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.

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, pre-trained 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*.