Custom Named Entity Recognition VS ChatGPT Prompting: A Paleontology Experiment

Konstantinos Tsitseklis, Georgia Stavropoulou, Symeon Papavassiliou
National Technical University of Athens
Athens, Greece
{ktsitseklis, gstavr}@netmode.ntua.gr, papavass@mail.ntua.gr

***Abstract*—Natural Language Processing is one of the most commonly applied techniques in the context of chatbots. User messages must be analyzed to detect entities and match them to specific intents in order to generate answers. Traditional approaches include Named Entity Recognizers trained over datasets relevant to the scope of the chatbot, utilizing techniques from the field of machine learning. On the other hand, Large Language Models (LLM), with OpenAI's ChatGPT as their spearhead, have gained recently significant attention and increased popularity among both scholars and the general population. Besides their ability to produce text and respond to the users' queries, these models can be instructed to perform certain actions through carefully designed prompts. In this paper, we perform a comparison between a custom built Named Entity Recognizer (NER) that is part of a chatbot designed for operation in the Paleontology Museum of Athens, and ChatGPT. To this end, through a prompt that defines the relevant entities and the rules that should be followed, ChatGPT is instructed to act as a NER designed for the same purpose. From the comparison over commonly employed metrics, we draw useful insights on the current limitations, capabilities and applicability of such models.**

***Index Terms*—NLP, NER, ChatGPT, LLM, Chatbots, prompting**

I. Introduction

Conversational agents, also known as chatbots, have been employed over the years in various use-cases aiming to facilitate users' interactions with online systems by providing them with relevant information, increasing in this way their engagement and quality of experience [1]. Chatbots oftentimes constitute parts of cultural websites and services. In this work, we focus on the case of the Paleontology Museum of Athens, where a chatbot is deployed offering information about the museum's collection to its visitors.

The most common method of interacting with a conversational agent is via text. This text requires analysis that allows specific entities to be extracted from the users' messages in order for their intents to be understood and addressed through the agent's responses. In order to detect specific entities, Named Entity Recognition, a well-known Natural Language Processing (NLP) technique is commonly employed. Named Entity Recognizers (NER) are trained using a dataset of examples consisting of annotated phrases corresponding to different intents and containing the relevant entities that need to be classified. Concerning the case of paleontology related data where no sufficient pre-trained models exist, a custom NER should be developed for addressing the needs of a conversational agent for such a museum, specifically in the context of users requesting for exhibits.

Probably one of the most groundbreaking chatbots released to this day is ChatGPT, developed by OpenAI [4]. Trained with huge amount of data available from the Internet and expanding on previous Language Models, ChatGPT utilizes NLP techniques in order to interpret the users' inputs and respond accordingly. OpenAI offers accessible (via a fee) Application Programming Interfaces (APIs) that the programmers can use to interact with the various available models, including GPT-3.5-turbo that constitutes the basis of ChatGPT [7]. The most intriguing feature of these models is their ability to be "programmed" to perform certain tasks after providing them with specific prompts. Building upon this capability, in this paper, we provide the model with prompts which enable it to detect paleontology related entities in sentences that are possible inputs to the museum's chatbot.

In this work, we present a comparison over commonly employed Machine Learning (ML) performance metrics, between the custom developed NER, trained on an artificially generated dataset, and a prompted operation of OpenAI's GPT-3.5 that is instructed to act as a NER, for which we provide the developed prompt. Moreover, we discuss the benefits and the drawbacks of each one of the presented approaches to the problem of NER in paleontology datasets, highlighting current limitations.

The remainder of this paper is structured as follows. In Section II, we present relevant approaches to the NLP tasks examined in the paper, focusing on cultural use-cases, and works for which LLMs have been employed for NLP tasks. Then, in Section III our custom built mechanism is briefly presented. Following this, in Section IV the prompt employed to OpenAI's API is presented alongside a discussion on generating such prompts. A short comparison of the two models over both English and Greek text is provided in Section V. Finally, Section VI concludes the paper with an insightful discussion on the current benefits and limitations of

This work is conducted within the "Research and Innovation Synergies in the Region of Attica" Action, and is realized within the framework of ESPA 2014-2020, co-financed by Greece and the European Union (European Regional Development Fund), under the project with title "Advanced Tour and Education Services in the virtual Paleontology and Geology Museum / v-Palm (ATTP4-0342153)" (award number: MIS 5185020).

979-8-3503-1884-5/24/$31.00 ©2024 IEEE

the presented methods.

II. Related Work

A. Traditional NER works

NER methods have been deployed in several works aiming to facilitate online museums' services or targeted to cultural purposes. In [14], the researchers developed a NER mechanism by creating a semi-supervised deep learning model employing Bidirectional Long Short-Term Memory (BiLSTM) and Conditional Random Field (CRF) models, trained with limited labeled data and plenty of unlabeled data collected from the National Museum of China. A repeat-labeled strategy is proposed for sample selection and embeddings from the language model (ELMo) are used to boost the model's performance. Despite limited labeled data, the model performs well in the cultural relics NER task, as demonstrated by experimental results and comparisons.

The authors in [3] deployed a BERT-based model consisting of Transformers and CRF layers to perform a NER task on Historical Multilingual Documents from the HIPE Corpus, a collection of digitized documents coming from newspapers, in English, French and German. By slightly altering each time the architecture and parameters of the model proposed, they performed several experiments, achieving satisfactory results in French and German, and eventually in English, by performing transfer learning from the other two languages.

NER tools for historical purposes have also been employed in [12], where the combination of five NER models, i.e., Stanford NER, NER-Tagger, Edinburgh Geoparser, spaCy, and Polyglot-NER, is evaluated concerning the identification of Place Names in Historical Corpora. The study showcased that the ensemble combination of the models for the purpose of entity detection in documents deriving from Mary Hamilton Papers and the Samuel Hartlib collection, outperformed the individual NER systems.

While the aforementioned works demonstrate efficient NER mechanisms for the extraction of entities in several languages and for different purposes, the models proposed have been trained over large datasets. The scarcity of paleontology data and relevant pre-trained models, especially in the context of user requests for exhibits, has led us to deploy a custom NER mechanism trained on a synthetically generated dataset, the details of which are presented in Section III.

B. LLMs used for NLP

Although not quite commonplace yet, ChatGPT has been used for NLP tasks in some cases. In [13] the authors present the current state of NLP in finance and discuss the benefits of employing GPT for tasks such as language understanding, language generation and text-based financial analysis but also consider ethical and legal concerns that arise when using GPT.

The authors in [8] attempt to evaluate GPT and GPT3.5 performance on 7 popular NLP tasks including Named Entity Recognition over commonly used datasets. The NER performance is reported quite low for both models. Although the authors do not disclose the actual prompt provided to the model, the reader is left to understand that they employ zero-shot learning without providing descriptions for each entity type. In this work, while we also do not provide specific inputs with their solutions, we describe what each entity represents in the context of paleontology. VicunaNER, presented in [6], is a NER mechanism based on an open-source LLM called Vicuna. The process has two phases, the first called Recognition while the second one called Re-recognition. In each phase, multiple turn dialogues with Vicuna take place in order to settle to a final entity recognition, remove false positives and detect true positives. This model can work both with zero-shot or few-shot prompting methods.

In [10] the Materials Knowledge Graph (MatKG) is presented. In this work, an LLM is employed in order to classify tokens originating from scientific publications about materials into seven categories and analyze their relationships with the goal to construct a KG that reflects the acquired materials-related knowledge. The authors use a very specifically trained LLM, MatBERT [11], in order to detect entities in the topic of material science, while in this paper we opt to study a general LLM for a specific task.

Facing the problem of training large models for specific tasks that require vast amounts of annotated data, the authors in [5] propose the usage of the PaLM 2 LLM [2], coupled with human experts in order to rapidly create labeled datasets. Their findings match our own regarding the temperature parameter of the model as well as the quality of results achieved solely by prompt engineering, without having to necessarily fine tune the model's parameters.

III. Description of custom NER mechanism

The detailed description and performance evaluation of the custom NER mechanism deployed, is presented in [9]. It is built as an integral component of a chatbot designed for the Paleontology Museum of Athens to perform both in English and Greek. The NER component has been designed to extract information from user messages entered through the chatbot's UI, when requesting for exhibits by their characteristics, which can be classified into different types of entities. The entities extracted through the NER component, enable the execution of relevant queries to the Knowledge Graph (KG) storing exhibit's information, to retrieve the items illustrating the requested characteristics. The entities of interest, related to the exhibits' characteristics that the mechanism has been trained to detect, are:

- Animal type: The type of animal the exhibit represents
- Body part: The exhibit's corresponding body part
- Habitat type: The habitat where the animal lived
- Location: The exhibit's discovery location
- Age: The paleontological period related to the exhibit.

To train the NER component, a dataset containing examples of possible user's requests for an exhibit based on some of its characteristics, has been generated and fed to the model, in both English and Greek. Instead of providing the explicit names of the characteristics in each sentence of the dataset, placeholders have been employed along with lists of potential values (e.g., I would like to see % s of % s from % s → I would like to see antlers of deer from Crete, I would like to see skulls of bears from Pikermi, etc). This process has achieved the generation of a vast amount of data targeted to a specific use-case, overcoming the lack of data available in the paleontology domain, especially in the Greek language, and the inadequate performance of the pre-trained models available, in the context described. The words replacing the placeholders have been annotated as entities of specific type (Animal type, Body Part, Habitat type, Location, Age) with the help of spaCy. The available pipelines ”en_core_web_lg” for English and ”el_core_news_lg” for Greek have been used and early stopping parameters have been set, to avoid overfitting.

The evaluation of the model, which is presented in the aforementioned paper, has been performed by a test set consisting both of heterogeneous sentence forms and lists of entity values that were not part of the model's training data, as well as data formats more familiar to the model. In this paper, a subset of this test set has been used for the comparison of the custom NER mechanism with ChatGPT.

IV. Prompting OpenAI

In Fig. 1 the prompt employed for OpenAI is presented. It is important to notice that the prompt is provided in a structured manner that clearly states the task to the model and informs it about the rules that should be followed in order for the NER task to be completed successfully. In more detail, the prompt consists of two major parts. In the first part of the prompt titled “NER”, details about which entities should be detected are given. It should be noted that only a brief description is enough for the model to understand the different entity types and distinguish them. Also, we opted not to few-shot the language model (i.e., do not provide it with specific examples and their solutions). Focusing on the second part of the prompt concerning the ”rules”, we inform the model that any input should be given inside specific tags. The model understands that each sentence contained inside the tags is input for the specified task and should not be treated in any other way. For example, in the sentence “I want to learn about dinosaurs”, the model should detect “dinosaur” as an entity “ANIMAL_TYPE” if the sentence is provided inside the specified tags, instead of it actually providing us with information about dinosaurs. Finally, the output format for the detected entities in each sentence is provided also as a rule that must be followed.

In Fig. 2, an overview of our approach for prompting and evaluating the LLM can be seen. In particular, each sentence included in the test dataset is appended to the instructions seen in Fig. 1 and fed to ChatGPT. The response is then saved alongside the responses from the other sentences.

```
”task”: ”Act as a Named Entity Recognizer
known as NER. Detect the described entities
and follow the rules.”
, ”NER”:
”entities”:[
”ANIMAL_TYPE (Denotes the
type of animal of the exhibit.)”,
”BODY_PART (The exhibit’s corresponding body part.)”,
”HABITAT_TYPE (The habitat where
the corresponding animal
lived (i.e, lake, land, mixed, forest)”,
”AGE (The paleontological period in which
the exhibit belongs. May be given as a name
of the period (like Precambrian) or in years
(e.g. ’50 million years’).)”,
”LOCATION (The geographical location
where the exhibit was discovered.)”
],
”rules”:[
”The input is provided inside the tags <phrase >and
</phrase >.”,
”The input is a single sentence”,
”Do not alter the provided text in your response”,
”The output is given in a single line
as <entity_type >: <detected_entity >, <entity_type2
>:<detected_entity2 >, ...”]
```

Fig. 1. Structured Prompt for OpenAI

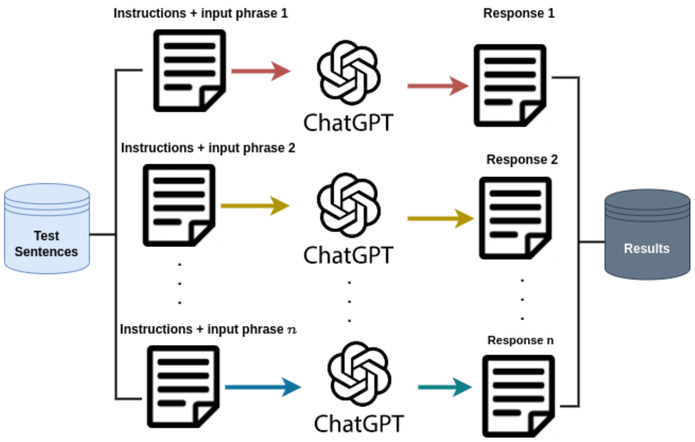
Instructions + input phrase 1
Response 1
ChatGPT
Instructions + input phrase 2
Response 2
ChatGPT
Test Sentences
Results
Instructions + input phrase n
Response n
ChatGPT

Fig. 2. Prompting ChatGPT

V. Evaluation

A. Experimental Setup

In order to compare the two approaches presented here for NER purposes, two datasets were generated, one in English and one in Greek, both containing 1000 artificially generated phrases (different than the ones that the custom NER mechanism was trained on), written in the style of a user requesting more information from a chatbot. The main reasons behind compiling small datasets were the recurrent time-outs, concerning the ChatGPT case, as well as service unavailable exceptions that abruptly terminated the scripts with no capability of recovery. Moreover, there is the issue of pricing that forces us to keep the cost low. It is important to note that initially, an approach in which the sentences were provided to ChatGPT in batches, in the form of a list, was followed but it suffered from the fact that the output's form, to the best of our efforts, was not consistent and required a lot of manual editing in order to compile a file that could be used for evaluation purposes. The sentences given as input to both models were created by combining several possible values of the entity types in predefined sentence "casts", as described in Section III.

In order to collect the responses from OpenAI, we used the provided API, submitting each time a single sentence. To mitigate the number of exceptions due to the large amount of traffic directed to the API we enforced a sleep period of 3 seconds per request. For the custom NER mechanism the test sentences were given one by one to the already trained model and the responses were collected into a text file. In the following, we compare the two methods over commonly employed metrics.

B. Comparison of models

Tables I and II present the results obtained for the two compared NER mechanisms for the English and Greek languages respectively. The employed metrics are the commonly applied to such tasks: precision, recall and F1-score.

Those were computed as follows:

$$Precision = \frac{\text{Number of correctly labeled entities}}{\text{Total number of detected entities}} \quad (1)$$

$$Recall = \frac{\text{Number of correctly labeled entities}}{\text{Number of ground truth entities}} \quad (2)$$

$$\text{F1-score} = \frac{2 \cdot Presicion \cdot Recall}{Precision + Recall} \quad (3)$$

As a general remark, we can see that both models perform better in the case of English sentences. This is expected, since NLP techniques are more advanced for the English language, while other less popular languages like Greek, face challenges such as, for example, poor lemmatizer performance. Also, it is highly probable that English documents are more represented in the ChatGPT's training data than documents in Greek.

The total scores for the English language show that the custom developed NER mechanism outperforms prompting across all metrics and most prominently in the Recall metric case. The evaluation scenario with Greek phrases presents more balanced results, with the NER mechanism outperforming ChatGPT in the Recall and F1-score metrics, while falling slightly behind in the Precision metric. The overall better performance of the custom mechanism indicates that, to this day, for specific use-cases, traditional machine learning techniques applied on models trained in domain-specific datasets are still capable of outperforming generic Large Language Models.

Besides the above more general conclusion, a closer look on how the models behave when detecting specific entities can reveal more interesting insights. For example, taking a look at the most prominent case in which ChatGPT suffers, we see that if fails in detecting "BODY_PART" entities. Instead of detecting the corresponding body parts, the model included them in the "ANIMAL_TYPE" entity. For example, the model detects as an animal "elephant tusk" instead of just "elephant". This leads to very low recall scores, thus contributing to the overall low recall score both in English and in Greek. Moreover, some cases in which the prompting model could not differentiate between two different entities that have a degree of semantic similarity were identified. These are the "HABITAT_TYPE" and the "LOCATION" entities. Although in the prompt provided through the API, the differences of these entities are explained via their descriptions and additional examples are given for the "HABITAT_TYPE" case (forest, lake, land, etc.), cases exist in the answers where "LOCATION" entities are classified as "HABITAT_TYPE". Concerning the low precision observed regarding the custom NER performance on the detection of the "AGE" entity in Greek, we found out that several entities belonging to different types were misclassified as "AGE" entities. The values of this entity type were often heterogeneous, comprising one or a few words and as a result the system classified low-confidence entities of other types to this entity type.

Another interesting aspect is that although ChatGPT was instructed specifically not to alter the provided text, there exist cases where it did not follow this rule. For example, there exist numerous cases in the Greek evaluation scenario where the response was given in the English language and although the translation was correct, it was considered an error. Moreover, a common modification of the provided text in Greek was the change of the case of a noun. Finally, for the English dataset, in a certain sentence it corrected the misspelled country "Kazahkstan" to the correct form of "Kazakhastan" and it was also considered an error. This is an indication that despite providing the model with clear and simple rules, these are not always followed.

TABLE I
NER COMPONENTS' PERFORMANCE IN ENGLISH

	Custom NER / ChatGPT		
	Precision (P)	**Recall (R)**	**F1-Score (F)**
Total	**97.50** / 94.00	**93.47** / 66.35	**95.44** / 77.79
Animal Type	**99.05** / 90.01	86.90 / **87.94**	**92.58** / 89.01
Habitat Type	**88.89** / 60.38	91.43 / 91.43	**90.14** / 72.73
Age	82.93 / **100**	91.89 / **94.59**	87.18 **/97.22**
Body Part	94.69 / **99.47**	**93.80** / 5.38	**94.24** / 10.12
Location	**99.68/** 99.46	**100** / 97.56	**99.84** / 98.50

TABLE II
NER COMPONENTS' PERFORMANCE IN GREEK

	Custom NER / ChatGPT		
	Precision (P)	**Recall (R)**	**F1-Score (F)**
Total	90.81 / **91.86**	**84.92** / 63.31	**87.77** / 74.96
Animal Type	**95.64** / 89.51	85.79 / **87.02**	**90.45** / 88.25
Habitat Type	**78.79** / 47.62	86.67 / **100**	**82.54** / 64.52
Age	32.39 / **83.33**	**100** / 86.96	48.94 **/85.11**
Body Part	85.92 / **86.21**	**72.86** / 2.82	**78.85** / 5.45
Location	95.20 / **97.79**	**94.99** / 94.25	95.10 / **95.99**

VI. Discussion and conclusion

In this paper, we performed a comparison between a traditionally trained NER mechanism based on Machine Learning techniques and prompting a LLM in zero-shot mode, in particular ChatGPT-3.5. The comparisons were performed over artificially generated datasets that represented possible input phrases to a chatbot operating in a paleontology museum's website. Despite the relatively small size of the datasets, the results obtained highlighted the strong aspects of developing and utilizing a custom NER for a specific task but also provided an indication of the capabilities of LLMs and in particular ChatGPT, for the same purpose.

Currently, there exist several limitations that discourage the usage of LLMs in the context of such specialized applications like domain specific NER. In the case of OpenAI's ChatGPT, these included the frequently limited availability of the API, resulting in exceptions and time-outs that terminated abruptly our scripts leading to incomplete datasets and higher cost, since the experiment would have to be repeated. In addition, there is also the cost factor that should be taken into consideration when deciding on a solution. The usage of OpenAI APIs comes with a cost, with higher prices for the more sophisticated models. Of great importance is also the fact that these models are not consistent throughout multiple executions, even when setting the temperature hyperparameter equal to 0, instructing in this way the model to produce more deterministic responses with the least amount of randomness or diversity. Different outputs may occur for the exact same sentence, because a rule was ignored, inducing in this way uncertainty for the owner of a platform.

Despite these facts, it remains impressive that just by compiling a short prompt with some details and a handful of rules written in plain English, even a non expert on the field can achieve satisfactory results on a NER task using a LLM, even without presenting the model with any examples (zero-shot). This paves the way to novice users with limited scientific background to perform complex tasks. Additionally, the use of such models that are pre-trained over vast datasets alleviates the data scarcity problem that might be faced by researchers especially in niche fields such as paleontology, where readily available data to train models are not easy to obtain.

Conclusively, we should not yet completely rely on LLMs if our goal is to get highly accurate and above all consistent results for complex and domain specific NLP tasks, however LLMs might be efficient for less specialized use cases.

As future research steps, we plan to acquire more accurate results from the LLM-based NER task by fine tuning our prompts. Moreover, we aim to deploy a chatbot that employs LLMs both for NER purposes but also for crafting well-structured replies to the users by combining the appropriate elements derived through the queries to the KG that follow the NER task.

References

[1] E. Adamopoulou and L. Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020.

[2] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] E. Boros, E. L. Pontes, L. A. Cabrera-Diego, A. Hamdi, J. G. Moreno, N. Sidère, and A. Doucet. Robust named entity recognition and linking on historical multilingual documents. In *Conference and Labs of the Evaluation Forum (CLEF 2020)*, volume 2696, pages 1–17. CEUR-WS Working Notes, 2020.

[4] Chatgpt. https://chat.openai.com. Accessed: 01-11-2023.

[5] A. Goel, A. Gueta, O. Gilon, C. Liu, S. Erell, L. H. Nguyen, X. Hao, B. Jaber, S. Reddy, R. Kartha, et al. Llms accelerate annotation for medical information extraction. In *Machine Learning for Health (ML4H)*, pages 82–100. PMLR, 2023.

[6] B. Ji. Vicunaner: Zero/few-shot named entity recognition using vicuna. *arXiv preprint arXiv:2305.03253*, 2023.

[7] Openai api. https://openai.com/blog/openai-api. Accessed: 01-09-2023.

[8] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

[9] K. Tsitseklis, G. Stavropoulou, A. Zafeiropoulos, A. Thanou, and S. Papavassiliou. Recbot: Virtual museum navigation through a chatbot assistant and personalized recommendations. In *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '23 Adjunct, page 388–396. ACM, 2023.

[10] V. Venugopal, S. Pai, and E. Olivetti. Matkg: The largest knowledge graph in materials science–entities, relations, and link prediction through graph representation learning. *arXiv preprint arXiv:2210.17340*, 2022.

[11] L. Weston, V. Tshitoyan, J. Dagdelen, O. Kononova, A. Trewartha, K. A. Persson, G. Ceder, and A. Jain. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of Chemical Information and Modeling*, 59(9):3692–3702, 2019. PMID: 31361962.

[12] M. Won, P. Murrieta-Flores, and B. Martins. Ensemble named entity recognition (ner): evaluating ner tools in the identification of place names in historical corpora. *Frontiers in Digital Humanities*, 5:2, 2018.

[13] A. Zaremba and E. Demir. Chatgpt: Unlocking the future of nlp in finance. *Available at SSRN 4323643*, 2023.

[14] M. Zhang, G. Geng, and J. Chen. Semi-supervised bidirectional long short-term memory and conditional random fields model for named-entity recognition using embeddings from language models representations. *Entropy*, 22(2):252, 2020.