and their respective properties `p` $\in$ `P`, a dataset of triples `t(e, p, o)` is finally generated.

### 3.1. Extract Relevant Entities

The set of relevant entities `E` is first extracted from the collection of textual documents. We rely on en-

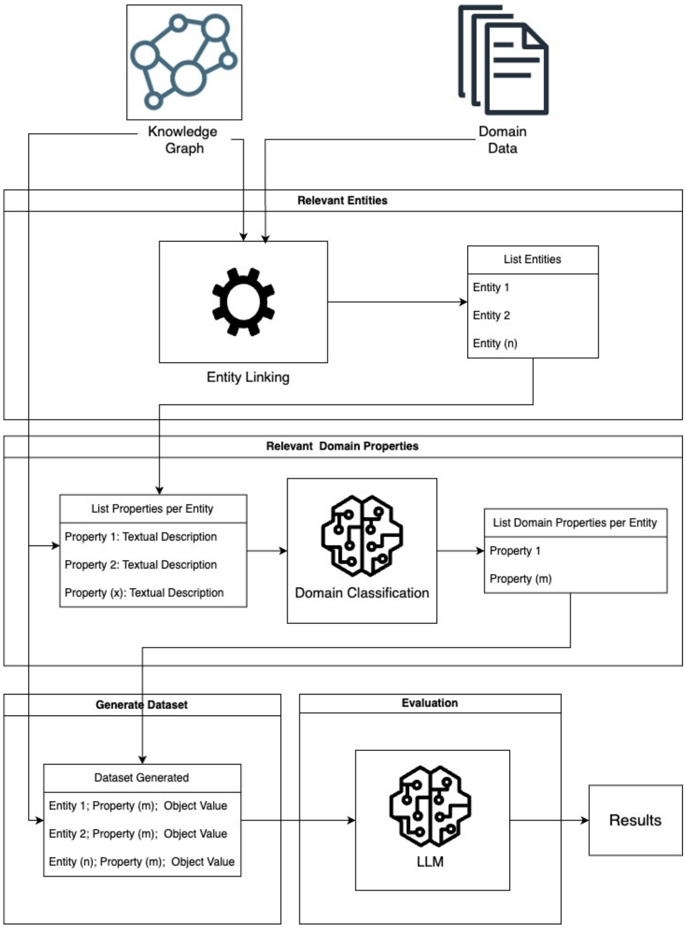

Knowledge Graph
Domain Data
Relevant Entities
Entity Linking
List Entities
Entity 1
Entity 2
Entity (n)
Relevant Domain Properties
List Properties per Entity
Property 1: Textual Description
Property 2: Textual Description
Property (x): Textual Description
Domain Classification
List Domain Properties per Entity
Property 1
Property (m)
Generate Dataset
Evaluation
Dataset Generated
Entity 1; Property (m); Object Value
Entity 2; Property (m); Object Value
Entity (n); Property (m); Object Value
LLM
Results

Figure 1: Approach for automating the evaluation of LLMs in given domains.

tity linking because we want to get ground knowledge on these entities, e.g., from a given KG.

Moreover, not all extracted entities will be relevant for the domain. So, having in mind the goal of assessing knowledge on the target domain, these should not be considered. Here, we assume that the most frequent entities in the input collection will also be the most relevant for the domain, and focus on these, i.e., extracted entities are ranked by their frequency, and only the top-`n` are used. These will constitute the set $E'$.

### 3.2. Gather Domain Properties

With each entity linked to a KG, many properties can be obtained. However, not all of them will be specific of the target domain. To consider only domain-related properties $\texttt{p} \in \texttt{P}$, we can use a text classifier trained for labelling the domain of given text. The input can be the name of the property but, if available, a longer description of the property can be used. If such classifier is not available and not enough annotated data is available for training, one can always opt for zero-shot text classification.

### 3.3. Dataset Generation

The last step is the construction of the dataset to be used for evaluation. From the each `e` → `p` pair, objects `o` are obtained from the KG, resulting in triples `(e, p, o)`. A possible triple is `(Portugal, hasCapital, Lisbon)`. The resulting dataset is a collection of such triples.

### 3.4. Automation of the Evaluation

Having a domain-oriented dataset of triples is an enabler of many evaluation possibilities where different approaches can be taken. One is follow the examples of probing, where we produce a sentence (prompt), provide it as input for a LLM, and evaluate the output of the model.

Depending on the target LLM, the creation of the prompt should be different, i.e., prompt engineering is strongly involved in this process.

## 4. Experimental setup

To materialise our vision, we experimented with different resources for entity linking, knowledge acquisition, classification and prompt generation, which resulted in an initial implementation, described in this section.

The diagram in Figure 2, instantiates the one in Figure 1, in what constitutes our first implementation. For creating a dataset, we first need to acquire domain knowledge from the provided textual data, i.e., domain-relevant entities $e$ and properties $p$. This is performed with the help of public sources of knowledge, namely DBpedia[1] (Lehmann et al., 2015) and Wikidata[2] (Vrandečić and Krötzsch, 2014), which can be queried from their respective SPARQL endpoints.

The production of this initial instance could help us identify challenges and problems, so we opted to streamline this implementation. For that, we opted to use existing applications and models in order to build and test our approach. A description of the implementation is detailed below.

## 4.1. Extract Relevant Entities

Entities were extracted from all the documents in the input collection. Since all of them should be on the target domain, we assume that the most frequently occurring entities are also the most relevant for the domain.

Entity extraction is made with the help of DBpedia Spotlight (Daiber et al., 2013), a tool for entity linking. Therefore, more than just identifying entities in text, Spotlight connects them to DBpedia resources, and thus to the Linked Open Data cloud, where knowledge about the target entities can be obtained from.

Instead of the obvious choice of using DBpedia directly, we opted for Wikidata as a Knowledge Base (KB) for being based on statements and having a community-curated ontology. This makes it, expectedly, more reliable than DBpedia, which is automatically generated from Wikipedia documents. Therefore, we must map the DBpedia entities to their Wikidata entries. For that, we relied on `owl:sameAs`, an Web Ontology Language (OWL) property that indicates that two URI references refer to the same thing in the world. Since there are `owl:sameAs` cross-links between DBpedia and Wikidata, we can use them to get the Wikidata URI corresponding to a DBPedia entity, i.e., `<Wikidata URI> owl:sameAs ?sameAsResource`.

Spotlight will extract numerous entities, but not all of them will be especially-relevant for the target domain. To get the most relevant for the domain, before mapping to Wikidata, extracted entities are ranked by frequency of occurrence in the input documents, and we focus only on the top-ranked.

## 4.2. Gather Domain Properties

The next step is to gather domain-relevant properties involving the domain entities. While it is trivial to get from Wikidata every property involving the selected entities, i.e., `<Wikidata URI> ?property ?value`, as it happens to the input text, not all properties will be relevant for the domain. However, in this case, selecting the most frequent properties will lead to many false positives, because of generic properties held by most entities. These include generic properties connecting to the entity class (e.g., *subclass of*) or to its source (e.g., *described by source*). We thus rely on a supervised classifier for discriminating between domain-relevant properties and other.

Depending on the domain, we might need to train our own classifier or resort to zero-shot learning. Yet, for many domains, state-of-the-art text classifiers are available. An example is RoBERTa-base, fine-tuned[3] in a dataset[4] based on the Human ChatGPT Comparison Corpus (HC3) (Guo et al., 2023), which classifies text in a broad range of domains.

Since the labels of some properties can be limited, whenever possible, we classify a text resulting from concatenating the property descriptions to the name of the property, i.e., `?property: ?schema:description`. Descriptions are longer and more in line with the data used for training available text classifiers. For example, for the property *retirement age* (P3001), the following text would be classified: "*retirement age: the age at which most people normally retire from work*".

The result is a set of domain-relevant triples `t(e, p, o)`. These are used for creating the dataset, where the goal would be to, given a subject (domain-relevant entity) and a property, obtain a valid object, e.g., `(Australia, retirement age, 67)`.

## 4.3. Automating the Evaluation

A possible approach for assessing an LLMs with the created dataset requires the definition of prompts. Specifically, the triple should be transformed to natural language sequences where the object is missing, to be completed by the model. Evaluation will rely on the proportion of triples for which the model completion is a valid object.

There are two types of LLMs: Generative Language Models (GLMs), where the model predicts

---

[1] https://www.dbpedia.org/

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

[3] https://huggingface.co/rajendrabaskota/hc3-wiki-domain-classification-roberta

[4] https://huggingface.co/datasets/rajendrabaskota/hc3-wiki-intro-dataset

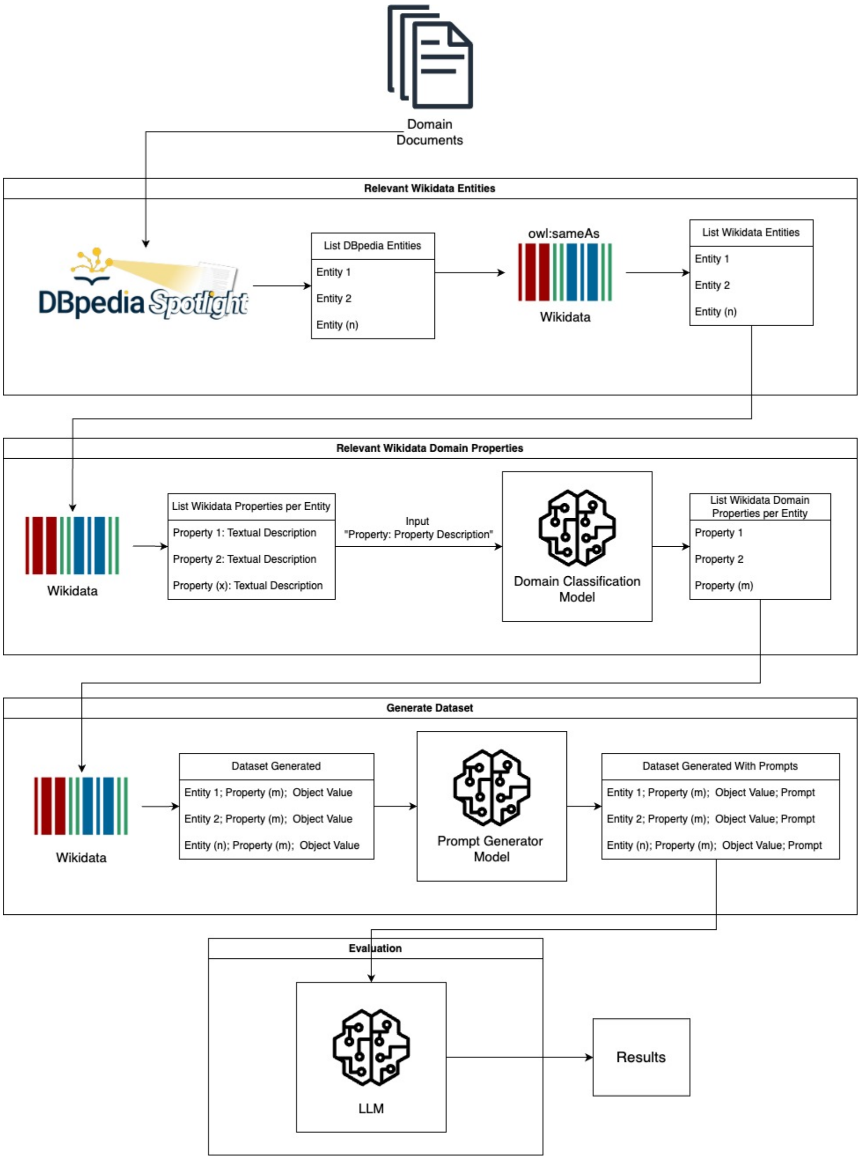

Domain Documents
Relevant Wikidata Entities
DBpedia Spotlight
List DBpedia Entities
Entity 1
Entity 2
Entity (n)
owl:sameAs
Wikidata
List Wikidata Entities
Entity 1
Entity 2
Entity (n)
Relevant Wikidata Domain Properties
Wikidata
List Wikidata Properties per Entity
Property 1: Textual Description
Property 2: Textual Description
Property (x): Textual Description
Input
"Property: Property Description"
Domain Classification Model
List Wikidata Domain Properties per Entity
Property 1
Property 2
Property (m)
Generate Dataset
Wikidata
Dataset Generated
Entity 1; Property (m); Object Value
Entity 2; Property (m); Object Value
Entity (n); Property (m); Object Value
Prompt Generator Model
Dataset Generated With Prompts
Entity 1; Property (m); Object Value; Prompt
Entity 2; Property (m); Object Value; Prompt
Entity (n); Property (m); Object Value; Prompt
Evaluation
LLM
Results

Figure 2: Representation of the instantiated implementation.

the next tokens in for a given sequence, while attending only to tokens on the left, models such as Generative Pretrained Transformer (GPT) (Radford et al., 2018) like; Masked Language Models (MLMs), where the model predicts the value of a masked token in a sequence, while attending to both the tokens in the left and right contexts, models as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) like. Depending on the type of language modelling, the prompt that interacts with the model should be different.

The experimentation reported in this work is limited to MLMs, where the boundaries of the predictions are easier to define. Having this in mind, we needed to define prompts with a masked token. For example, after replacing the object of the triple `(Australia, retirement age, 67)` by a mask, the result would be `(Australia, retirement age, [MASK])`. For simplicity, we decided to use the masked token always in place of the `object`. To provide a natural input similar to the data used for model training, i.e., natural language text, the triple is finally transformed to a prompt like “The retirement age in Australia is [MASK]”.

Creating such prompts for every single property would be tedious and a limitation of the proposed approach. Therefore, prompts are generated automatically, with the help of a 7B open generative LLM, Large Language Model Meta AI (LLaMA) 2 (Touvron et al., 2023), used in its quantized version through the ollama[5] tool. LLaMA 2 was instructed to produce a sentence based on the triple provided with the following prompt: “You are

---

[5] `https://ollama.ai/`

a model that only converts triples into sentences and nothing else. You get a triple as input, for example ['Portugal', 'currency', '[MASK]'], and you need to transform the triple into a simple human-readable sentence, for example: 'The currency of Portugal is [MASK]' or '[MASK] is the currency of Portugal' or 'Portugal has [MASK] has its currency'. Choose only the best sentence possible and return it." The option for a completely automatic generation of prompts brought pros and cons, to be discussed further ahead.

## 5. Evaluation of MLMs

We tested our implementation using two different datasets, one focused on the financial domain and the other on the medical domain. Both domains hold significant significance in our society due to their impact. This is advantageous for the analysis of our initial implementation as it enables us to comprehend (at a high level) the general knowledge of both domains, i.e., related entities and properties.

Having that into account, we can evaluate two components that resulted from our implementation:

1. The generated dataset for each domain;
2. The output predictions of the models;

In terms of the generated datasets, we can verify 1) if the `n` top entities found and 2) if the relevant entities properties extracted are relevant for each domain.

For the evaluation of the models, we decided to follow the norm and use the mean precision at `k` (`P@k`). The value is 1 if the object is ranked among the top `k` results, and 0 otherwise. We used `k = 1`, `k = 5`, and `k = 10`.

We considered a broad range pre-trained MLMs, namely, BERT (Devlin et al., 2018)[6], Robustly Optimized BERT Approach (RoBERTa) (Liu et al., 2019)[7], A distilled version of BERT (DistilBERT) (Sanh et al., 2019)[8], Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) (Clark et al., 2020)[9], and A Lite BERT (ALBERT) (Lan et al., 2019)[10].

### 5.1. Generated Datasets

In this initial implementation we aimed to use datasets as a source of "seeds", i.e., given a dataset from a specific domain it is more likely to obtain entities related to that domain.

The datasets used for both domains are publicly available in HuggingFace. For the financial domain we used a dataset[11] that contains news sentences from *Yahoo-Finance* and for the medical domain we used a dataset[12] that contains abstracts of *Pubmed* articles.

**Financial** dataset was relatively small, containing 25k small sentences where a total of 7567 distinct entities were found. We used the top 200 entities to build our dataset. In Table 1 are present the 10 more relevant entities present in the financial dataset, which in our opinion seems right. There are entities that are directly related to the financial domain, e.g., "Inflation", "European Central Bank", "Yen", which is logical, and in the other hand there are entities that, although not directly related to the domain, is very understandable there presence, e.g., "Reuters", "Chief Executive Officer", "Apple".

| Entity | Detail | QID |
|---|---|---|
| Inflation | Rise in price level over time | Q35865 |
| China (Mexico) | Municipality Location | Q942154 |
| Reuters | International News Agency | Q130879 |
| Board of Governors of the Federal Reserve System | Governing Body of the US Federal Reserve System | Q5440396 |
| Chief Executive Officer | highest-ranking corporate officer | Q484876 |
| Artificial Intelligence | field of computer science | Q11660 |
| European Central Bank | central bank of the European Union | Q8901 |
| Yen | official currency of Japan | Q8146 |
| Japan | island country in East Asia | Q17 |
| Apple | Technology Company | Q312 |

Table 1: Top 10 entities extracted from the finance dataset.

In terms of financial related properties of the entities, our implementation extracted some of the following properties present in Table 2). The table is a small subset of five of the 120 obtained properties classified as relevant for the financial domain by the domain classification model. Under analysis, not all the 120 properties obtained are relevant for the domain, but a considerable part is.

Our resulting dataset, containing the entities and their respective properties, is com-

---

[6] `https://huggingface.co/google-bert/bert-base-uncased`

[7] `https://huggingface.co/FacebookAI/roberta-base`

[8] `https://huggingface.co/distilbert/distilbert-base-uncased`

[9] `https://huggingface.co/google/electra-base-generator`

[10] `https://huggingface.co/albert/albert-base-v2`

[11] `https://huggingface.co/datasets/ugursa/Yahoo-Finance-News-Sentences`

[12] `https://huggingface.co/datasets/ccdv/pubmed-summarization`

| Relation | PID |
|---|---|
| has subsidiary | P749 |
| owner of | P1830 |
| retirement age | P3001 |
| Indeed company ID | P10285 |
| owned by | P127 |

Table 2: Relevant properties gathered for the finance domain.

posed of 1115 different triplets `(entity, property, object)`. An actual example of an extracted triple: `(Microsoft, owned by, [Bill Gates, BlackRock, The Vanguard Group])`, and from that triple the generated prompt to interact with the MLMs was “The owner of Microsoft is [MASK].”.

**Medical** dataset is considerable bigger, with 130k abstracts from *Pubmed* Papers. As the abstracts are longer than the financial news sentences, we decided not to use the entire dataset. From a total of 130k abstracts we used 9k. We obtained a total of 16791 distinct entities which indicates how much more complex is this medical dataset (compared with the financial dataset used). The following Table 3 contains the 10 more relevant entities in the 9k abstracts used.

| Entity | Detail | QID |
|---|---|---|
| Riboflavin | Chemical | Q130365 |
| Cancer | Disease | Q12078 |
| Protein | Biomolecule | Q8054 |
| Mortality Rate | Measure Deaths | Q58702 |
| Pain | Unpleasant Feeling | Q81938 |
| Metastasis | Spread of a Disease | Q181876 |
| Gene | Unit of Heredity | Q7187 |
| Obesity | Excess Body Fat | Q12174 |
| Brain | Organ | Q1073 |
| Insulin | Pancreas Hormone | Q50265665 |

Table 3: Top 10 entities extracted from the medical dataset.

In terms of relevant medical properties, the domain classification model only obtained 11 distinct properties, which is considerably less compared with the financial domain, however, all the 11 extracted are related to the medical domain.

| Relation | PID |
|---|---|
| health specialty | P1995 |
| PatientsLikeMe condition ID | P4233 |
| PatientsLikeMe treatment ID | P4235 |
| possible treatment | P924 |
| symptoms and signs | P780 |

Table 4: Relevant medical properties gathered.

The resulting dataset, containing the entities and their respective properties, is composed of 172 different triplets. An actual example of an extracted triple: `(stroke, health specialty, [neurology, neurosurgery])`, and from that triple the generated prompt was “The health specialty of stroke is [MASK].”.

## 5.2. Models Evaluation

For the prompts generated we ran the mentioned masked LLMs, obtaining as a result a set of outputs that we could compare with the ground truth of the dataset created. Follows the results that each model obtained in each domain, financial (Table 5) and medical (Tabel 6).

| Model | Acc@1 | Acc@5 | Acc@10 |
|---|---|---|---|
| ALBERT | 0.009 | 0.022 | 0.023 |
| BERT | 0.012 | 0.021 | 0.025 |
| DistilBERT | 0.009 | 0.022 | 0.025 |
| ELECTRA | 0.008 | 0.016 | 0.019 |
| RoBERTa | 0.008 | 0.030 | 0.036 |

Table 5: Accuracy results in the finance domain

| Model | Acc@1 | Acc@5 | Acc@10 |
|---|---|---|---|
| ALBERT | 0.038 | 0.077 | 0.108 |
| BERT | 0.045 | 0.089 | 0.108 |
| DistilBERT | 0.051 | 0.102 | 0.115 |
| ELECTRA | 0.006 | 0.045 | 0.096 |
| RoBERTa | 0.045 | 0.096 | 0.108 |

Table 6: Accuracy results in the medical domain

The results obtained in both domains are subpar. That occurs for a multitude of reasons, i.e., challenges and limitations that exist in our implementation.

The analysis of the outcomes, focusing on their overall precision, is impractical, as there are numerous enhancements that need to be made, mainly in two areas: the selection of domain-relevant properties, and the generation of the prompt from the extracted triples.

However, if we analyse the results by property, there are some properties that the models obtained good performance. Table 7 shows two properties from each domain that all five models got reasonable results for. There are other properties that each model performed better, but these are the two properties in the “top 10” properties of each model.

## 5.3. Limitations Analysis

The quality of the overall results in both datasets generated and the evaluation of the LLMs was decreased by some decisions we took to accelerate the process of implementation. Several times,

| | Financial | | Medical | |
|---|---|---|---|---|
| **Model** | **name** Acc@10 | **retirement age** Acc@10 | **symptoms and signs** Acc@10 | **PatientsLikeMe condition ID** Acc@10 |
| ALBERT | 0.143 | 0.125 | 0.077 | 0.231 |
| BERT | 0.286 | 0.500 | 0.385 | 0.192 |
| DistilBERT | 0.286 | 0.625 | 0.308 | 0.192 |
| ELECTRA | 0.286 | 0.750 | 0.308 | 0.154 |
| RoBERTa | 0.286 | 0.875 | 0.231 | 0.269 |

Table 7: Two of the best best performing properties in the financial and medical domains

those decisions were to apply already existing applications and available models, which was not optimal.

To acquire domain knowledge, we relied on DBpedia Spotlight for entity linking, but used Wikidata as the KB. At this stage, we encountered some challenges that we need to overcome. The initial obstacle is that employing `owl:sameAs` is not a flawless solution, as multiple entities can be identified within the same `owl:sameAs` query. We have decided to employ a second query to quantify the *inlinks* of each entity identified and select the entity with the highest number of *inlinks*. The solution is not perfect because the entity with most *inlinks* could not be the entity originally mentioned in the domain data.

To obtain the properties of the extracted entities, we employed a query for retrieving 400 properties. Since there is no built-in method for obtaining a ranking of the most relevant properties, we decided to introduce a limit to make our experiments possible. Without a limit, the query along with the domain classification task would take too long to run. Additionally, even though DBpedia Spotlight is a great resource, there are newer and better solutions for entity linking, solutions that should be considered in future work.

As mentioned previously, we relied on an existing model for the domain classification task. On the one hand, the proposed solution is robust in terms of implementation, however, not all properties were classified correctly. A future possibility would be to train a classifier for our needs, or, going in line with recent advances, use a powerful LLM with zero-shot or `n`-shot for the domain classification task.

The LLaMA 2 model was utilized for the generation of prompts from the triples obtained. This decision was judicious. However, some prompts generated did not adhere to the predetermined restrictions, thus we decided not to utilize them in the evaluation of the models. This is a significant challenge to address as we aim to automate the knowledge evaluation of LLMs.

# 6. Conclusion

LLMs have a significant impact on many sectors, jobs and tasks. They are now part of our lives and their usage will continue to grow. Given this trend, we advocate for the adoption of LLMs in a controlled manner.

We took inspiration from previous works and propose an approach for automating the evaluation of the knowledge in LLMs, based on the dynamic generation of datasets for given domains. In the paper, we describe our vision that encompasses four major steps: extraction of relevant entities, gathering domain properties, dataset generation, and automation of the evaluation.

To materialise our vision, we experimented with different resources for entity linking, knowledge acquisition, classification, and prompt generation. The result can be seen as its first implementation and constitutes important steps towards the automation of the evaluation of LLMs.

Datasets were generated for assessing the presence of knowledge in two domains, finance and medicine, and a range of MLMs were evaluated. Poor performance suggests that domain knowledge is limited in the models tested, which were trained on generic data. But we also note that the adopted zero-shot prompting approach was very straightforward, and did not go through specific prompt engineering. Moreover, the performed experimentation was useful for highlighting some limitations of the current implementation, which will be the focus of future work.

In the future, we intend to address several issues, such as the mapping of entities in DBPedia to Wikidata, which should allow for the gathering of additional domain-related properties, as well as for the generation of better prompts. For instance, we will consider entity linking tools that link directly to Wikidata, as well as the extraction of additional domain terms. We will also analyse datasets generated after different rankings on entities and properties, hopefully focusing more on the target domains, and devise balancing strategies.

The proposed approach will open the door to an easier evaluation of the knowledge in LLMs, thus contributing to faster conclusions on the suitability of different models or prompting strategies.

Therefore, we plan to take advantage of it for further evaluating different state-of-the-art models, e.g., GPT 4 (Achiam et al., 2023), Gemma (Banks and Warkentin, 2024), or to compare the performance of generic models versus models pretrained or fine-tuned in domain data. We further plan to test the impact of different prompting strategies, including few-shot prompts, and different approaches for the automatic generation of prompts.

On top of this, KGs can be created from LLMs and contribute to alternative human-interpretable representations of these “black-boxes” models. The pros and cons of each representation, or of their combination, should be further analysed, not only in terms of performance in different tasks, but also on aspects like transparency, consistency, and computational cost.

## Acknowledgements

This work was partially supported by the Portuguese Recovery and Resilience Plan (PRR) through project C645008882-00000055, Center for Responsible AI; and by FCT –Foundation for Science and Technology, I.P., within the scope of the project CISUC –UID/CEC/00326/2020 and by the European Social Fund, through the Regional Operational Program Centro 2020.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jeanine Banks and Tris Warkentin. 2024. Gemma: Introducing new state-of-the-art open models.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Benedikt Fecher, Marcel Hebing, Melissa Laufer, Jörg Pohle, and Fabian Sofsky. 2023. Friend or foe? exploring the implications of large language models on the science system. *Ai & Society*, pages 1–13.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn’t. In *Procs of NAACL 2016 Student Research Workshop*, pages 8–15. ACL.

Hugo Gonçalo Oliveira and Ricardo Rodrigues. 2023. GPT3 as a Portuguese Lexical Knowledge Base? In *Proceedings of the 4th Conference on Language, Data and Knowledge (LDK 2023), Vienna, Austria*, pages 358–363. NOVA CLUNL.

Dagmar Gromann, Hugo Gonçalo Oliveira, Lucia Pitarch, Elena-Simona Apostol, Jordi Bernad, Eliot Bytyçi, Chiara Cantone, Sara Carvalho, Francesca Frontini, Radovan Garabik, Jorge Gracia, Letizia Granata, Fahad Khan, Timotej Knez, Penny Labropoulou, Chaya Liebeskind, Maria Pia di Buono, Ana Ostroški Anić, Sigita Rackevičienė, Ricardo Rodrigues, Gilles Sérasset, Linas Selmistraitis, Mahammadou Sidibé, Purificação Silvano, Blerina Spahiu, Enriketa Sogutlu, Ranka Stanković, Ciprian-Octavian Truică, Giedrė Valūnaitė Oleškevičienė, Slavko Zitnik, and Katerina Zdravkova. 2024. MultiLexBATS: Multilingual Dataset of Lexical Semantic Relations. In *Proceedings of LREC-COLING (to appear)*. ELRA.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Jan-Christoph Kalo, Sneha Singhania, Simon Razniewski, and Jeff Z Pan. 2023. Lm-kbc 2023: 2nd challenge on knowledge base construction from pre-trained language models. In *Joint proceedings of 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC)*, volume 3577 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu

Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Timothee Mickus, Eduardo Calò, Léo Jacqmin, Denis Paperno, and Mathieu Constant. 2023. „Mann “is to “Donna” as「国王」is to « Reine » Adapting the Analogy Task for Multilingual and Contextual Embeddings. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 270–283, Toronto, Canada. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Nikil Ravi, Pranshu Chaturvedi, EA Huerta, Zhengchun Liu, Ryan Chard, Aristana Scourtas, KJ Schmidt, Kyle Chard, Ben Blaiszik, and Ian Foster. 2022. Fair principles for ai models with a practical application for accelerated high energy diffraction microscopy. *Scientific Data*, 9(1):657.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sneha Singhania, Tuan-Phong Nguyen, and Simon Razniewski. 2022. Lm-kbc: Knowledge base construction from pre-trained language models. 3274.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Anna Strasser. 2023. On pitfalls (and advantages) of sophisticated large language models. *arXiv preprint arXiv:2303.17511*.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wenxuan Wang, Juluan Shi, Zhaopeng Tu, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. The earth is flat? unveiling factual errors in large language models.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

# Self-Evaluation of Generative AI Prompts for Linguistic Linked Open Data Modelling in Diachronic Analysis

**Florentina Armaselu, Chaya Liebeskind, Giedre Valunaite Oleskeviciene**
University of Luxembourg, Jerusalem College of Technology, Mykolas Romeris University
florentina.armaselu@uni.lu, liebchaya@gmail.com, gvalunaite@mruni.eu

## Abstract

This article addresses the question of evaluating generative AI prompts designed for specific tasks such as linguistic linked open data modelling and refining of word embedding results. The prompts were created to assist the pre-modelling phase in the construction of LLODIA, a linguistic linked open data model for diachronic analysis. We present a self-evaluation framework based on the method known in literature as LLM-Eval. The discussion includes prompts related to the RDF-XML conception of the model, and neighbour list refinement, dictionary alignment and contextualisation for the term *revolution* in French, Hebrew and Lithuanian, as a proof of concept.

**Keywords:** prompt engineering, generative AI, linguistic linked open data, diachronic word embeddings

## 1. Introduction

Recent developments in large language models (LLMs), mostly originated in the transformer architecture (Vaswani et al., 2017), and generative AI (GenAI) agents that use these models to generate content based on textual prompts (HAI, 2023), have determined the emergence of prompt engineering. This new field of research refers to the design and optimisation of input prompts that guide the responses of the GenAI agents (Chen et al., 2023). In this article we address the question of how to evaluate generative AI prompts designed for specific tasks such as linguistic linked open data (LLOD) modelling and refining of word embedding results. We created a set of prompts for conversational agents GPT-3.5, GPT-4, and Microsoft Copilot (Brown et al., 2020; OpenAI, 2023; Ortiz, 2023) to assist us with the pre-modelling phase of a linguistic linked open data model for diachronic analysis (LLODIA) (Armaselu et al., 2024).[1] The prompts were intended for RDF-XML-based conception of the model, neighbour list refinement, dictionary alignment and contextualisation for the term *revolution*, as a proof of concept.

Given the GenAI agents' abilities to perform a variety of tasks and the impact of prompt attributes on the quality of the generated response, various methods and benchmarks for evaluating these prompts have been designed (Chen et al., 2023; Bach et al., 2022; Ajith et al., 2023). It is assumed that this form of assessment and AI-prompt reporting will become common practice with the increase in use of these types of agents in multiple areas of research, including LLOD. For the evaluation of our prompts we have chosen LLM-Eval (Lin and Chen, 2023) for its relative simplicity and applicability to our use case. The method consists of asking, in a single-prompt scenario, a GenAI agent to evaluate an LLM-based conversation, taking into account multiple assessment criteria, such as content, grammar, relevance and appropriateness of the dialogue response, on a 0-5 continuous evaluation scale. The main hypothesis is that the quality of the dialogue response reflects the quality of the prompts themselves. The question was how the method, applied to a selection of prompts, compared with our own assessment of the GenAI interaction results. Section 2 presents our approach, sections 3 and 4 discuss the findings and concluding remarks.

## 2. Methodology

The construction of the LLODIA model and proof of concept implied the use of static word embedding on five diachronic corpora in French, Hebrew, Latin, Lithuanian and Romanian and three main phases. (1) In the pre-modelling phase a series of prompts have been designed for GenAI conversations to model in RDF-XML a set of examples based on the French word embedding results and dictionary consultation. (2) We analysed the conversation results and compared them with existing LLOD vocabularies, knowledge repositories and models, such as Dublin Core, DBPedia and OntoLex-Lemon and FrAc (McCrae et al., 2017; Chiarcos et al., 2022). The observations were generalised, taking into account the broader LLOD context, to build and validate the classes and properties of the LLODIA model. (3) The following step involved enrichment with a proof of concept consisting of examples in all the five languages included in the project, and

---

[1] `https://github.com/nexuslinguarum/llodia/`.

*DLnLD: Deep Learning and Linked Data @LREC-COLING-2024*, pages 86–91
21 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

refinement of the model.

In this study, we focus on the pre-modelling phase that included several categories of prompts used in conversations with ChatGPT-3.5, ChatGPT-4 and Microsoft Copilot. The first category contained general questions about RDF-XML syntax, class and property generation (Copilot), or instructions asking the agent to extract examples from an OntoLex-FrAC article (Chiarcos et al., 2022) and express them in RDF-XML (ChatGPT-4). These dialogues were conceived as a preliminary task to assist the humanistic team, who had experience with XML-based formats but was less familiar with the semantic Web vocabularies and RDF-XML syntax. This was needed mainly because RDF-XML expert assistance was not available. The second category, which will be discussed in this paper, included more specific instructions for (a) RDF-XML encoding of resources such as corpora and dictionaries and embedding results, and (b) sense discrimination, dictionary alignment and contextualisation derived from the French use case on the term *révolution*. The goal was to create templates that could be used for the modelling examples in the other languages of the project. Listing 1 presents an example of prompt from task (a), when the user asks the agent to encode in RDF-XML specific resources that are represented in the model by different classes..

Listing 1: Task a/RDF-XML encoding (ChatGPT-4).

```
User: Please add the RDF−XML
    representation of the following
    resources from the categories
    dictionary and corpus. For the
    representation of languages, please
    use three letters according to the
    ISO 639−2 Code. ... The second
    dictionary has the label =
    d_wict_mul, title = Wiktionary,
    language = multilingual, publisher =
     Wikimedia Foundation ...
Chatbot: ... The resources include two
    dictionaries and one corpus. Each
    resource will be represented as an
    RDF entity with the relevant
    properties. ...
<!−− Dictionary 2: Wiktionary −−>
<rdf:Description rdf:about="https://www.
    wiktionary.org/">  ...
    <dcterms:title>Wiktionary</dcterms:
        title>
    <dcterms:publisher>Wikimedia
        Foundation</dcterms:publisher>
    <dcterms:language rdf:resource="http
        ://lexvo.org/id/iso639−3/mul"/>
        ...
    <dcterms:identifier>d_wict_mul</
        dcterms:identifier>
</rdf:Description>
```

In the final version of the LLODIA model, we replaced definitions such as `rdf:Description` shown above with the dedicated classes `llodia:Dictionary` and `llodia: Corpus`. The distinction between these types of resources was considered important since we assumed that combining corpus- and dictionary-based evidence in an explicit manner enabled richer depiction and contextualisation of semantic change phenomena.

Prompts from task (b) were intended for neighbour list refinement and dictionary sense alignment. We applied gensim word2vec (Rehurek and Sojka, 2010; Mikolov et al., 2013), 100-dimension vectors, 5-word context window, to a selection of French monographs from the National Library of Luxembourg Open Data collection,[2] with a time span from 1690 to 1918, cut into six time slices for analysis. The references used for dictionary attestations and cross-lingual relations were the CNRTL's lexical portal[3] and Wiktionary.[4] Listing 2 shows the alignment of neighbours (most similar words with the term *révolution*, computed through cosine similarity) and a corpus citation with a sense in the reference dictionary pertaining to mechanics and clock functionality.

Listing 2: Task b/dictionary alignment (ChatGPT-4).

```
User: Please associate the sub−list and
    the citation below to one of the
    senses of ... "révolution" ... Sub−list:
     moyene, tige, relier, engrennat, ...
    Citation: "La roue de longue tige ou
     grande moyene fait une révolution
    par heure ..." Please output your
    response in a table with 3 columns,
    Sub−list, Sense, Confidence, ... and
    your confidence level for this
    association.
Chatbot: Based on the ... analysis of the
    senses of "révolution" ... the sub−
    list ... and the citation relate to
    the mechanical or technical sense ...
    This sense aligns with the
    description ... within a device,
    likely a clock ... Therefore, the
    associated sense ... is related to the
     mechanical or movement aspects of
    "révolution." ...
```

The GenAI prompts were designed to refine the resulting neighbour lists, by selecting for the proof of concept the neighbours that could more likely be associated to a certain dictionary sense or by splitting the lists to discriminate between different dictionary senses.[5] We used ChatGPT-4, uploaded

---

[2] Bibliothèque nationale du Luxembourg (BnL) Open Data MONOGRAPH TEXT-PACK: `https://data.bnl.lu/data/historical-newspapers/`.

[3] `https://www.cnrtl.fr/portail/`.

[4] `https://www.wiktionary.org/`.

[5] We tried to address the "meaning conflation" issue of

a CNRTL dictionary file for the term *révolution* and asked the agent to align the neighbour lists and sublists, and associated corpus citations, to the senses provided in the file. Therefore, we could identify and link neighbours and citations from the corpus segments and time intervals to various senses of *révolution* and domains of knowledge. For the French corpus, these senses corresponded to (1) mechanics, circular motion of a body around its axis, for the time slice 1690-1794 (AI agent's confidence 95%); (2) geometry, motion of a geometric form around an axis, for 1831-1866 (95%); (3) geophysics, natural phenomena changing the physical characteristics of the Earth, and (4) politics, sudden overthrow of the political regime of a nation, for 1867-1889 (95%, 90%), and (5) the French Revolution, for the segment 1890-1918 (95%). Similar prompts for sense discrimination and refinement of neighbour lists, or contextual enrichment (task b) were devised for the proof of concept examples and experiments in the other languages, as discussed below for Hebrew and Lithuanian. Additional prompt examples are presented in table 2.

We utilised the gensim word2vec model (100-dimension vectors, 5-word context window) to extract neighboring words from the Responsa[6] dataset (Liebeskind and Liebeskind, 2020) for our generative AI studies in Hebrew. The term מהפכה (revolution) is present in three eras of the corpus (first, third and fourth). For each period, we supplied ChatGPT-3.5 with a list of neighboring words and requested it to determine the meaning of the given list. Next, we requested ChatGPT-3.5 to align its assignments with one of the three senses from Wictionary or a fourth sense given by Milog[7], and to indicate the level of confidence in its assignment.

For the experiments with generative AI in Lithuanian we asked ChatGPT 3.5 to determine the neighbor words related to the senses of the target word "revoliucija" in Lithuanian. We also asked to provide a short description for each assigned sense of the target word "revolution" in Lithuanian and attach a degree of confidence to it expressed by percent. Then we asked to provide a time slice of usage for each assigned sense of the target word "revolution" in Lithuanian and attach a degree of confidence to it. Finally, we wanted to find out the first mention of the target word "revoliucija" in Lithuanian.

For evaluation, we applied the LLM-Eval method to a selection of dialogues with GenAI agents from the pre-modelling phase, tasks (a) and (b). The GenAI agents used in evaluation were ChatGPT-4 and Gemini (Team et al., 2023) and the evaluated agents were ChatGPT-3.5 and ChatGPT-4. The dialogues were attached as PDF files to the conversations with ChatGPT-4, and directly inserted into the prompts for Gemini. We followed the LLM-Eval model for the evaluation of the chatbots' response according to four criteria (appropriateness, content, grammar and relevance) on a scale from 0.0 to 5.0, to which we added the evaluators' confidence in their assessment (percentage), as shown in listing 3. In line with LLM-Eval, it was assumed that higher scores reflect higher prompt quality.

Listing 3: LLM-Eval prompt (ChatGPT-4).

```
User: Please score the chatbot response
    from the attached file ... on a
    continuous scale from 0.0 to 5.0.
    The criteria to be evaluated are:
    appropriateness, content, grammar
    and relevance. The output will be a
    table with columns for the four
    criteria and an additional column
    for your confidence level on the
    assessment (in percentage).
```

## 3. Results and Discussion

Table 1 shows the results of the evaluation for 10 dialogues, 4 from the category RDF-XML encoding (task a) and 6 (2 for each of the 3 languages) from the categories neighbour list refinement, dictionary sense alignment and contextualisation (task b).

In general, the agents assigned higher scores and confidence to the dialogues from the category RDF-XML encoding (task a), although in some cases, especially related to OntoLex-FrAC, the namespace or some properties were not always accurate. The criterion with highest score was grammar, which is not surprising given the training characteristics of LLMs. Slightly lower scores or confidence were observed for Dialogue 6 (ChatGPT-4), 7 (Gemini) and 8 (both agents) (task b). ChatGPT-4 explained the slight deductions for appropriateness and content in Dialogue 6 (French), designed for neighbour list splitting and alignment with dictionary senses, as due to the "challenges in categorizing words without additional context and verifying the precision of these categorizations against the document." For dialogue 7 (Hebrew), Gemini assigned a surprising score of 0.0 with confidence 10% for relevance, which is justified by the fact that the "core functionality of the chatbot (understanding Hebrew text) is not applicable to the user's request." For dialogue 8 (Hebrew), both ChatGPT-4 and Gemini assigned a lower score to content, the former explaining the deduction for the "assumption that the categories are exhaustive or perfectly accurate", while the latter referred to the fact that "Sense 3 ("chaos or disorder") could be further refined." For dialogue 9 (Lithuanian), a lower score

---

static word embedding (Camacho-Collados and Pilehvar, 2018, pp. 5-6), i.e., the word vectors and neighbours may refer to different meanings of the target word.

[6] https://www.responsa.co.il/.

[7] https://milog.co.il/.

|  | ***ChatGPT-4*** |  |  |  | ***Gemini*** |  |  |  |
|---|---|---|---|---|---|---|---|---|
| **Dial.** | **App.(C%)** | **Cnt.(C%)** | **Grm.(C%)** | **Rel.(C%)** | **App.(C%)** | **Cnt.(C%)** | **Grm.(C%)** | **Rel.(C%)** |
| Dial1 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial2 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial3 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial4 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial5 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial6 | 4.5 (90) | 4.0 (80) | 5.0 (100) | 4.5 (90) | 5.0 (95) | 5.0 (90) | 5.0 (100) | 5.0 (95) |
| Dial7 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 4.5 (80) | 5.0 (90) | 5.0 (95) | 0.0 (10) |
| Dial8 | 4.5 (90) | 4.0 (85) | 5.0 (100) | 4.5 (90) | 5.0 (95) | 4.5 (85) | 5.0 (99) | 4.0 (80) |
| Dial9 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (90) | 4.5 (80) | 5.0 (95) | 4.0 (85) |
| Dial10 | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (100) | 5.0 (95) | 5.0 (95) | 5.0 (95) | 5.0 (95) |

Table 1: Dialogue response evaluation scores (0.0 to 5.0) for appropriateness, content, grammar, relevance, and confidence (%) mentioned in brackets after each score; task a: dialogues 1-4 (ChatGPT-4), task b: dialogues 5, 6 (French, ChatGPT-4), 7, 8 (Hebrew, ChatGPT-3.5), 9, 10 (Lithuanian, ChatGPT-3.5)

was assigned by Gemini to relevance with the observation that the concept of ”neighbor words” should be considered together with the assumption that ”words might appear as neighbors depending on the specific context.” Table 3 presents additional excerpts of verbal assessment by the two agents for some of the dialogues discussed in this section. Generally, Gemini's explanations on scores and confidence levels seemed a bit more nuanced than ChatGPT-4's, this also possibly due to the slightly lower values assigned by it, which required more detailed explanations.

While the LLM-Eval experiments produced relatively high scores in the evaluation of the dialogue responses, which may be interpreted as an indicator of good prompting quality, it should be noted that they included only simple extracts from the dialogues (one dialogue turn, User prompt - Chatbot response). Our interactions with the GenAI agents involved longer conversations, step by step information addition and clarifications. The experiments for French with ChatGPT-4 showed that defining precise pieces of information to be encoded for the RDF-XML generation (task a), and providing neighbour lists, citations and the source with the dictionary senses for task (b), can produce good results. However, the generated RDF-XML code had to be checked and refined and in the case of sense discrimination based on the uploaded file with dictionary information, the agent needed to be recalled from time to time to use that file and not the senses that it could derive from its own pretraining. For the other two languages, a different GPT version was used in the experiments (task b).

ChatGPT-3.5 successfully determined the meaning of the term מהפכה across multiple periods by analyzing the neighboring words. The first period (11th century until the end of the 15th century) was designated with a confidence level of 80% as representing ”Social or Moral Upheaval.” The third period (the 17th through the 19th centuries) was designated as representing ”Societal Disintegration or Degradation” with a confidence level of 85%, while the fourth period (the 20th century until the present day) was identified as representing ”Technological or Medical Revolution” with a confidence level of 75%. Nevertheless, when we requested ChatGPT-3.5 to synchronize its assignments with a specific sense from the dictionary, it inadvertently combined the several senses together. The first period was aligned with the sense of ”A historical event that significantly altered the trajectory of a specific nation or the course of human civilization as a whole” with 80% confidence. The third period was aligned with the sense of ”Chaos, commotion, a state of evident disarray” with 70% confidence. The fourth period was aligned with the sense of ”Full restoration, altering the current arrangement and routine” with 60% confidence. When we requested ChatGPT-3.5 to carefully examine an alternative interpretation of the word that we deemed more appropriate, it displayed an unwillingness to alter its perspective.

We analyzed two citations for each period. One example from the first period was erroneously attributed (with a confidence level of 90%) to the מהפכה Biblical meaning of destruction. The ancient style of the citation was the reason for this, as it had no connection to destruction. The third period corresponds to the historical periods of the French corpus, since it represents the violent attacks and persecutions that Jews endured throughout this time. Both citations pertain to the French Revolution. The first citation was attributed to the meaning of ”A historical event that significantly altered the trajectory of a specific nation or the course of human civilization as a whole” with a confidence level of 85%. On the other hand, the second citation was assigned to the meaning of ”chaos, commotion, a state of evident disarray” with a confidence level of 90%, despite the explicit mention of the French revolution in the text. [8] The citations from

---

[8] ”ברחתי עם כל אשר לי בעת המהפכה כאוד מוצל מאש שריפת

the fourth period relate to two distinct revolutions: the Humanist revolution and a spiritual revolution. The first citation was assigned the meaning of ”Full restoration, altering the current arrangement and routines” with a confidence level of 80%. On the other hand, the second citation was assigned the meaning of a historical event with a confidence level of 85%, which is somewhat confusing. When we asked ChatGPT-3.5 to separate the list of each period into sub-lists and assign to each sub-list the most likely sense of the word מהפכה , for the first and third periods we got a mixture of senses. However, all the words in the list for the fourth period were assigned the sense of ”Full restoration, altering the current arrangement and routines” with varying levels of confidence.[9]

The Generative AI agent (ChatGPT-3.5) identified the neighboring words which provide a sense of the various contexts in which the word ”revoliucija” can be used in Lithuanian. The contexts included political, social, cultural, technological, scientific, industrial, and economic senses identified with high confidence of 90%. However, the agent could not provide dictionary attestations and identify when the target word ”revoliucija” was first mentioned in Lithuanian sources as it does not have access to the specific dictionaries and relies only on the data used to train it and its ability to generate language-based responses.

## 4. Conclusion and future work

In this article we discussed various forms of prompting and interaction with GenAI agents, to automate or assist in LLOD generation, in tasks that required RDF-XML modelling and refinement of word embedding results for diachronic analysis. Our qualitative evaluation and preliminary testing with the LLM-Eval method showed that the integration of generative AI agents into LLOD workflows can be informed by techniques from the emerging field of prompt engineering and its new ways of reflecting on how we communicate with technology. According to this type of evaluation, prompting for RDF-XML generation (task a) seems to produce more stable results, while sense alignment and contextualisation (task b) may be more influenced by nuances in defining relevant concepts, such as neighbour and context. Further work is intended to explore in more depth how prompting in the evaluation method itself can elicit subtler assessment statements and fine tuning in assessing the linguistic modelling and production of LLOD encoding elements within GenAI-assisted processes.

## 5. Acknowledgments

This article is based upon work from COST Action *Nexus Linguarum, European network for Web-centred linguistic data science*, supported by COST (European Cooperation in Science and Technology). `www.cost.eu`.

## 6. Authors’ contribution

F.A., sections 1, 2 and 3 (general, LLM-Eval, French), 4; C.L., sections 2 and 3 (Hebrew); G.V.O., sections 2 and 3 (Lithuanian). All the authors critically revised the final version of the manuscript.

## 7. Bibliographical References

### References

Anirudh Ajith, Chris Pan, Mengzhou Xia, Ameet Deshpande, and Karthik Narasimhan. 2023. Instructeval: Systematic evaluation of instruction selection methods. (arXiv:2307.00259). ArXiv:2307.00259 [cs].

Florentina Armaselu, Chaya Liebeskind, Paola Marongiu, Barbara McGillivray, Giedrė Valūnaitė Oleškevičienė, Elena Simona Apostol, and Ciprian-Octavian Truică. 2024. Linguistic Linked Open Data for Diachronic Analysis (LLODIA).

Stephen Bach, Victor Sanh, Zheng Xin Yong, and al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, page 93–104, Dublin, Ireland. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, and al. 2020. Language models are few-shot learners. (arXiv:2005.14165). ArXiv:2005.14165 [cs].

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, and al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. (arXiv:2303.12712). ArXiv:2303.12712 [cs].

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings:

---

הצרפתים...″ (I fled with all my possessions during the revolution, closely pursued by the flames ignited by the French).

[9] We inquired ChatGPT-3.5 about the reason for not assigning the list of the third era to sense1, given that the words in the list pertain to the medical or industrial revolution. It answered: ”You’re absolutely correct, and I appreciate your point. Upon reevaluation, List 3 could indeed describe developments related to medical or industrial revolutions”. Consequently, the sense assignment of the words in the list were properly modified.

A survey on vector representations of meaning. (arXiv:1805.04032). ArXiv:1805.04032 [cs].

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. (arXiv:2310.14735). ArXiv:2310.14735 [cs].

Christian Chiarcos, Elena-Simona Apostol, Besim Kabashi, and Ciprian-Octavian Truică. 2022. Modelling frequency, attestation, and corpus-based information with OntoLex-FrAC. In *International Conference on Computational Linguistics*, pages 4018–4027.

Stanford HAI. 2023. *Generative AI: Perspectives from Stanford HAI*.

Chaya Liebeskind and Shmuel Liebeskind. 2020. Deep learning for period classification of historical Hebrew texts. *Journal of Data Mining & Digital Humanities*, 2020:5864.

Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-Eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, page 47–58, Toronto, Canada. Association for Computational Linguistics.

John P McCrae, Julia Bosque-Gil, Jorge Gracia, and Paul Buitelaar. 2017. The OntoLex-Lemon model: development and applications. In *Proceedings of eLex 2017 Conference*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781 [cs]*. ArXiv: 1301.3781.

OpenAI. 2023. *GPT-4 Technical Report*.

Sabrina Ortiz. 2023. What are Microsoft's different Copilots? Here's what they are and how you can use them.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, page 45–50, Valletta, Malta. ELRA.

Gemini Team et al. 2023. Gemini: A family of highly capable multimodal models. (arXiv:2312.11805). ArXiv:2312.11805 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. (arXiv:1706.03762). ArXiv:1706.03762 [cs].

# 8. Appendix

| **Task/Dial** | **Prompt** | **Agent** |
|---|---|---|
| a/Dial2 | Please add to the RDF-XML representation ..., using OntoLex and OntoLex-FrAC, a lexical entry for the French word "révolution" with the label = ..., canonical form and writtenRep = révolution, and part of speech = noun. | Chat GPT-4 |
| b/Dial6 | Please extract the senses of the French word "révolution" from the attached file ... You have to separate the list of neighbours (most similar words) below into sub-lists that can most likely correspond to these senses ... | Chat GPT-4 |
| b/Dial8 | Please try to separate list1 into sub-lists and assign to each sub-list the most likely sense of the word "מהפכה" . List 1: 1. מהטעות (mahata'ut) - Mistake/Error 2. החיסרון (hakhisaron) - Deficiency ... | Chat GPT-3.5 |
| b/Dial9 | Please determine to which senses of the target word "revoliucija" in Lithuanian could be [related] the neighbor words. | Chat GPT-3.5 |

Table 2: Prompts by task, dialogue and agent

| **Task/Dial** | **Verbal assessment** | **Agent** |
|---|---|---|
| a/Dial2 | This evaluation reflects the chatbot's ability to ... respond to a specific RDF-XML modification request, demonstrating a high level of appropriateness, content accuracy, grammatical correctness, and relevance to the user's needs. | Chat GPT-4 |
| a/Dial2 | This is another exceptional response that demonstrates a strong understanding of the user's query and ... a well-explained solution. The response includes the requested RDF-XML snippet showcasing the new lexical entry. | Gemini |
| b/Dial6 | Confidence Level: The chatbot's assignment of confidence levels to each categorization reflects a thoughtful analysis, although the actual accuracy of these assignments depends on the detailed content of the document [which comprised the dictionary senses used in responding to the prompts, not available in the evaluation]. | Chat GPT-4 |
| b/Dial6 | My confidence level in scoring this response remains high ... The response is factually correct ... While a human expert on historical French documents might be able to refine some of the categorizations ..., the response overall is comprehensive and accurate. | Gemini |
| b/Dial8 | Relevance (4.5): The chatbot's response directly addresses the user's request, providing a structured and meaningful classification of the terms ... The slight deduction accounts for the inherent limitations of the chatbot in comprehensively capturing and interpreting all possible nuances without more extensive context or clarification from the user. | Chat GPT-4 |
| b/Dial8 | Relevance (4.0): The response is highly relevant to the user's query. However, a slight deduction is made because Sense 3 could benefit from a more nuanced explanation specific to the religious context. | Gemini |
| b/Dial9 | Content: The chatbot identifies a broad spectrum of revolution senses, including political, social, technological, cultural, scientific, industrial, and economic revolutions. For each category, it provides specific Lithuanian terms ... This comprehensive categorization reflects a deep and nuanced understanding of the word's potential meanings and applications. | Chat GPT-4 |
| b/Dial9 | Content (4.5, 80% confidence): The response offers a comprehensive overview of various revolution types (political, social, technological, etc.) It includes ... relevant neighbor words ... While all the revolution types might not be neighbors in every context, it provides a good starting point. | Gemini |

Table 3: Assessment by task, dialogue and agent

# Author Index

Armaselu, Florentina, 86

Barbu Mititelu, Verginica, 66
Blum, Moritz, 1

Cartier, Emmanuel, 31
Cimiano, Philipp, 1

da Silva, Viviane Torres, 45
di Buono, Maria Pia, 66

Ell, Basil, 1

Ferreira, Bruno Carlos Luís, 76
Fiorini, Sandro Rama, 45

Gonçalo Oliveira, Hugo, 76

Jiang, Shufan, 53

Liebeskind, Chaya, 86
Lima, Guilherme, 45

Mosteiro, Pablo, 14

Nolano, Gennaro, 1

Peetermans, Emile, 31

Rademaker, Alexandre, 45

Sack, Harald, 53
Sarhan, Injy, 14
Silva, Catarina, 76
Spahiu, Blerina, 66

Tan, Mary Ann, 53
Toth, Bendegúz, 14

Valunaite Oleskeviciene, Giedre, 86

Wang, Shihan, 14