Large Language Models as Annotators of Named Entities in Climate Change and Biodiversity: A Preliminary Study

Elena Volkanovska
Technische Universität Darmstadt / Residenzschloss 1, 64283 Darmstadt, Germany
elena.volkanovska@tu-darmstadt.de

Abstract

This paper examines whether few-shot techniques for Named Entity Recognition (NER) utilising existing large language models (LLMs) as their backbone can be used to reliably annotate named entities (NEs) in scientific texts on climate change and biodiversity. A series of experiments aim to assess whether LLMs can be integrated into an end-to-end pipeline that could generate token- or sentence-level NE annotations; the former being an ideal-case scenario that allows for seamless integration of existing with new token-level features in a single annotation pipeline. Experiments are run on four LLMs, two NER datasets, two input and output data formats, and ten and nine prompt versions per dataset. The results show that few-shot methods are far from being a silver bullet for NER in highly specialised domains, although improvement in LLM performance is observed for some prompt designs and some NE classes. Few-shot methods would find better use in a human-in-the-loop scenario, where an LLM's output is verified by a domain expert.

1 Introduction

Analysing the language of climate change is an important step in following and understanding ongoing debates in this field. In a corpus linguistics setting, an important precondition for performing such an analysis is the access to corpora that have been annotated with morpho-syntactic and semantic features at the token level. Named entities (NEs) belong to the latter category and constitute an important part of linguistic analysis: Glaser et al. (2022) underline that linguistic choices in terms of decisions to explicitly name or leave out a certain entity or concept is an important notion in analysing political speeches. This line of thinking can easily apply to texts of various genres from the climate change domain, too.

In many instances, available corpora for corpus linguistics research, such as those hosted on English-Corpora.org,[1] rarely offer token-level annotations that extend beyond lemma and part-of-speech (POS) tags, and eventually syntactic dependency tags. These corpora can be obtained as pre-tokenized data; preserving existing token-level features and enriching them with custom NE annotations is contingent upon having (a) an annotation tool capable of processing tokenized input, and (b) having sufficient data to train a custom NER component within the tool. Depending on the annotation tool, such training data must usually be annotated in the IOB or BIOES/BILOU format[2] or contain character span information about the NE instance.

Challenge (a) is alleviated by the fact that (1) some known annotation tools, such as stanza (Qi et al., 2020) and trankit (Nguyen et al., 2021), can accept pre-tokenized input, and (2) obtaining high-quality morpho-syntactic features by re-annotating the corpus is generally unproblematic.[3] Challenge (b) is more complex, especially concerning the annotation of specialised corpora. In the context of adding climate-change-related token-level NE annotations that would be relevant to analysing the scientific climate change discourse, which could involve NE categories such as *greenhouse-gases* or *climate-datasets*, the first step would be to define a set of relevant categories, and the second to obtain a high-quality annotated corpus of sufficient size to train an NER compo-

[1] https://www.english-corpora.org/

[2] "IOB" stands for "inside, outside, beginning", while "BIOES/BILOU" stands for "beginning, inside, outside, end/last, single (element)/unit (element)"

[3] This would not be an ideal solution if the goal is to preserve the original token-level features.

Proceedings of the 1st Workshop on Ecology, Environment, and Natural Language Processing (NLP4Ecology2025), pages 24–33

nent of a tool for linguistic annotation. Creating richly annotated specialised corpora is thus a time- and resource-intensive activity.

Meanwhile, large language models (LLMs) have been seen as possible "destabilizers" of "inequalities of academic research", as they might allow moderately-funded labs to perform analyses that were previously accessible to well-funded institutes only (Törnberg, 2024, p.17). Motivated by the positive results in LLM-powered few-shot NER for specialised domains reported in Ashok and Lipton (2023), this paper employs a number of few-shot experiments to investigate whether this "destabilization effect" also transfers onto the annotation of NEs in scientific literature on climate change and biodiversity. Experiments undertaken in the scope of this study should answer two questions: (Q1) *Can LLMs be used as reliable "annotators" of named entities at the token and sentence levels in the domains of climate change and biodiversity?* and (Q2) *Does providing tokenized input affect an LLM's performance when identifying named entities in these domains?* Descriptive information about the datasets used in the study and extensive supplementary materials related to the experiments and the results are available in a dedicated GitHub repository.[4] Finally, an effort is made to refrain from using anthropomorphic language when disucssing LLMs (Inie et al., 2024), as long as this does not hinder the description of LLM-based systems, methodologies and functionalities.

2 Related work

Jehangir et al. (2023) distinguish between three types of NER techniques: a rule-based approach, unsupervised learning, and supervised learning. A rule-based approach entails the careful crafting of domain-specific rules to extract and classify patterns representing NEs of interest. Unsupervised learning is used in data-poor contexts, but can yield results that are difficult to evaluate. Supervised learning utilizes manually annotated data to learn representations of relevant NE categories. Corpus annotation libraries, such as CoreNLP (Manning et al., 2014), spaCy, stanza, and trankit, have incorporated supervised learning in a modular pipeline design, allowing researchers to train their own NER component provided that they have sufficient data.

The advent of Transformer-based LMs has put the limelight on transfer learning and fine-tuning, methodologies that demonstrate robust results with fewer manually labelled training examples. In fine-tuning, the architecture of an LM is modified in line with the task requirements: Wang et al. (2022) present a methodology for learning an LM to understand language structure, and then test its performance on downstream tasks including NER. Many of the tools developed in this way, such as BiodivBERT (Abdelmageed et al., 2023), are models that have been developed for an NER task only and merging their output with the morpho-syntactic token-level features obtained from a linguistic annotation library is not always a seamless process due to variations in tokenization approaches.[5]

The increased availability of open-source and paid text-generation and question-answering models, alongside reports of pre-trained LLMs performing well on NLP tasks in zero- and few-shot settings in data-poor contexts (Brown et al., 2020), have fuelled the interest in experimenting with zero-shot and few-shot NER approaches. In most instances, this means that NER is defined as a question-answering task, where the LLM is expected to generate an answer based on a prompt sent to the system. Epure and Hennequin (2022) perform zero-shot and few-shot NER using GPT-2. Before prompting the model, they ensure a low ambiguity level between NE categories by merging possibly confusing NE labels into a single, unambiguous label. They also simplify the task by prompting the model to recognise one NE category at a time. Wang et al. (2023) ensure that the input sequence from which the model is expected to extract NEs is semantically similar to the example sequence in the prompt template by retrieving the k nearest neighbour of the input sequence. They also prompt the model to enclose the NE into special tokens, which should allow for span retrieval. Ashok and Lipton (2023) have presented an intuitive approach to NER, where they propose a prompt template that can easily be customized to any project using own NE categories and definitions. Their approach has been implemented in

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

[5] A "token" can be a unit at the word- or punctuation-, character-, or sub-word level. Discussing tokenization approaches is beyond the scope of this study; however, it is worth mentioning that LMs using transformer architecture (Vaswani et al., 2017) mostly rely on sub-word units.

spacy-llm's NE annotation pipeline, where users can define NE categories on the fly and annotate their data with an LM of their choice.[6]

This study builds on existing work in the field of few-shot NER and conducts experiments using different prompt templates and a varying number of task examples. It differs from previous methods in (1) the format of the input given to the model and the requested output, and (2) the use of highly-specialised NER datasets, which, to the best of my knowledge, have not been used in a few-shot NER setting previously.

3 Data

Basic descriptive information about the two NER datasets that are used in the experiments described in Section 4 is provided below; a comprehensive dataset description involving definitions of each NE class, information about the distribution of NE instances per category and per data split, descriptive statistical sentence- and token-level information, as well as the ten most and least frequent NE instances per each NE class, are provided in the dataset documentation available in the dedicated GitHub repository referred to in Section 1.

Climate-Change-NER is a publicly-available dataset[7] for English-language NER in scientific texts on climate change, developed in an IBM Research AI[8]-led initiative, involving NASA[9] (Bhattacharjee et al., 2024) among other organisations. The dataset has 13 climate-specific NE classes, which originate from complex taxonomies used in climate-related literature. These are: *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*. Seed keywords, such as *wildfire* and *floods*, had been used to collect a total of 534 abstracts from the Semantic Scholar Academic Graph (Kinney et al., 2023), which were then manually annotated with the IOB tagging scheme, with the help of a set of class-specific dictionaries (Pfitzmann, 2024). The train and test data splits, which are used in the experiments of this paper, contain 985 and 177 sentences and 3029 and 555 NE instances respectively.

BiodivNER is a publicly-available dataset[10] for English-language NER in the biodiversity domain (Abdelmageed et al., 2022). The dataset has 6 biodiversity-related NE classes: *organism*, *phenomena*, *matter*, *environment*, *quality*, and *location*. The annotated corpus comprises of abstracts, tables, and metadata files collected by using a set of keywords from Semedico,[11] BEF-China,[12] and data.world[13] and manually annotated with the IOB tagging scheme. BiodivNER's train and test data splits contain 1828 and 229 sentences and 6709 and 1277 NE instances respectively.

4 Methodology

This section presents the steps taken to preprocess the data, the prompt design, the LLMs used in the experiments, the evaluation approach, and the baseline against which the LLMs' performance is compared.

4.1 Data preprocessing

The NER data is used in two settings: (1) to train a custom NER component in spaCy, and (2) to design prompts for few-shot learning. Use case (1) requires span information about each NE instance, while for use case (2) each sentence needs to be saved as a Python list, with each token index and token saved as sublists and as a string. To achieve (1) and guarantee compatibility between each dataset's and spaCy's tokenization, all sentences were re-tokenized and only those that were identical to the tokenized sentences in the original datasets were taken into account. All re-tokenized sentences for Climate-Change-NER were identical; from BiodivNER, 90 re-tokenized sentences from the train file, and 11 from the development and test file each were not identical.

4.2 Prompt design

To explore whether the task input-output format influences a model's performance, the study adopts a custom prompt design that differs from the few-shot prompt design suggested by Ashok and Lipton (2023) in the following features: (1) the definition of each NE class is followed by

[6] *spacy-llm* is spaCy's LLM-supporting package, available at `https://github.com/explosion/spacy-llm`.

[7] `https://huggingface.co/datasets/ibm/Climate-Change-NER`

[8] International Business Machines Corporation

[9] National Aeronautics and Space Administration

[10] `https://zenodo.org/records/6575865`

[11] A semantic search engine for the life sciences.

[12] `https://bef-china.com/`

[13] `https://data.world/`

several real-world instances of the respective NE class; (2) the task examples (TEs) include sentences presented either as a Python string or a Python list of tokens and token indices, hereinafter referred to as *string-based* and *token-based* input-output, and an answer section containing the expected output from the model; (3) the format of the task input sentence corresponds to the format of the task examples described in (2) i.e. is either a Python list or a string; (4) the LLM is not prompted to emulate "reasoning" for its decision; (5) only true NE instances are provided as examples of correct answers. The features (4) and (5) were implemented after the preliminary tests showed that they did not contribute to consistent improvement in the results. Each prompt has three sections: (a) a *definitions-and-instances* section, where real-world instances of the NE class accompany its definition, (b) a *task example* section, which includes an n number of examples of the task the model is expected to complete, and (c) a *task* section, where the model is "asked" to annotate a sentence and return its output in a specific format. Figure 1 provides an overview of the prompt design.

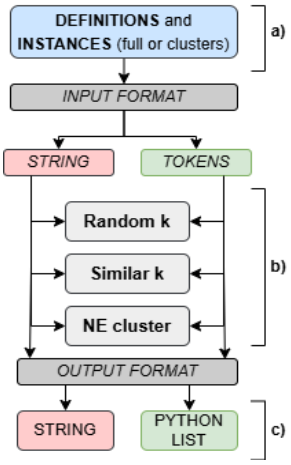

Figure 1: Blueprint for prompt design. The *string-based* input-output format refers to the task of identifying NEs at the sentence level, while a *token-based* format involves identifying NEs at the token level.

Section (a) remains unchanged in each prompt of the prompt versions described below. For BiodivNER, the definitions of the NE categories included in section (a) have been obtained from the description of the dataset creation and annotation process, available in Abdelmageed et al. (2022). The definitions of the NE classes contained in Climate-Change-NER are available in the dataset card on Hugging Face, referred to in the dataset description in Section 3. Sections (b) and (c) are created by applying two formats for the input-output requirements as described in prompt features (2) and (3), and by introducing three different selection criteria for examples included in the task-example (TE) pairs of section (b).[14]

Prompt version one: random k-examples A k number of random TEs is extracted from the train data split, where k can be 3, 4, or 5 TE pairs, and section (b) is populated with the selected TE pairs. This prompt version, where a k number of randomly chosen sentences is used in the TE section, follows the prompt design adopted in the work of Ashok and Lipton (2023).

Prompt version two: semantically similar k-examples Motivated by the prompt design presented in Wang et al. (2023), each sentence of the test split of both datasets is paired with five sentences of the train data split, which have the highest similarity score with the test sentence. Semantic text similarity is calculated with the library sentence-transformers[15] (Reimers and Gurevych, 2019) and the model *sentence-transformers/stsb-distilroberta-base-v2*. The idea is to investigate whether LLMs' performance can be improved by including in the TE pairs sentences that have a degree of similarity to the sentence the model is expected to process. Section (b) of the prompt is populated with k number of semantically similar TE pairs, where k can be 3, 4, or 5.

Prompt version three: clustered NE classes To simplify the task at hand, clusters of NE classes within each dataset are created on the basis of the classes' perceived relatedness. The idea behind this prompt design choice is to (1) frame the models' output into a narrower, topic-related semantic field and (2) rather than collapse NE categories that bear a perceived degree of similarity, test if LLMs can differentiate between them. Four NE class clusters are created for Climate-Change-NER and three for BiodivNER. Prompt sections (a) and (b) are pop-

[14] A limitation of a maximum number of 60 tokens was introduced for TE pairs from BiodivNER's training data, due to the observation that the data contained tokenized sentences whose length varied from 3 to 1053 tokens. Such a limitation was not necessary for Climate-Change-NER training samples, as the length of sentences varied between 32 and 115 tokens.

[15] https://sbert.net/

ulated with definitions and four randomly selected TE pairs pertaining only to the cluster's classes. The NE clusters for Climate-Change-NER 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*.

Input-output format For **string-based** input, the TEs include a string and the correct NE instances and their categories in parenthesis. The model is expected to generate the correct NE instance and its category in parentheses, but not the token indices pertaining to the tokens within the span. For **token-based** input, the TEs include tokenized sentences containing a token and a token index. The model is expected to identify the NE instance, its category, and the start- and end-token indices. Ideally, the token-based output should allow for simple integration of a model's annotation with existing token-level features.

Prompt version	***k*=3**	***k*=4**	***k*=5**
Random *k*	177	177	177
Similar *k*	177	177	177
NE cluster 1	0	177	0
NE cluster 2	0	177	0
NE cluster 3	0	177	0
NE cluster 4	0	177	0
Prompts, per input type	**354**	**1062**	**354**
Prompts, both input types	**708**	**2124**	**708**

Table 1: Number of prompts for test sentences of Climate-Change-NER for each prompt version and input type (token/string based).

4.3 Language models

The choice of LLMs was guided by two factors: previous successful deployment in similar tasks and cost. Two models of OpenAI's GPT family, gpt-4o-2024-05-13 (hereinafter: gpt-4o) and gpt-4o-mini,[16] were run using OpenAI's API. OpenAI's models were chosen over other proprietary models of similar performance and price range due to their previous successful deployment in a similar setting (Ashok and Lipton, 2023). The experiments are also run on two open-source models: Meta-Llama-3.1-70B-Instruct (hereinafter: Llama-70B) and Meta-Llama-3.1-405B-Instruct (hereinafter: Llama-405B), both developed by Meta and run through an API of Nebius AI Studio.[17] The total cost of the experiments is reported in Section 7.

Prompt version	***k*=3**	***k*=4**	***k*=5**
Random *k*	229	229	229
Similar *k*	229	229	229
NE cluster 1	0	229	0
NE cluster 2	0	229	0
NE cluster 3	0	229	0
Prompts, per input type	**458**	**1145**	**458**
Prompts, both input types	**916**	**2290**	**916**

Table 2: Number of prompts for test sentences of Climate-Change-NER for each prompt version and input type (token/string based).

4.4 Evaluation

Baseline The performance of the four models on the BiodivNER dataset is compared against the results of BiodivBERT (Abdelmageed et al., 2023), an LM pre-trained and fine-tuned specifically for an NER task in the biodiversity domain, with a reported F1 score of **0.87**. For Climate-Change-NER, the baseline is that of the model INDUSBASE (Bhattacharjee et al., 2024), an LM pre-trained and fine-tuned on relevant scientific data, with a reported F1 score of **0.64**.

Custom NER components within tools for linguistic annotation To measure how the number of NE instances per category affects the performance of a custom NER component within an annotation tool, custom NER components were trained on each dataset using spaCy and the model en_core_web_lg[18] as a base model. SpaCy's NER tagger achieves an F1 score of 0.73 on BiodivNER's test data, and 0.43 on the Climate-Change-NER's test data.

Token-based prompts Micro F1 score is calculated and reported in accordance with the standard CoNLL metric (Sang and De Meulder, 2003), as well as simple span-and-category matches (Chinchor and Sundheim, 1993). The former refer to

[16] https://platform.openai.com/docs/models

[17] https://studio.nebius.ai/.

[18] https://spacy.io/models/en_core_web_lg

a complete match in NE instance, label, and NE span boundaries (start- and end-token), while the latter takes into account only the NE instance and label, but not token indices. Reporting simple span-and-category matches serves as a point of comparison with the results of the string-based prompts.

String-based prompts For these prompts, the goal is to identify NEs at the sentence level, the F1 score is based on span-and-category matches, with strict span boundaries. Partial span matches are not considered true positives.

5 Results and analysis

Tables 3 and 4 summarize the F1 scores for experiments conducted on the test data split of Climate-Change-NER and BiodivNER involving the prompts described in Section 4.2. One iteration was performed on each prompt set and on each model. In the tables, *k* stands for the number of TE pairs included in the prompt. Prior to calculating the results, each model's output was cleaned from misspelled or non-existing categories (e.g. *organsim* instead of *organism*).

Tables 5 and 6 present the percentage of span-and-category matches between a model's predicted NEs and the gold standard. *Span-and-category matches* measure instances where the model correctly identifies the span of an NE instance and the NE class. For token-based input and output, this means that the token indices are not taken into account when calculating the percentage of span-and-category matches, while for string-based input and output, the model is not expected to generate token indices at all. Therefore, these two tables allow one to gauge the degree to which a model is affected by the input-output format.

5.1 Quantitative analysis

Even the best-performing all-class token-based prompt & model combinations substantially lag behind the baseline NER models for the datasets, more so in the case of BiodivNER, where the baseline F1 score is 0.87 and spaCy's NER classifier F1 score is 0.73. For Climate-Change-NER, which has a baseline score of 0.64, the best-performing all-class prompt & model combination achieve an F1 score of 0.44, which is similar to spaCy's score of 0.43.

Model performance The average F1 scores for all prompts achieved by the tested LLMs is within the 0.24 to 0.43 range for both datasets. Per prompt type, the highest F1 score of 0.53 is achieved by gpt-4o on the token-based NE class cluster 1 of Climate-Change-NER and the lowest F1 score of 0.16 by Llama-70B on the string-based NE class cluster 4 of the same corpus. For **token-based** prompts, gpt-4o has the highest average score, followed by Llama-405B; gpt-4o-mini and Llama-70B come third and achieve equal performance. For **string-based** prompts, Llama-405B performs slightly better on Climate-Change-NER, followed by gpt-4o and the two smaller models; for BiodivNER, it is a tie between gpt-4o and Llama-405B.

In terms of overall model ranking, gpt-4o seems to be the best performer, closely followed by Llama-405B. Llama-70B comes third due to its slightly better performance on the BiodivNER dataset relative to gpt-4o-mini, the latter coming in fourth.

Prompt performance As expected, prompt design can affect the quality of the output. In general, including more TE pairs in the prompt yields better results for both random and similar TEs, with a few exceptions that were mostly noticed in the output of Meta's models for the random-k prompt version in BiodivNER; the number of TEs also seems to be more important than TEs' similarity to the task sentence. Task simplification by grouping NE classes showed benefits only in NE class cluster 1 of Climate-Change-NER; in all other instances, this step did not lead to better performance.

The impact of the input-output format is measured by calculating the simple **span-and-category matches** of the output with the gold standard in the test data split. For token-based prompts, this is the percentage of correctly predicted NEs when the token indices are not considered. Tables 5 and 6 show that the models handle token-based input well - in fact, token-based prompts achieve better average results on both datasets. Llama-405B ranks first in this performance measure on the Climate-Change-NER dataset, while gpt-4o outperforms the other three models on the BiodivNER dataset.

Per-class performance Given that token-based prompts outperformed string-based prompts, an analysis of per-class performance of **token-based** prompts was done on the two datasets. Per **dataset**, the best-performing and worst-

Prompt version	k	Total instances	gpt-4o-mini		gpt-4o-2024-05-13		Meta-Llama-3,1-70B-Instruct		Meta-Llama-3,1-405B-Instruct		Average, all models	Average, all models
			Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings
NE class cluster 1	4	85	**0,39**	**0,39**	**0,53**	**0,5**	0,33	**0,25**	**0,47**	**0,41**	**0,43**	**0,39**
NE class cluster 2	4	176	0,23	0,19	0,33	0,34	0,28	0,24	0,22	0,26	0,27	0,26
NE class cluster 3	4	226	0,33	0,29	0,43	0,23	**0,38**	0,23	0,42	0,31	0,39	0,27
NE class cluster 4	4	68	0,17	0,28	0,38	0,35	0,21	0,16	0,4	0,25	0,29	0,26
Random k examples	3	555	0,32	0,28	0,38	0,29	0,35	0,36	**0,42**	0,39	0,37	0,33
Random k examples	4	555	0,33	0,29	0,41	0,3	**0,37**	0,37	0,4	0,41	0,38	0,34
Random k examples	5	555	**0,36**	**0,32**	**0,44**	**0,32**	0,36	**0,39**	0,39	**0,42**	**0,39**	**0,36**
Similar k examples	3	555	0,33	0,33	0,38	0,38	**0,36**	0,4	0,38	0,43	0,36	0,39
Similar k examples	4	555	**0,36**	0,32	0,4	0,41	0,28	0,4	0,42	0,43	**0,37**	0,39
Similar k examples	5	555	**0,36**	**0,38**	**0,42**	**0,43**	0,3	0,4	**0,39**	**0,44**	**0,37**	**0,41**
Average F1 score (all prompts)			0,32	0,32	**0,41**	0,36	0,32	0,32	0,39	**0,38**	0,36	0,35

Table 3: Climate-Change-NER results: F1 scores for all versions of token- and string-based input-output prompts.

Prompt version	k	Total instances	gpt-4o-mini		gpt-4o-2024-05-13		Meta-Llama-3,1-70B-Instruct		Meta-Llama-3,1-405B-Instruct		Average, all models	Average, all models
			Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings
NE class cluster 1	4	186	0,21	**0,28**	0,34	**0,28**	0,2	0,16	0,22	**0,29**	0,24	**0,25**
NE class cluster 2	4	573	**0,26**	0,24	**0,42**	0,2	**0,27**	**0,25**	**0,33**	0,28	**0,32**	0,24
NE class cluster 3	4	518	0,21	0,25	0,35	0,22	0,24	0,23	0,23	0,28	0,26	**0,25**
Random k examples	3	1277	0,23	0,26	0,32	0,29	**0,28**	**0,27**	**0,31**	**0,33**	**0,29**	0,29
Random k examples	4	1277	**0,25**	0,27	**0,33**	**0,39**	0,25	0,26	0,3	**0,33**	**0,29**	**0,31**
Random k examples	5	1277	**0,25**	**0,29**	0,31	**0,39**	0,27	**0,27**	0,26	0,3	0,27	**0,31**
Similar k examples	3	1277	0,34	0,36	0,4	**0,46**	0,34	0,33	0,35	0,37	0,36	**0,38**
Similar k examples	4	1277	0,34	0,36	**0,46**	0,38	0,35	**0,36**	0,35	**0,39**	**0,38**	0,37
Similar k examples	5	1277	**0,35**	**0,38**	0,37	0,38	**0,38**	**0,36**	**0,38**	0,4	0,37	**0,38**
Average F1 score (all prompts)			0,27	0,3	**0,37**	**0,33**	0,29	0,28	0,3	**0,33**	0,31	0,31

Table 4: BiodivNER results: F1 scores for all versions of token- and string-based input-output prompts.

performing classes for Climate-Change-NER are *climate-organizations* (0.59)[19] and *climate-assets* (0.23) respectively. For BiodivNER, the best and worst performing classes are *organism* (0.48) and *matter* (0.18). Per **model**, for Climate-Change-NER, gpt-4o-mini and Llama-70B perform best on *climate-organizations* (0.63 and 0.51), while gpt-4o and Llama-405B on *climate-greenhouse-gases* (0.62 and 0.74). For BiodivNER, all models perform best on the class *organism* (score range of 0.43 to 0.52) and worst on *mater* (0.15 to 0.19).

5.2 Qualitative analysis

The two worst-performing classes in the output of the highest-F1 score models for all-class token-based prompts were further investigated. For Climate-Change-NER, this is the model gpt-4o with a prompt containing 5 random TEs, while for BiodivNER this is the same model with a prompt containing 4 similar TEs.

Climate-Change-NER The two worst-performing classes are *climate-assets* and *climate-problem-origins*. When annotating instances of *climate-assets*, defined as "objects or services of value to humans that can get destroyed or diminished by climate-hazards", the model tends to prefer the longest-span option: it annotates the span *pavement structure*, instead of *pavement*, *bioclimatic skyscrapers* instead of *skyscrapers*, *livestock industry* instead of just *livestock*. The model does not delineate well between *climate-assets*, *climate-nature*, and *climate-mitigations*. The model annotates as *climate-problem-origins*, defined as "problems that describe why the climate is changing", instances such as *global warming*, considered non-entity in the test split of the gold dataset. It also fails to annotate *emissions* as an entity of this class only when it is used in the context of climate change. Sources of energy, including *hydropower*, are also annotated with this class.[20]

BiodivNER The two lowest-scoring classes in this instance are *matter* (F1 of 0.18) and *location* (0.25). Instances incorrectly annotated with the class *matter*, defined as "chemical and biological compounds, and natural elements", usually involve cases when the model only annotates

[19] Average F1 score from all prompts and all models.

[20] In the gold dataset, *hydropower* is annotated with the class *climate-mitigations*.

Prompt version	k	Total instances	gpt-4o-mini		gpt-4o-2024-05-13		Meta-Llama-3,1-70B-Instruct		Meta-Llama-3,1-405B-Instruct		Average, all models	Average, all models
			Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings
NE class cluster 1	4	85	**0,61**	**0,38**	**0,75**	**0,54**	**0,68**	**0,51**	**0,74**	0,47	**0,7**	**0,48**
NE class cluster 2	4	176	0,38	0,34	0,47	0,46	0,48	0,38	0,4	**0,53**	0,43	0,43
NE class cluster 3	4	226	0,46	0,31	0,42	0,2	0,49	0,31	0,49	0,37	0,47	0,3
NE class cluster 4	4	68	0,34	0,22	0,56	0,35	0,5	0,35	0,5	0,34	0,48	0,3
Random k examples	3	555	0,35	0,29	0,36	0,3	0,34	0,35	**0,45**	0,39	0,38	0,33
Random k examples	4	555	0,39	0,31	0,4	0,31	**0,38**	**0,37**	0,44	**0,43**	**0,4**	0,36
Random k examples	5	555	**0,41**	**0,35**	**0,45**	**0,33**	**0,38**	0,4	0,42	**0,43**	**0,4**	**0,38**
Similar k examples	3	555	0,36	**0,35**	0,35	0,37	**0,34**	0,39	0,42	**0,45**	**0,38**	0,39
Similar k examples	4	555	**0,39**	0,34	0,38	0,41	0,26	**0,4**	**0,45**	0,44	0,37	0,4
Similar k examples	5	555	**0,39**	0,41	**0,39**	**0,45**	0,29	**0,4**	0,44	**0,45**	**0,38**	**0,43**
Average simple span score			0,41	0,33	0,45	0,37	0,41	0,39	**0,48**	**0,43**	0,44	0,38

Table 5: Climate-Change-NER: Span-and-category matches for token- and string-based input-output prompts. The values are given as percentages of total instances.

Prompt version	k	Total instances	gpt-4o-mini		gpt-4o-2024-05-13		Meta-Llama-3,1-70B-Instruct		Meta-Llama-3,1-405B-Instruct		Average, all models	Average, all models
			Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings	Tokens	Strings
NE class cluster 1	4	186	0,21	0,23	0,41	0,24	0,33	0,21	**0,47**	0,28	0,36	0,24
NE class cluster 2	4	573	**0,26**	0,22	**0,57**	0,31	0,29	0,24	0,43	0,26	**0,39**	0,26
NE class cluster 3	4	518	0,21	**0,36**	0,46	**0,44**	**0,36**	**0,36**	0,38	**0,37**	0,35	**0,38**
Random k examples	3	1277	0,23	**0,28**	0,34	0,38	0,31	0,27	**0,31**	**0,34**	0,3	0,32
Random k examples	4	1277	**0,25**	0,27	**0,36**	**0,41**	**0,34**	0,26	0,3	**0,34**	**0,31**	0,32
Random k examples	5	1277	**0,25**	**0,28**	0,32	0,4	0,27	**0,29**	0,26	0,29	0,28	0,32
Similar k examples	3	1277	**0,34**	0,36	0,39	0,46	0,37	0,32	0,28	0,35	0,35	0,38
Similar k examples	4	1277	**0,34**	0,37	**0,49**	0,46	**0,42**	**0,34**	0,26	0,37	**0,38**	**0,39**
Similar k examples	5	1277	0,35	**0,38**	0,34	0,46	0,39	**0,34**	**0,37**	**0,38**	0,36	**0,39**
Average simple span score			0,27	0,31	**0,41**	**0,40**	0,34	0,29	0,34	0,33	0,34	0,33

Table 6: BiodivNER: Span-and-category matches for token- and string-based input-output prompts. The values are given as percentages of total instances.

a nested span, which can function as an NE instance on its own and within a longer span (capturing only *woody debris* instead of *woody debris item*). The wrongly-annotated instances of *location*, defined as a "geographic location, such as China", are interesting, as they reveal plausible NE candidates that have not been included in the gold dataset, such as *Turkey*, *Papua New Guinea* and *tropical South America*.

6 Discussion and future work

The experiments reveal that few-shot NER methods are not a turnkey solution for highly-specialised NE annotation at token- and sentence-level, which answers Q1 and further highlights the importance of reflecting on and reporting LLMs' limitations on domain-specific tasks, especially at a time of benchmark-centric research. Nevertheless, the results also reveal possible use cases for LLMs in the context of NER, which include testing the robustness of datasets and further simplifying the task by focusing on isolated NE classes and extensive task descriptions; both of these are discussed in subsection 6.1.

Regarding the **input-output format** investigated within Q2, the experiments show that LLMs achieved slightly better performance on token-based than on string-based input. A plausible explanation for this might be that repeated NE instances in a single sentence are more likely to be identified with a token-based approach, as the LLM processes each token individually. In future iterations, it would be useful to investigate whether the prompt for string-based processing could benefit by including an instruction for the LLM to extract repeated occurrences of the same NE instance. Since LLMs' performance could improve with more context, it would be worthwhile investigating whether redefining the string-based prompt as a document-level NER task would yield better performance. Finally, it was noticed that the BiodivNER dataset contained many tokens that were remnants of PDF parsing, which might also have affected the LLMs' output for string-based prompts.

In many cases, there was an overlap in the classes on which the LLMs performed well or poorly. The experiment results seem to hint that

the **complexity** of the task could be rooted in the LLMs not having been exposed to sufficient data about the specialised domains. It would be interesting to test this approach on a domain-specific LLM developed for climate change question-answering, such as models belonging to the ClimateGPT family (Thulke et al., 2024). Unfortunately, this was not realistic for this study due to infrastructure constraints.

6.1 Possible use-cases

Testing robustness of datasets While LLMs cannot be considered reliable "annotators" in an end-to-end pipeline for corpus annotation, they could be valuable assets in testing the definitions and labels of an existing NER dataset. This is corroborated by the fact that in BiodivNER, the models identified valid NE candidates of the category *location*. This experimental setup would be an affordable way of probing NE definitions and categories prior to embarking on manual annotation. Such "probing" could also uncover class ambiguities, where an instance could make a plausible NE candidate of two or more classes.

Focusing on isolated NE classes While LLMs were not capable of capturing NEs in the same way a dedicated NE classifier would do, their performance on certain categories, such as *climate-greenhouse-gases* and *climate-organisations*, was acceptable. It would be interesting to explore how the models would perform in a single-class scenario with a more extensive task description.

7 Ethical considerations

This study uses publicly available datasets. The experiments do not require specialised infrastructure and can be reproduced using an API and the prompts provided in the dedicated GitHub repository. The costs for all experiments, per language model family are: ca. 40 EUR for OpenAI's GPT4 models, and ca. 20 EUR for Llama's 3.1 models.

Limitations

The experiments use text generation in an LLM-as-a-service setup, which makes them vulnerable to non-responsive APIs. Given that an LLM may not yield the same result twice even when prompted with the same text, it is impossible to guarantee 100% reproducibility. Guardrails against bias and offensive content are recommended before real-world deployment. Information considered confidential or sensitive should not be sent in API calls.

Funding

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.

Acknowledgments

The author would like to thank the anonymous reviewers for their helpful feedback.

References

Nora Abdelmageed, Felicitas Löffler, Leila Feddoul, Alsayed Algergawy, Sheeba Samuel, Jitendra Gaikwad, Anahita Kazem, and Birgitta König-Ries. 2022. Biodivnere: Gold standard corpora for named entity recognition and relation extraction in the biodiversity domain. *Biodiversity Data Journal*, 10.

Nora Abdelmageed, Felicitas Löffler, and Birgitta König-Ries. 2023. Biodivbert: a pre-trained language model for the biodiversity domain. *CEUR-WS. org*, pages 62–71.

Dhananjay Ashok and Zachary Chase Lipton. 2023. Promptner: Prompting for named entity recognition. *ArXiv*, abs/2305.15444.

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukumaran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Rong Zhang, Bharath Dandala, Rahul Ramachandran, Manil Maskey, et al. 2024. Indus: Effective and efficient language models for scientific applications. *arXiv preprint arXiv:2405.10725*.

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.

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Elena V. Epure and Romain Hennequin. 2022. Probing pre-trained auto-regressive language models for named entity typing and recognition. In *Proceedings of the Thirteenth Language Resources and*

Evaluation Conference, pages 1408–1417, Marseille, France. European Language Resources Association.

Luis Glaser, Ronny Patz, and Manfred Stede. 2022. Unsc-ne: A named entity extension to the un security council debates corpus. *Journal for Language Technology and Computational Linguistics*, 35(2):51–67.

Nanna Inie, Stefania Druga, Peter Zukerman, and Emily M Bender. 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*, pages 2322–2347.

Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. 2023. A survey on named entity recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. 2023. The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

Birgit Pfitzmann. 2024. Personal correspondence. Personal correspondence with Birgit Pfitzmann on 2 September 2024.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Nils Reimers and Iryna Gurevych. 2019. 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.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Rricha Jalota, Floris Fok, Michael Ramos, Ian van Wyk, Abdallah Nasir, Hayden Goldstein, et al. 2024. Climategpt: Towards ai synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*.

Petter Törnberg. 2024. Best practices for text annotation with large language models. *arXiv preprint arXiv:2402.05129*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

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 preprint arXiv:2304.10428*.