Entity Name Recognition: an LLM-friendly set-based approach to Knowledge Extraction

Gianmarco Pappacoda[a,*]

[a]Alma Mater Studiorum — Università di Bologna, Department of Computer Science and Engineering (DISI)
ORCID (Gianmarco Pappacoda): https://orcid.org/0009-0001-6609-4156

Abstract. Knowledge Extraction (KE) is the process of extracting knowledge from data. Until recent advances in NLP, KE was mostly confined to structured data. With the advent of Large Language Models (LLMs), performing KE on unstructured data such as natural language documents is gaining momentum. In Natural Language Processing (NLP) core tasks such as Named Entity Recognition are close to KE. However, the tasks definitions are biased towards discriminative methods and not well-suited for generative ones. Moreover, models are usually domain-specific and work in a closed setting. In this paper we reframe tasks towards a more general, knowledge-oriented and generative-friendly way in order to explore existing capabilities of LLMs in core NLP tasks. In addition, we propose fine-tuning as a method to instruct LLMs to learn these skills. Our findings suggest that base models are unsuited to the tasks, while fine-tuning is a good method to instruct LLMs to perform Knowledge Extraction. Furthermore, the use of generative methods on these tasks opens up the way to open-world and domain-free models.

1 Introduction

Human knowledge is a valuable asset. Historically, knowledge is passed on using language, for example, through books. To leverage such knowledge, humans have to read. Unlike humans, machines struggle with knowledge, and up until recently were only effective at extracting information from structured data. The first approaches to knowledge extraction from unstructured data stem from Information Retrieval, specifically Information Extraction.

The last decade's progress in NLP and the advent of LLMs created new possibilities for knowledge extraction. However, the conceptualization of knowledge extraction tasks in NLP is, to some extent, tailored to pre-LLM approaches, creating a contrast between generative AI systems and knowledge extraction tasks and resources.

In this paper we investigate knowledge extraction under the lens of generative AI and address the following research questions:

- What are the current capabilities of existing pre-trained LLM in performing Knowledge Extraction?
- Is prompting enough to elicit the desired behaviour in pre-trained LLMs?
- Is there a way we can teach/enhance such capabilities?

2 Tasks

The adaptation of traditional NLP core tasks such as Named Entity Recognition and Relation Extraction poses an interesting problem for generative models due to tasks definitions. These definitions were conceived when NLP techniques were different from today's, and with a closed world in mind. Previous methods used sequence labelling or span-based extraction framing, which originated in classification, where each token or phrase is assigned a label. By contrast, today's generative models can generate words not found in the original input, therefore escaping the closed world and posing the problem of evaluating such outputs. The available technology also lead to a conceptualization of knowledge extraction which agglomerates multiple tasks. For example, NER, defined as a sequence-labeling task, in principle could be conceptualized as the combination of two separate tasks: Named Entity Identification (NEI) and Named Entity Classification (NEC).

In the rest of the section we reframe these tasks in a more general set-oriented way in order to capture the behaviour of modern generative models. Moreover, we split the tasks into subtasks, decoupling identification and classification.

2.1 Entity Name Recognition (ENR)

Named Entity Recognition (NER) is a long-standing task in NLP and arguably one of the most studied [7]. With the advent of LLMs there have been numerous attempts using generative models [9]. One of the earliest approach is GPT-NER [8] which uses a generative model [1] to generate strings with delimited entities. Other relevant approaches include GLiNER [10] which uses BERT [3] as backbone achieving SOTA performance and UniversalNER [11]. These attempts all have in common the idea of moving from closed Information Extraction to open Information Extraction. In its most common formulation, NER is framed as a sequence labelling task, most often using IOB tagging [5] or as a span-based extraction task. However, these definitions fail to capture the flexibility of modern generative language models, as the output is strictly dependent on the surface string, thus on the words forming up a named entity.

Example 1 (Similar names)**.**
Text*: "John F. Kennedy was president of the United States"*
GT*: {"John F. Kennedy", "United States"}*
Possible output*: {"John Kennedy", "US"}.*

Ground truth and output refer to the same concept, however due to the task definition the answer would be deemed incorrect. A tra-

[*] Corresponding author. Email: gianmarco.pappacoda@unibo.it.

ditional model, instead would produce the exact same surface string and thus be less penalized. To address the limitations of the list-based definitions, we propose a parallel set-based definition in which we identify Entity Names as unique identifiers. This definition allows us better to capture the generative part of modern language models and form a basis for a more general and knowledge-oriented framing of the NER task. We emphasize that NER and ENR are different tasks. NER identifies and classifies all entities in a text, while ENR identifies and classifies unique entities names, abstracted from the text.

2.2 Entity Name Identification (ENI)

In this task we are concerned with finding Entity Names from text. The model should recognize the Entity Names without identifying all the instances, which is the main difference between ENI and NEI. Moreover, the type of the Entity Name should not lead to a duplicate entity at this stage. As in Example 2 on page 2 "JFK" should not appear twice.

2.3 Entity Name Classification (ENC)

In this task we provide the text as well as the Entity Names contained therein as input, and we expect to have the Entity Names be cast and associated to types as output. This step is crucial as it allows for concepts that appear in text with multiple instances to be correctly framed with respect to their type.

Example 2.
Input text: *"John Kennedy, also called JFK served as United States president ... The homonymous JFK airport ..."*
Input names: *{"JFK", " United States"}*
Output: *{("JFK", PER), ("JFK", LOC), ("United States", LOC)}*.

In this definition, we don't specify types beforehand, allowing generative models to be creative about new types. Since the task depends on associated types, we also define a variation called *ENC+t* that provides the set of types to choose from. This variation is used to guide modern language models to follow pre-defined ontologies.

3 Experiments

In order to investigate the performance of Pretrained Language Models (PLM) in the proposed tasks related to our research questions, we propose the following experiments. Each experiment involves the use of all the selected models applied to one task. We consider for each task the following models: a base SLM, an instruction-tuned version of it, a fine-tuned version of the instruction-tuned one, two SOTA LLMs.

Experiment description In each experiment, the model is prompted to extract relevant elements and returning structured output. We enforce strict rules for output formatting using an XML-like structure defined by a Context-Free Grammar. We instruct the models to use additional tokens to delimit extracted knowledge, which is crucial for fine-tuning and evaluation to match the ground turht. For the fine-tuned SLM we utilize QLoRA [2] and train a different model for each task.

Models We selected the Llama 3 family of models [4] as prominent decoder-base architecture. The chosen SOTA models are GPT-4.1-mini and GPT-4.1.

Dataset The *CoNLL-2023* [7] is a standard dataset in NLP literature for NER. It is a manually annotated dataset and is pre-split into three segments. We use only the English part of the dataset which contains $\approx 300'000$ tokens across the three splits, the distribution of entities is the same across each of the splits. The dataset is in CoNLL format, a type of IOB [5]. We converted it to a set-based format that we refer to as "T2G", where each entity is a tuple. The first element is the entity name made up of original text's tokens, the second is the type.

Metrics We use the metrics from SemEval 2013 Task 9 [6]. More specifically: Partial boundary for ENI; Type for ENC; Strict for ENR. We use macro average for aggregation as for our research questions it is more important to understand the overall performance of the classifier rather than the performance per-class. Moreover, since we moved to a set-based approach, we penalize over/undergenerated entities.

Experimental setup We set each experiment with the same hyper-parameters, fixing the seeds (6, 42, 1234) and averaging across the three. The most prominent hyperparameter being temperature $= 0.5$. For fine-tuning: LoRA: $r = 64$, $\alpha = 128$, dropout $= 0.05$, 4bit quantization with NF4 format; epochs $= 3$, warmup equaling 10% of the training process. Experiments carried out using Nvidia® RTX 4090.

4 Results and discussion

model	ENI (partial)	ENC (type)	ENR (strict)
Llama-3.2-1B	0.019 ± 0.003	0.001 ± 0.000	0.000 ± 0.000
gpt-4.1-mini	0.647 ± 0.002	0.021 ± 0.001	0.030 ± 0.004
gpt-4.1	0.672 ± 0.002	0.025 ± 0.006	0.001 ± 0.000
Llama-3.2-1B-FT	**0.709 ± 0.002**	**0.712 ± 0.000**	**0.664 ± 0.001**

Table 1.

Base PLM have shown no potential with respect to examined tasks, thus we omit reporting results. This is likely due to the fact base PLMs are trained to complete the input given by the user rather than executing instructions.

Instruction-tuned PLM have shown little potential. Most of the times the obtained output is not structured enough to be parsed correctly, therefore producing scarce results.

SOTA PLMs are capable of producing structured results without fine-tuning, however failed in ENC/ENR due to mismatches between expected/predicted types (e.g. PER and Person).

Fine-tuned PLM have shown the most potential. The models could reliably perform the proposed tasks.

These experiments results show that a smaller SOTA model can complete the tasks, but a specialised SLM can surpass them, suggesting the domain is crucial. The main problem with the tasks lies in correctly classifying types. The SOTA LLMs failed at ENC due to the fact they cannot access the correct types. This can be solved by performing ENC+t or defining metrics that use substring matching instead of exact matches for types.

5 Conclusion and future work

This paper investigates new formulations of classical NLP tasks to account for the generative nature of modern LMs.

We have conducted an initial experimentation with a SLM (1B) LLM and two SOTA LLMs. SLMs have not exhibited the desired behaviour using prompting alone, however we are led to believe that models of increasing size may yield better results.

As this is a preliminary study, in the future we plan to enrich the conceptualization with a formal definition, as well as a definition of an alignment task between ENR/NER and ENI/NEI.

Furthermore, we plan to expand our experiments with more emphasis on ENC/ENC+t and a comparison with SOTA methods for NER. Moreover, we will define new metrics that better capture the generative nature of LLMs as the existing ones over-penalized SOTA PLMs and offered little insight into their true performance.

Acknowledgements

Thanks to DecisionBrain for providing computational resources to run the experiments.[1]

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[2] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc., 2023.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[4] A. Grattafiori et al. The Llama 3 Herd of Models, Nov. 2024.

[5] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995.

[6] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In S. Manandhar and D. Yuret, editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[7] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[8] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, G. Wang, and C. Guo. GPT-NER: Named entity recognition via large language models. In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275, Albuquerque, New Mexico, Apr. 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.239.

[9] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen. Large Language Models for Generative Information Extraction: A Survey, Oct. 2024.

[10] U. Zaratiana, N. Tomeh, P. Holat, and T. Charnois. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.300.

[11] W. Zhou, S. Zhang, Y. Gu, M. Chen, and H. Poon. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition, Jan. 2024.

[1] https://decisionbrain.com