Elena Volkanovska

A Study of Errors in the Output of Large Language Models for Domain-Specific Few-Shot Named Entity Recognition

Abstract

This paper proposes an error classification framework for a comprehensive analysis of the output that large language models (LLMs) generate in a few-shot named entity recognition (NER) task in a specialised domain. The framework should be seen as an exploratory analysis complementary to established performance metrics for NER classifiers, such as F1 score, as it accounts for outcomes possible in a few-shot, LLM-based NER task. By categorising and assessing incorrect named entity predictions quantitatively, the paper shows how the proposed error classification could support a deeper cross-model and cross-prompt performance comparison, alongside a roadmap for a guided qualitative error analysis.

1 Introduction

The advent of generative large language models (LLMs) created an increased interest in experimenting with few-shot methods for named entity recognition (NER). With LLMs, NER can be defined as a question-answering task, where a model is prompted to identify[1] named entities based on a named entity definition and named entity examples provided in the prompt. In real-world scenarios, the need for few-shot NER is driven by scarcity of resources, legal constraints for sharing annotated data, and the cost of annotation (Moscato, Postiglione, & Sperlí, 2023). However, the success of few-shot NER techniques is not consistent. Some studies using known NER datasets and LLMs have reported promising results (Ashok & Lipton, 2023; Epure & Hennequin, 2022; Wang et al., 2023).[2] At the same time, experiments using more specialised NER datasets, such as the one described in Section 3, do not achieve the same degree of success. Moscato et al. (2023) also mention that the success of few-shot NER in real-world deployment scenarios is yet to be proven.

This study investigates the possible causes of such inconsistencies by analysing LLMs' output in experiments that yielded F1 scores that were substantially below the task baseline. Rather than discarding the output as noise, the paper aims to identify what

[1] An effort was made to refrain from using anthropomorphising terms when describing LLMs (see Inie, Druga, Zukerman, and Bender (2024) for more information on this topic); nevertheless, this type of language is common in the context of generative language models and, in some cases, difficult to evade.

[2] Some authors acknowledge that data contamination i.e. the likelihood of the used LLMs having been previously exposed to the NER datasets might affect the outcome.

lessons can be learned by proposing a draft framework for a descriptive error analysis. To do so, the study first reviews existing approaches to error analysis in few-shot NER in Section 2, followed by a brief description of the experiments underpinning the analysed data in Section 3. The proposed error classification and the insights it provides into LLM performance are discussed in Sections 4 and 5 respectively.

2 Related Work

Generative pre-trained language models employed in some studies exploring few-shot methods for in- and cross-domain NER include the Pretrained Conditional Generation Model of Flan-T5-XXL (11B) (Chung et al., 2024), GPT-3.5 (Brown et al., 2020), and GPT4 (Achiam et al., 2023), all of which have been used in the study by Ashok and Lipton (2023); GPT-3 (davinci-003) used by Wang et al. (2023), and a medium-sized GPT-2 model used in few-shot NER experiments by Epure and Hennequin (2022).

These studies showed that the named entities (NEs) identified by LLMs can lead to valuable insights. Ashok and Lipton (2023) conduct a human survey of errors, where they (1) create a list containing 20 randomly selected examples of predicted named entity instances, (2) create a ground truth list containing NEs from the same sentences used to create list (1), and (3) ask three different human annotators to evaluate each entity of lists (1) and (2). The human annotators are given a definition of the NER problem relevant to the dataset from which the lists are created. The results from this evaluation show that many of the predictions could be acceptable NE candidates and were not considered errors by the human annotators.

The evaluation approach adopted by Epure and Hennequin (2022) for NER in a few-shot setting is case-insensitive and accommodates for output where the model generates an NE with a different spelling or when it fails to follow the instruction for sentences containing no entities. The study dubs as *confusion patterns* cases when the LM fails to generate the correct entity type, conflating, for example, *corporation* or *group* with *location*. The study's authors provide a brief overview of NE categories that perform well and categories that do not. Wang et al. (2023) also find that the LLM conflates *location* and *geographical entities* in a nested NER scenario.

While it is evident that language models' output is manually inspected, with researchers working in few-shot NER performing an error analysis in order to compare the effects of various prompt designs and task requirements, the insights that come from the manual inspection are mostly captured in the recommendations for prompt design in future studies. In other words, such analyses have not amounted to a systematic classification of errors identified in models' output.

Contribution This paper proposes a descriptive error analysis method for LLM output in a few-shot NER task on two domain-specific NER datasets. It combines categories from existing NER evaluation metrics, such as F1 scores, and error analyses encountered in previous studies on few-shot NER into a single error classification framework for model output analysis. This framework could be used to (1) gauge weak points in the task design and in the LLMs' performance and (2) make informed decisions for qualitative error analysis and iterative changes to the prompt design.

3 Data

LLMs and datasets The data analysed in this study is the LLM output from a series of few-shot NER experiments, where 7762 prompts are run on four LLMs: OpenAI's gpt-4o-2024-05-13 and gpt-4o-mini (hereinafter: gpt-4o and gpt-4o-mini), and Meta's Meta-Llama-3.1-70B-Instruct and Meta-Llama-3.1-405B-Instruct (hereinafter: Llama-70B and Llama-405B). The experiments are conducted on the test data splits of two NER datasets comprising scientific texts: Climate-Change-NER (Bhattacharjee et al., 2024) with 13 climate-change-relevant NE categories (*climate-assets*, *climate-datasets*, *climate-greenhouse-gases*, *climate-hazards*, *climate-impacts*, *climate-mitigations*, *climate-models*, *climate-nature*, *climate-observations*, *climate-organisms*, *climate-organizations*, *climate-problem-origins*, and *climate-properties*), and BiodivNER (Abdelmageed et al., 2022) with 6 biodiversity-relevant NE categories (*organism*, *phenomena*, *matter*, *environment*, *quality*, and *location*). The LLMs' output and dataset information are available in a dedicated GitHub repository.[3]

Prompts The rationale behind the prompting methodology, the prompt design, and the results for each prompt and language model are described in detail in Volkanovska (2025). The prompt design was inspired by the study of Ashok and Lipton (2023), with the final promts differing in three major ways: (1) the input/output requirement (either a Python string or a tokenized sentence i.e. a Python list of word-based tokens and their indices), (2) the number of NE categories tested, and (3) the method of selecting task examples (TEs) in the prompt. Under (1), the prompts can have either *string-based* or *token-based* input (TEs) and output (a requirement for the model to generate an answer in a format that corresponds to the TEs). Under (2), there are *full prompts*, where models are tested on the complete set of NE categories, and *cluster prompts*, where the models are tested on subgroups of NE categories.

The category *full prompts* contains 6 prompt versions, which differ in the number of TEs provided to the model (3, 4 or 5). Regarding *cluster prompts*, named entities are divided into clusters of categories. For Climate-Change-NER, the clusters are: (1) *climate-hazards*, *climate-problem-origins*, *climate-greenhouse-gases*; (2) *climate-impacts*, *climate-assets*, *climate-nature*, *climate-organisms*; (3) *climate-datasets*, *climate-models*, *climate-observations*, *climate-properties*, and (4) *climate-mitigations*, *climate-organisations*. For BiodivNER, the three clusters are: (1) *environment*, *location*; (2) *organism*, *matter*, and (3) *phenomena*, *quality*. Finally, under (3), TEs contained either randomly selected sentences from the train data split, or sentences with a high semantic similarity score to the sentence the model was to annotate. Semantic similarity scores were calculated with the library sentence-transformers (Reimers & Gurevych, 2019) and the model *sentence-transformers/stsbdistilroberta-base-v2*.

[3] https://github.com/volkanovska/NER-annotation-with-LLMs

The different prompting scenarios showed that token-based prompts performed, on average, slightly better than string-based prompts. For the former, LLMs' averaged F1 scores[4] ranged between 0.27 (lowest) and 0.41 (highest). For string-based prompts, the averaged F1 scores ranged from 0.28 to 0.39. LLMs generally performed better when there were more TEs, while the TEs' similarity to the task sentence had a greater impact on the result when the original dataset contained some noise, most likely introduced by text extraction from PDF sources. As token-based prompts performed slightly better than string-based prompts, the error analysis proposed in this paper is conducted on the output from token-based prompts. See Appendix 7 for a prompt example.

4 Methodology

In the context of this study, *error* encompasses all instances where the model's output does not fully match the correct answer. For a candidate entity to be considered *correct*, there must be a full span-and-category match between the candidate and the gold standard named entity. Partial matches, as well as minor hallucinations, such as an incorrectly spelled entity type, are considered errors.

The LLM output of named entity candidates is thus analysed as follows: first, a count of all predicted entities is provided. Perfect and missed matches of (entity, entity category) are counted by comparing the model's predictions to the gold standard. Then, predicted entities that are not **perfect** matches are divided into four error classes: (1) LLM output where a valid NE instance[5] is assigned the wrong category from the set of valid NE categories[6] (dubbed **sources of confusion**), (2) a valid NE category is assigned to spans that have not been identified as named entities in the original dataset (**possible candidates**), (3) a valid named entity is assigned a named entity category that is not part of the original dataset (**new categories**) and (4) neither the named entity span nor the assigned entity category is valid (**pure noise**).

This error classification is a descriptive overview of the errors found in the models' output and aims to complement established evaluation metrics. Missed and perfect matches, as well as *sources of confusion* and *possible candidates*, are output categories that have been accounted for in existing evaluation metrics.[7] The classes *new categories* and *pure noise* are added to capture LLM-specific issues arising from LLMs' "hallucinations".

Counting error instances For *cluster prompts*, the counts of each error class represent the number of unique error instances found in each error class per cluster. For example, in cluster 1 of Climate-Change-NER (*climate-hazards*, *climate-problem-origins*, *climate-greenhouse-gases*), errors of the class **sources of confusion** are counted for this cluster only for each LLM. For *full prompts*, the reported counts per error class

[4] An average of the F1 scores calculated for each prompt.

[5] *Valid NE instance* is an instance that exists as a named entity span in the dataset.

[6] *Valid NE category* is a named entity category that is part of the dataset's entity types.

[7] These include missed entity spans, hypothesised entity spans where there are none, entity spans that are assigned the wrong category, entity spans with incorrect boundaries and correct NE category, and entity spans with incorrect boundaries and incorrect NE category.

represent the average from the six full-prompt versions. For example, the reported count of the error class **sources of confusion** will be the sum of the error counts for each of the six prompt versions[8] divided by six. The Python script for classification of error instances, and the tables with error counts for each error class and each model are available in the GitHub repository.

Points of comparison In a supervised NE recognition task, a model's output is only compared to the test split of the gold dataset, given that the train and development splits are used in the model's training. In the few-shot scenario described in Section 3, however, the model had not been exposed to the development set at all and had been exposed to a maximum of five sentences from the train set. For this reason, the LLMs' output is also compared to the combinations *test and train* and *test and development* data splits of the gold standard dataset. Differences in the number of missed matches between a model's predictions and the gold standard across the three points of comparison will show whether some of the candidates generated by the model are valid entities in the development and the train data splits.

In terms of F1-score, comparative performance has been seen between the larger models, gpt-4o and LLama-405B, and the smaller models, gpt-4o-mini and Llama-70B. For this reason, error classes are further analysed per two groups of models: **large** and **small**. The error class ranking for individual models is available in the GitHub repository.

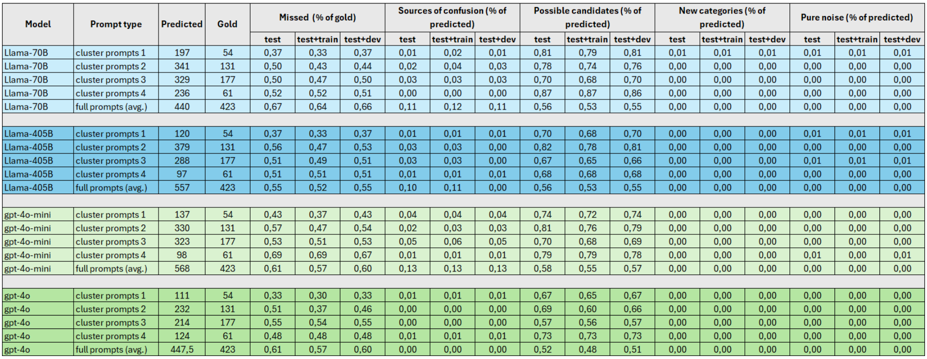
5 Results

Tables 1 and 2 summarize the error class counts per each prompt type and model, shown as percentages: the **missed** column shows the percentage of missed unique gold entities, while the other four columns show the percentage the respective error class has in the total number of unique predicted entity candidates. The columns **predicted** and **gold** capture the unique pairs of (named entity, named entity type) in a model's output and in the gold dataset, respectively. The recurrence of instances is not taken into account for the calculation of percentages in the two tables, as the focus is on the portion of unique instances in each error class; however, repeated occurrences are accounted for in the rankings of most-frequently represented categories and named entities in each error class; see the discussion under *Zeroing in on error classes* for more details.

All models generate a substantially higher number of entity candidates in a cluster-prompt scenario in Climate-Change-NER and across all prompt scenarios in BiodivNER. In terms of model families, Llama models generate, on average, more entity candidates, while OpenAI models tend to be more conservative.

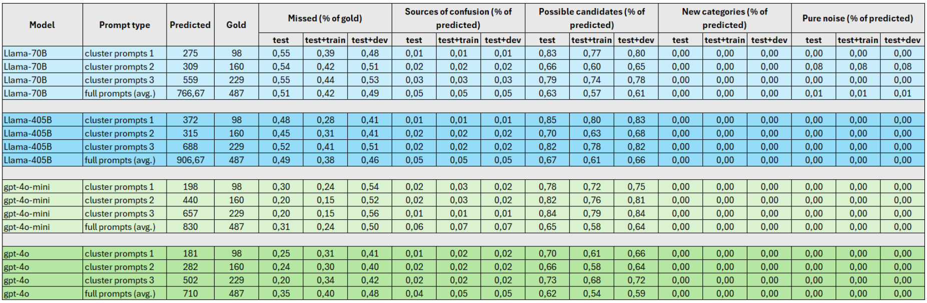
A higher number of entity candidates does not necessarily translate into better performance, as can be seen from the error count results for smaller models, which

[8] Prompts with random task examples with 3, 4 and 5 shots, and prompts with similar task examples with 3, 4 and 5 shots.



Model	Prompt type	Predicted	Gold	Missed (% of gold)			Sources of confusion (% of predicted)			Possible candidates (% of predicted)			New categories (% of predicted)			Pure noise (% of predicted)		
				test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev
Llama-70B	cluster prompts 1	197	54	0,37	0,33	0,37	0,01	0,02	0,01	0,81	0,79	0,81	0,01	0,01	0,01	0,01	0,01	0,01
Llama-70B	cluster prompts 2	341	131	0,50	0,43	0,44	0,02	0,04	0,03	0,78	0,74	0,76	0,00	0,00	0,00	0,00	0,00	0,00
Llama-70B	cluster prompts 3	329	177	0,50	0,47	0,50	0,03	0,03	0,03	0,70	0,68	0,70	0,00	0,00	0,00	0,00	0,00	0,00
Llama-70B	cluster prompts 4	236	61	0,52	0,52	0,51	0,00	0,00	0,00	0,87	0,87	0,86	0,00	0,00	0,00	0,00	0,00	0,00
Llama-70B	full prompts (avg.)	440	423	0,67	0,64	0,66	0,11	0,12	0,11	0,56	0,53	0,55	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	cluster prompts 1	120	54	0,37	0,33	0,37	0,01	0,01	0,01	0,70	0,68	0,70	0,00	0,00	0,00	0,01	0,01	0,01
Llama-405B	cluster prompts 2	379	131	0,56	0,47	0,53	0,03	0,03	0,00	0,82	0,78	0,81	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	cluster prompts 3	288	177	0,51	0,49	0,51	0,03	0,03	0,00	0,67	0,65	0,66	0,00	0,00	0,00	0,01	0,01	0,01
Llama-405B	cluster prompts 4	97	61	0,51	0,51	0,51	0,01	0,01	0,01	0,68	0,68	0,68	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	full prompts (avg.)	557	423	0,55	0,52	0,55	0,10	0,11	0,00	0,56	0,53	0,55	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 1	137	54	0,43	0,37	0,43	0,04	0,04	0,04	0,74	0,72	0,74	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 2	330	131	0,57	0,47	0,54	0,02	0,03	0,03	0,81	0,76	0,79	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 3	323	177	0,53	0,51	0,53	0,05	0,06	0,05	0,70	0,68	0,69	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 4	98	61	0,69	0,69	0,67	0,01	0,01	0,01	0,79	0,79	0,78	0,00	0,00	0,00	0,01	0,00	0,01
gpt-4o-mini	full prompts (avg.)	568	423	0,61	0,57	0,60	0,13	0,13	0,13	0,58	0,55	0,57	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 1	111	54	0,33	0,30	0,33	0,01	0,01	0,01	0,67	0,65	0,67	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 2	232	131	0,51	0,37	0,46	0,00	0,00	0,00	0,69	0,60	0,66	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 3	214	177	0,55	0,54	0,55	0,00	0,00	0,00	0,57	0,56	0,57	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 4	124	61	0,48	0,48	0,48	0,01	0,01	0,01	0,73	0,73	0,73	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	full prompts (avg.)	447,5	423	0,61	0,57	0,60	0,00	0,00	0,00	0,52	0,48	0,51	0,00	0,00	0,00	0,00	0,00	0,00

Table 1: Climate-Change-NER: Missed entities as % of gold entities and error class counts as % of predicted entities.



Model	Prompt type	Predicted	Gold	Missed (% of gold)			Sources of confusion (% of predicted)			Possible candidates (% of predicted)			New categories (% of predicted)			Pure noise (% of predicted)		
				test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev	test	test+train	test+dev
Llama-70B	cluster prompts 1	275	98	0,55	0,39	0,48	0,01	0,01	0,01	0,83	0,77	0,80	0,00	0,00	0,00	0,00	0,00	0,00
Llama-70B	cluster prompts 2	309	160	0,54	0,42	0,51	0,02	0,02	0,02	0,66	0,60	0,65	0,00	0,00	0,00	0,08	0,08	0,08
Llama-70B	cluster prompts 3	559	229	0,55	0,44	0,53	0,03	0,03	0,03	0,79	0,74	0,78	0,00	0,00	0,00	0,00	0,00	0,00
Llama-70B	full prompts (avg.)	766,67	487	0,51	0,42	0,49	0,05	0,05	0,05	0,63	0,57	0,61	0,00	0,00	0,00	0,01	0,01	0,01
Llama-405B	cluster prompts 1	372	98	0,48	0,28	0,41	0,01	0,01	0,01	0,85	0,80	0,83	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	cluster prompts 2	315	160	0,45	0,31	0,41	0,02	0,02	0,02	0,70	0,63	0,68	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	cluster prompts 3	688	229	0,52	0,41	0,51	0,02	0,02	0,02	0,82	0,78	0,82	0,00	0,00	0,00	0,00	0,00	0,00
Llama-405B	full prompts (avg.)	906,67	487	0,49	0,38	0,46	0,05	0,05	0,05	0,67	0,61	0,66	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 1	198	98	0,30	0,24	0,54	0,02	0,03	0,02	0,78	0,72	0,75	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 2	440	160	0,20	0,15	0,52	0,02	0,03	0,02	0,82	0,76	0,81	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	cluster prompts 3	657	229	0,20	0,15	0,56	0,01	0,01	0,01	0,84	0,79	0,84	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o-mini	full prompts (avg.)	830	487	0,31	0,24	0,50	0,06	0,07	0,07	0,65	0,58	0,64	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 1	181	98	0,25	0,31	0,41	0,01	0,02	0,02	0,70	0,61	0,66	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 2	282	160	0,24	0,30	0,40	0,02	0,02	0,02	0,66	0,58	0,64	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	cluster prompts 3	502	229	0,20	0,34	0,42	0,02	0,02	0,02	0,73	0,68	0,72	0,00	0,00	0,00	0,00	0,00	0,00
gpt-4o	full prompts (avg.)	710	487	0,35	0,40	0,48	0,04	0,05	0,05	0,62	0,54	0,59	0,00	0,00	0,00	0,00	0,00	0,00

Table 2: BiodivNER: Missed entities as % of gold entities and error class counts as % of predicted entities.

generate more noise. Across all models and almost all prompt types, the number of possible candidates drops once the spans from the train split of the gold dataset are added to the comparison set. This means that the models generated spans that are part of the train split - albeit not under the right category. This tendency is present, to a lesser extent, in the comparison with the development set. The miscategorisation of entity instances also explains why error counts of the category **sources of confusion** slightly increase once the train data split is added to the comparison. Percentage-wise, the error classes **new categories** and **pure noise** have generally very low values across the two datasets and all models. This indicates that the models can "follow" the guidance for identifying entities belonging to certain categories only.

Zeroing in on error classes The top three categories of **possible** entity candidates in **Climate-Change-NER**, identified by larger and smaller LLMs alike, albeit in different order, are: *climate-models*, *climate-nature*, and *climate-properties*. Among the most frequent candidates for *climate-models* are instances such as *GCM* or *General Circulation Models*, which in the gold datasets are only sometimes annotated as *climate-models*, usually when the term is more narrowly defined.[9] This echoes some of the findings by Epure and Hennequin (2022), who notice that in few-shot settings, pre-trained models tend to prioritize named entity cues more than context cues. The fact that the acronym *GCM* appears both as an entity and a non-entity adds a layer of complexity in the recognition stage that the LLMs cannot resolve based on context cues i.e. the term being narrowly-defined or not. All models identify spans such as *random forests* as valid instances, which indicates that there seems to be no differentiation between a climate-specific model and a general model that can be used in a climate scenario. LLMs sometimes delete extra whitespaces found in the gold dataset. Models would thus extract *WRF-UCM* instead of *WRF - UCM* as a climate model.

In the top-three categories of the **missed** error class, the category *climate-models* came in third for large and small models alike, following *climate-nature* and *climate-properties*. It included instances of LLMs failing to extract acronyms separately from the full name of a climate model, in situations where the acronym followed the name of a climate model.[10]

Small models tend to generate more invalid categories than their larger counterparts, especially in the error class **pure noise**. The invalid NE categories range from misspellings (*climate-greenhouse-gasses*, *climate-impats*), labels that are seemingly correct but contain a combination of Latin and Cyrillic letters, to categories that are not part of the original label set at all (*climate-projects*, *climate-regulations*, *climate-study-field...*).

For **BiodivNER**, the top three categories of **possible** candidates identified by large LLMs are: *quality*, *organism*, and *environment*; a slightly different frequency ranking was noticed in smaller LLMs, namely: *organism*, *quality*, and *phenomena*. While some of the candidates could be considered valid instances, such as *guinea pig* and *termites* for *organism*, other candidates include names of organisations and people, which is not in line with the NE class description.[11]

The top three entity types in the **missed** error class for large and small models are: *quality*, *matter*, and *organism*. Some of the most frequently missed instances include *species*, *tree*, and *plant*, which are found in the error class **possible candidates** as parts of longer spans.

When it comes to “hallucinations”, models that are on the smaller side tend to generate them more frequently and in greater variety. Large models did not have any errors in the *new categories* error class, and generated only 4 invalid categories in the

[9] For example, in the span *NASA / GIS GCM*, *GCM* is annotated as a climate model.

[10] In the gold dataset, acronyms are annotated as separate entities. For example, in the span *Coupled Model Intercomparison Project Phase 5 (CMIP5)*, *Coupled Model Intercomparison Project Phase 5* and *CMIP5* are two separate entities of the type *climate-models*.

[11] The class is defined as “All individual life forms such as microorganisms, plants, animals, mammals, insects, fungi, bacteria etc.”

pure noise error class. Smaller models, on the other hand, generated 6 new categories for existing spans and identified 56 invalid spans across more than 15 invalid categories, including combined labels such as *organism (quality)*.

6 Conclusion and discussion

This paper proposes a methodology for classifying errors detected in the output of LLMs following a few-shot NER task, where NER is defined as a question-answering task with a specific output requirement. The proposed error classification provides a snapshot of how LLMs fail and a systematic comparison of the output from multiple LLMs. The descriptive error counts could serve as a basis for (a) additional quantitative and (b) guided qualitative analyses. Under (a), one may explore what percentage of the errors classified as *possible candidates* are partial matches with spans from the gold dataset. Another useful information would be the average span lengths across entity instances in different error categories, and possible variations in the lengths of sentences where entities belonging to different error categories are found. This could help steer efforts under (b), which might include a hands-on comparison of sentences where repeated error instances are found.

In this study, the counts of errors in different prompt versions (random and similar task examples with 3, 4, and 5 shots) were averaged due to the limited variations in the F1 score achieved by different prompts and the primary focus being on the comparison of the four models' performance rather than prompt-specific variations. It would be beneficial to conduct error comparison per prompt output, which might show if and how each model's generation had been affected by the prompt design.

Finally, the few-shot NER task might benefit from a (self)-verification step (Li et al., 2024; Madaan et al., 2023), where either the same model or a different model "checks" the errors classified as *possible candidates* by the *annotator* model and flags up valid entity candidates. In addition, the prompt may include an instruction for the LLM to not change the input text, which might help with cases where the model removes whitespaces in the generated texts.

7 Acknowledgements

The research presented in this paper was conducted within the research project InsightsNet (`https://insightsnet.org/`), which is funded by the Federal Ministry of Education and Research (BMBF) under grant no. 01UG2130A. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The author would like to thank the anonymous reviewers for their insightful feedback and comments.

References

Abdelmageed, N., Löffler, F., Feddoul, L., Algergawy, A., Samuel, S., Gaikwad, J., … König-Ries, B. (2022). Biodivnere: Gold standard corpora for named entity recognition and relation extraction in the biodiversity domain. *Biodiversity Data Journal*, *10*.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F., … others (2023). Gpt-4 technical report. arxiv. *arXiv preprint arXiv:2303.08774*.

Ashok, D., & Lipton, Z. C. (2023). Promptner: Prompting for named entity recognition. *ArXiv*, *abs/2305.15444*. Retrieved from https://api.semanticscholar.org/CorpusID:258887456

Bhattacharjee, B., Trivedi, A., Muraoka, M., Ramasubramanian, M., Udagawa, T., Gurung, I., … others (2024). Indus: Effective and efficient language models for scientific applications. *arXiv preprint arXiv:2405.10725*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., … others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., … others (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, *25*(70), 1–53.

Epure, E. V., & Hennequin, R. (2022, June). Probing pre-trained auto-regressive language models for named entity typing and recognition. In N. Calzolari et al. (Eds.), *Proceedings of the thirteenth language resources and evaluation conference* (pp. 1408–1417). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2022.lrec-1.151

Inie, N., Druga, S., Zukerman, P., & Bender, E. M. (2024). From" ai" to probabilistic automation: How does anthropomorphization of technical systems descriptions influence trust? In *The 2024 acm conference on fairness, accountability, and transparency* (pp. 2322–2347).

Li, Z., Xu, X., Shen, T., Xu, C., Gu, J.-C., Lai, Y., … Ma, S. (2024). Leveraging large language models for nlg evaluation: Advances and challenges. *arXiv preprint arXiv:2401.07103*.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., … others (2023). Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, *36*, 46534–46594.

Moscato, V., Postiglione, M., & Sperlí, G. (2023). Few-shot named entity recognition: Definition, taxonomy and research directions. *ACM Transactions on Intelligent Systems and Technology*, *14*(5), 1–46.

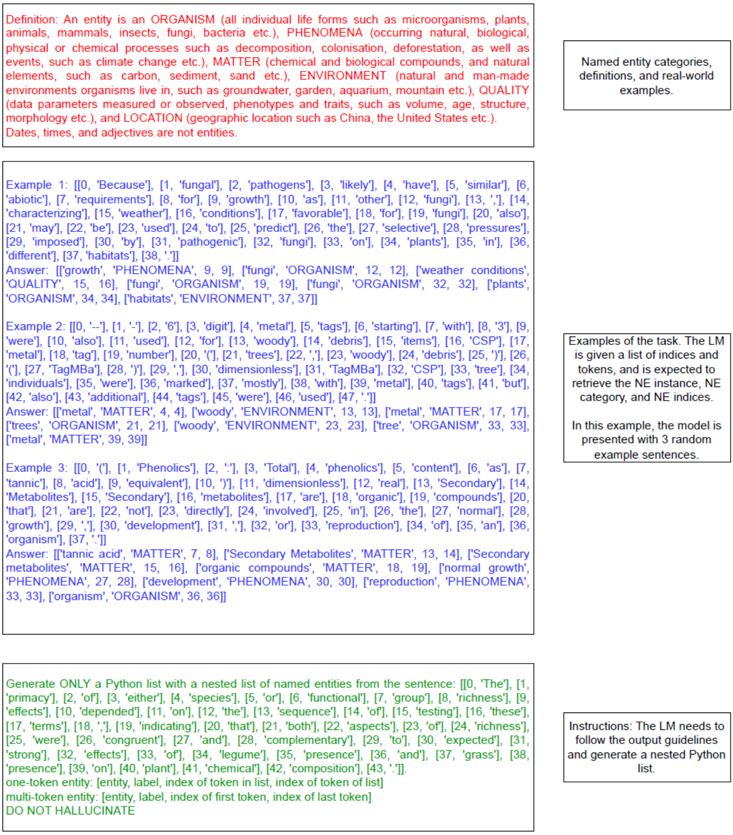
Reimers, N., & Gurevych, I. (2019, 11). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing.* Association for Computational Linguistics. Retrieved from https://arxiv.org/abs/1908.10084

Volkanovska, E. (2025, March). Large language models as annotators of named entities in climate change and biodiversity: A preliminary study. In V. Basile, C. Bosco, F. Grasso, M. O. Ibrohim, M. Skeppstedt, & M. Stede (Eds.), *Proceedings of the 1st workshop on ecology, environment, and natural language processing (nlp4ecology2025)* (pp. 24–33). Tallinn, Estonia: University of Tartu Library. Retrieved from https://aclanthology.org/2025.nlp4ecology-1.7/

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., . . . Wang, G. (2023). Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Appendix A: Prompt example

The prompt included here is motivated by the prompt used by Ashok and Lipton (2023). One major difference is that in this prompt, the LLM processes a task requirement comprised of natural language and Python code, and is instructed to generate output as a Python list. The prompt in this Appendix contains three random task examples from the dataset BiodivNER.



Definition: An entity is an ORGANISM (all individual life forms such as microorganisms, plants, animals, mammals, insects, fungi, bacteria etc.), PHENOMENA (occurring natural, biological, physical or chemical processes such as decomposition, colonisation, deforestation, as well as events, such as climate change etc.), MATTER (chemical and biological compounds, and natural elements, such as carbon, sediment, sand etc.), ENVIRONMENT (natural and man-made environments organisms live in, such as groundwater, garden, aquarium, mountain etc.), QUALITY (data parameters measured or observed, phenotypes and traits, such as volume, age, structure, morphology etc.), and LOCATION (geographic location such as China, the United States etc.).
Dates, times, and adjectives are not entities.

Named entity categories, definitions, and real-world examples.

Example 1: [[0, 'Because'], [1, 'fungal'], [2, 'pathogens'], [3, 'likely'], [4, 'have'], [5, 'similar'], [6, 'abiotic'], [7, 'requirements'], [8, 'for'], [9, 'growth'], [10, 'as'], [11, 'other'], [12, 'fungi'], [13, ','], [14, 'characterizing'], [15, 'weather'], [16, 'conditions'], [17, 'favorable'], [18, 'for'], [19, 'fungi'], [20, 'also'], [21, 'may'], [22, 'be'], [23, 'used'], [24, 'to'], [25, 'predict'], [26, 'the'], [27, 'selective'], [28, 'pressures'], [29, 'imposed'], [30, 'by'], [31, 'pathogenic'], [32, 'fungi'], [33, 'on'], [34, 'plants'], [35, 'in'], [36, 'different'], [37, 'habitats'], [38, '.']]
Answer: [['growth', 'PHENOMENA', 9, 9], ['fungi', 'ORGANISM', 12, 12], ['weather conditions', 'QUALITY', 15, 16], ['fungi', 'ORGANISM', 19, 19], ['fungi', 'ORGANISM', 32, 32], ['plants', 'ORGANISM', 34, 34], ['habitats', 'ENVIRONMENT', 37, 37]]

Example 2: [[0, '--'], [1, '-'], [2, '6'], [3, 'digit'], [4, 'metal'], [5, 'tags'], [6, 'starting'], [7, 'with'], [8, '3'], [9, 'were'], [10, 'also'], [11, 'used'], [12, 'for'], [13, 'woody'], [14, 'debris'], [15, 'items'], [16, 'CSP'], [17, 'metal'], [18, 'tag'], [19, 'number'], [20, '('], [21, 'trees'], [22, ','], [23, 'woody'], [24, 'debris'], [25, ')'], [26, '('], [27, 'TagMBa'], [28, ')'], [29, ','], [30, 'dimensionless'], [31, 'TagMBa'], [32, 'CSP'], [33, 'tree'], [34, 'individuals'], [35, 'were'], [36, 'marked'], [37, 'mostly'], [38, 'with'], [39, 'metal'], [40, 'tags'], [41, 'but'], [42, 'also'], [43, 'additional'], [44, 'tags'], [45, 'were'], [46, 'used'], [47, '.']]
Answer: [['metal', 'MATTER', 4, 4], ['woody', 'ENVIRONMENT', 13, 13], ['metal', 'MATTER', 17, 17], ['trees', 'ORGANISM', 21, 21], ['woody', 'ENVIRONMENT', 23, 23], ['tree', 'ORGANISM', 33, 33], ['metal', 'MATTER', 39, 39]]

Example 3: [[0, '('], [1, 'Phenolics'], [2, ':'], [3, 'Total'], [4, 'phenolics'], [5, 'content'], [6, 'as'], [7, 'tannic'], [8, 'acid'], [9, 'equivalent'], [10, ')'], [11, 'dimensionless'], [12, 'real'], [13, 'Secondary'], [14, 'Metabolites'], [15, 'Secondary'], [16, 'metabolites'], [17, 'are'], [18, 'organic'], [19, 'compounds'], [20, 'that'], [21, 'are'], [22, 'not'], [23, 'directly'], [24, 'involved'], [25, 'in'], [26, 'the'], [27, 'normal'], [28, 'growth'], [29, ','], [30, 'development'], [31, ','], [32, 'or'], [33, 'reproduction'], [34, 'of'], [35, 'an'], [36, 'organism'], [37, '.']]
Answer: [['tannic acid', 'MATTER', 7, 8], ['Secondary Metabolites', 'MATTER', 13, 14], ['Secondary metabolites', 'MATTER', 15, 16], ['organic compounds', 'MATTER', 18, 19], ['normal growth', 'PHENOMENA', 27, 28], ['development', 'PHENOMENA', 30, 30], ['reproduction', 'PHENOMENA', 33, 33], ['organism', 'ORGANISM', 36, 36]]

Examples of the task. The LM is given a list of indices and tokens, and is expected to retrieve the NE instance, NE category, and NE indices.

In this example, the model is presented with 3 random example sentences.

Generate ONLY a Python list with a nested list of named entities from the sentence: [[0, 'The'], [1, 'primacy'], [2, 'of'], [3, 'either'], [4, 'species'], [5, 'or'], [6, 'functional'], [7, 'group'], [8, 'richness'], [9, 'effects'], [10, 'depended'], [11, 'on'], [12, 'the'], [13, 'sequence'], [14, 'of'], [15, 'testing'], [16, 'these'], [17, 'terms'], [18, ','], [19, 'indicating'], [20, 'that'], [21, 'both'], [22, 'aspects'], [23, 'of'], [24, 'richness'], [25, 'were'], [26, 'congruent'], [27, 'and'], [28, 'complementary'], [29, 'to'], [30, 'expected'], [31, 'strong'], [32, 'effects'], [33, 'of'], [34, 'legume'], [35, 'presence'], [36, 'and'], [37, 'grass'], [38, 'presence'], [39, 'on'], [40, 'plant'], [41, 'chemical'], [42, 'composition'], [43, '.']].
one-token entity: [entity, label, index of token in list, index of token of list]
multi-token entity: [entity, label, index of first token, index of last token]
DO NOT HALLUCINATE

Instructions: The LM needs to follow the output guidelines and generate a nested Python list.

Figure 1: Prompt example: Three randomly selected task examples (question-answer pairs) from BiodivNER's training data.

Correspondence

Elena Volkanovska

Technische Universität Darmstadt
Institute of Linguistics and Literary Studies
Darmstadt, Germany
elena.volkanovska@tu-darmstadt.de