Does Prompt Engineering Help Turkish Named Entity Recognition?

Arda Serdar Pektezol
Department of Computer Engineering
MEF University
İstanbul, Türkiye
pektezola@mef.edu.tr

Ahmet Batuhan Ulugergerli
Department of Computer Engineering
MEF University
İstanbul, Türkiye
ulugergerlia@mef.edu.tr

Volkan Öztoklu
Department of Computer Engineering
MEF University
İstanbul, Türkiye
oztokluv@mef.edu.tr

Seniz Demir
Department of Computer Engineering
MEF University
İstanbul, Türkiye
demirse@mef.edu.tr

2024 9th International Conference on Computer Science and Engineering (UBMK) | 979-8-3503-6588-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/UBMK63289.2024.10773605

***Abstract*—The extraction of entity mentions in a text (named entity recognition) has been traditionally formulated as a sequence labeling problem. In recent years, this approach has evolved from recognizing entities to answering formulated questions related to entity types. The questions, constructed as prompts, are used to elicit desired entity mentions and their types from large language models. In this work, we investigated prompt engineering in Turkish named entity recognition and studied two prompting strategies to guide pre-trained language models toward correctly identifying mentions. In particular, we examined the impact of zero-shot and few-shot prompting on the recognition of Turkish named entities by conducting experiments on two large language models. Our evaluations using different prompt templates revealed promising results and demonstrated that carefully constructed prompts can achieve high accuracy on entity recognition, even in languages with complex morphology.**

Keywords- Named entity recognition, prompt engineering, Turkish

I. Introduction

Named entity recognition (NER) is one of the fundamental tasks in natural language processing. The aim is to identify and classify entity mentions (anything that can be referred to using a proper noun) in a given text using predefined classes such as person, location, and organization. Recognizing mentions plays a crucial role in a variety of applications including information extraction and question answering. The methodologies used to recognize mentions have evolved from rule-based systems [1] to transformer-based learning systems [2] which were shown to outperform long short-term memory (LSTM) networks and conditional random fields (CRF) [3].

Pre-trained large language models LLMs (such as BERT and GPT) have led a revolution in the way how language processing tasks are handled. The models which encode the nature and characteristics of a language can be integrated into or fine-tuned to different domains and downstream tasks including named entity recognition. NER has been traditionally formulated as a sequence labeling problem [4]. However, recent advancements in LLMs have opened a new pathway to address NER as a machine reading comprehension (MRC) task [5]–[8]. In MRC setting, the language model is expected to first comprehend a given text and then answer questions (queries) about the text by exploiting existing data efficiently. To utilize MRC in named entity recognition, a query (q_y) is formed for each entity type (y) and answer fragments (q_y, $x_{start:end}$, X) are returned by the LLM for the given input text X, where $x_{start:end}$ corresponds to an annotated entity mention. Notably, MRC formulation tackles some well-known challenges of sequence labeling based NER approaches such as nested entity mentions (overlapping entities with different types) and the cost associated with collecting data and training systems accordingly. MRC setting also eliminates the need of fine-tuning or training a language model in deep-learning based NER approaches.

In order to harness the full potential of LLMs, task specific queries (prompts) should be carefully crafted. Prompts which are natural language instructions elicit accurate and relevant responses from LLMs without modifying or tuning model parameters. The process of forming a prompt (prompt engineering) thus has become an essential technique for achieving optimized model performances [9]. Structuring an appropriate prompt is not straightforward since a prompt should be clear and unambiguous to minimize the risk of generating irrelevant responses, specific enough to retrieve targeted responses, and contextualized (enhanced with context information) to obtain more coherent responses. This process has an iterative nature in that prompts need to be continuously refined by leveraging feedback from model's outputs. Nonetheless, prompt engineering requires careful consideration of the challenges involved such as handling language-specific differences and model dependencies. For instance, different models might respond variably to the same prompt, requiring extensive experimentation to find the optimal prompt for each model.

In this work, we explored the use of prompt engineering in Turkish named entity recognition where the task is formulated as MRC. For this purpose, we formed a series of task-specific prompts and assessed their effectiveness on two state-of-the-art large language models, namely Google's Gemini 1.5 and OpenAI's ChatGPT 3.5. These prompts allowed us to exploit LLMs' ability to address entity recognition in zero-shot (no prior examples are given to the model) and few-shot (a few examples are provided to help the model understand the instruction better) settings [10], [11]. We did not perform prompt-tuning (fine-tuning prompt parameters) or model tuning during evaluations. For the experiments, we used a publicly available dataset [12] where the texts are manually annotated with three entity types (person, location, organization). Rather

979-8-3503-6588-7/24/$31.00 ©2024 IEEE

than crafting a distinct prompt for each entity type, we formed a single prompt that targets all types. The annotations generated by LLMs were compared to human annotations using f-measure scores. Our results demonstrated that LLMs achieve an f-score between 0.80 and 0.90 in matching human annotations and yield the highest scores in recognizing location entities. Recently, prompt engineering has been investigated in generating paraphrases [13] and recovering surface forms of sentences from treebank annotations [14] for Turkish. To our best knowledge, this work is the first that studies prompt engineering in Turkish named entity recognition using zero-shot and few-shot settings. Our work is part of an ongoing project whose main objective is to answer following research questions:

- In a low-resourced language Turkish, can prompt engineering be used as an annotator to aid human annotators and minimize related costs?
- Can MRC be applied effectively to token level tasks in morphologically-rich language Turkish?
- Is zero-shot or few-shot prompting setting more effective on multilingual LLMs that cover Turkish?

The rest of the paper is organized as follows. Section II presents related work on the use of prompt engineering in named entity recognition. Section III introduces prompt templates and the dataset used in the study. Section IV discusses the results of the experiments. Section V concludes the paper and presents future work.

II. Related Work

In the literature, prompt engineering research has followed two main directions to predict entity mentions and identify their types. In the first approach, a language model is given with a list of possible spans extracted from a text and asked to predict entity class labels of these text spans. An earlier study [15] used text spans up to eight tokens of a given sentence to populate templates of the form '[Span$_x$] is a [Type$_k$] entity' and these templates were ranked by a sequence-to-sequence network where BART being used as the language model. The proposed approach achieved 92.55 f-measure score on the CoNLL03 dataset. However, in the second approach, a language model is given an entity class label and asked to locate entity mentions that belong to that class in a given text. The study of Liu et al. [16] generated prompts of the form 'What is [Type$_k$]?' and used the given sentence as the context in a question answering architecture with BERT as the language model. Despite their effectiveness, both approaches suffer from high computational costs such as creating multiple templates for each text span. A recent study [17] addressed the need for multi-round prompting by combining entity locating and type identification tasks in a single prompt template design. For a given text, a fixed number of dual-slot templates in the form of '[Position$_x$] is a [Type$_k$] entity' was used in an encoder-decoder architecture. The approach demonstrated an f-measure score of 92.41 with BERT-large and 93.08 with RoBERTa-large as the language model on the CoNLL03 dataset. Several other research works also exploited the use of prompting in named entity recognition [18]–[21].

Previous research has also studied the effect of different prompts on named entity recognition once the task is formulated as MRC [22], [23]. Our work is closely related to a prompting study conducted on another morphologically rich language Russian [24]. Although the effectiveness of multiple prompts were evaluated in that study, Russian NER dataset contained nested entity mentions and hence a particular attention was devoted to address nested mentions in prompt design. In addition, we followed a different path in selecting example keywords and sentences to be used in prompt templates.

III. Methodology

A. Prompt Templates

In order to optimize LLMs' responses, we investigated a range of prompts with varying levels of specificity and details as follows:

- **Instruction Prompt:** This prompt contains only the instruction (the description of the task) that the model has to follow. *"Act as a Turkish Named Entity Recognizer. For the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@. Show the annotated texts."*

- **Definition Prompt:** This prompt contains the instruction and the definition of each named entity class (formed according to the interpretations given in a vocabulary).
"Act as a Turkish Named Entity Recognizer. You are given a set of definitions.
Definition: person refers to names of individuals, including first names, last names, and full names.
Definition: location refers to names of places, whether specific or general including cities, countries, regions, and more.
Definition: organization refers to companies, institutions, agencies, organizations, and groups.
Using these definitions, for the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@. Show the annotated texts."

- **Keyword Prompt:** This prompt contains the instruction and three keyword examples for each named entity class (randomly selected from a dataset).
"Act as a Turkish Named Entity Recognizer. For the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@.
Person is an entity such as Ahmet, Mehmet Bey, Ayşe Dündar.
Location is an entity such as İstanbul, Marmara Bölgesi, Türkiye Cumhuriyeti'nin.
Organization is an entity such as Futbol Federasyonu, Planlama Komisyonu, Ekonomik Politikalar Araştırmaları Bölümü.
Show the annotated texts."

- **Definition + Keyword Prompt:**: This prompt contains the instruction, and the definition and three keyword examples of each named entity class.

- **Contextual Prompt:** This prompt contains the instruction and two example sentences for each named entity class (randomly selected from a dataset) where named entities are not explicitly marked. The prompt

allows the language model to interpret surrounding contexts of named entities in order to provide more accurate responses.
"Act as a Turkish Named Entity Recognizer. For the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@.
Here are example sentences with a person named entity: '21 Mart 2014'ten bu yana komada olan Kenan Işık, dün sabah hayatını kaybetti.' 'Yenilmez, ortaokul öğrenimi gerçekleştirdiği dönemde amatör tiyatro yapmaya başladı.'
Here are example sentences with a location named entity: 'Sanatçı memur bir ailenin çocuğu olarak 1947'de Malatya'da dünyaya geldi.' 'ABD'nin yerel tiyatrolarını ziyaret etti.'
Here are example sentences with an organization named entity: 'Sanatçı, 2000 yılında Devlet Tiyatroları'ndan emekli oldu.' 'Son yıllarda meydana gelen değişimler sonucunda Türkiye İstatistik Kurumu'nun politikası güncellendi.'
Show the annotated texts."

- **Single Masked Prompt:** This prompt contains the instruction and two example sentences for each named entity class (randomly selected from a dataset) where only one named entity is replaced with its entity class label. The prompt allows the language model to implicitly interpret the use of each named entity class in surrounding texts.
"Act as a Turkish Named Entity Recognizer. For the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@.
Here are example sentences with a person named entity: '21 Mart 2014'ten bu yana komada olan <u>person</u>, dün sabah hayatını kaybetti.'...
Here are example sentences with a location named entity: 'Sanatçı memur bir ailenin çocuğu olarak 1947'de <u>location</u> dünyaya geldi.'...
Here are example sentences with an organization named entity: 'Sanatçı, 2000 yılında <u>organization</u> emekli oldu.' ...
Show the annotated texts."

- **Multiple Masked Prompt:** This prompt contains the instruction and two example sentences for each named entity class (randomly selected from a dataset) where all named entities are replaced with their entity class labels.

- **Framed Prompt:** This prompt contains the instruction and two example sentences for each named entity class (randomly selected from a dataset) where all named entities are replaced with tuples in the form of @named entity - type@.
"Act as a Turkish Named Entity Recognizer. For the following Turkish texts, find named entities that correspond to person, location, or organization. Annotate the named entities with their types in the texts as @named entity - type@.
Here are example sentences where named entities that correspond to a person are annotated: '20. yüzyılın son genel seçimlerinde de daha önce olduğu gibi merkez sağın lokomotifi Rekortmen Başbakan @Helmut Kohl - person@ ile 16 yıldır muhalefette olan Sosyal Demokratların başbakan adayı @Gerhard Schröder - person@ mücadele edecek.' ...
Here are example sentences where named entities that correspond to a location are annotated: 'Cuma sabahı @Çavuşbaşı Köyü - location@ ile @Ümraniye Mezarlığı'nı - location@ birleştiren yolda , göletin ardındaki yoğun ormanlık alandan siyah ve çok yoğun bir duman çıktığını gören şöförüm , beni uyararak yolda durdu.'...
Here are example sentences where named entities that correspond to an organization are annotated: '@TBMM - organization@ Başkanı, @Devlet Bakanlığı - organization@ ve milletvekilliğinden istifasını değerlendirirken gerekeni yaptım değerlendirmesini yaptı.' ...
Show the annotated texts."

- **Multiple Masked + Framed Prompt:** This prompt contains the instruction, two example sentences for each named entity class where all named entities are replaced with their entity class labels, and two other example sentences for each named entity class where all named entities are replaced with tuples in the form of @named entity - type@.

- **Definition + Multiple Masked Prompt:** This prompt contains the instruction, the definition of each named entity class, and two example sentences for each named entity class where all named entities are replaced with their entity class labels.

- **Definition + Framed Prompt:** This prompt contains the instruction, the definition of each named entity class, and two example sentences for each named entity class where all named entities are replaced with tuples in the form of @named entity - type@.

B. Dataset

The dataset used in this study [12] consists of articles collected from a national newspaper Milliyet between the period of 1 January 1997 and 12 September 1998. The texts in the dataset were manually annotated with person, location, and organization entities using the BIO2 scheme in CoNNL format. For the evaluations, we randomly selected 300 sentences from the dataset such that each sentence contains at least one named entity. In total, named entities that refer to 125 person, 150 location, and 110 organization were used during the evaluations.

IV. Experiments and Results

In our experiments, we modeled zero-shot setting by using the 'Instruction Prompt' template and utilized other templates to mimic few-shot setting. Even in the 'Definition Prompt' template, we provided some sort of task-specific information to LLMs. We measured the performance of two well-known language models, namely Google's Gemini 1.5 and OpenAI's ChatGPT 3.5 which have multiple language support including

TABLE I
EVALUATION RESULTS

	Gemini			GPT		
Prompt Template	*Precision*	*Recall*	*F-Measure*	*Precision*	*Recall*	*F-Measure*
Instruction	0.89	0.91	0.90	0.82	0.84	0.83
Definition	0.85	0.92	0.88	0.82	0.88	0.85
Keyword	0.88	0.92	0.90	0.78	0.84	0.81
Definition + Keyword	0.88	0.88	0.88	0.80	0.86	0.83
Contextual	0.87	0.92	0.89	0.80	0.88	0.83
Single Masked	0.90	0.91	0.90	0.80	0.86	0.83
Multiple Masked	0.86	0.91	0.89	0.79	0.83	0.80
Framed	0.87	0.91	0.89	0.82	0.86	0.84
Multiple Masked + Framed	0.85	0.91	0.88	0.84	0.88	0.85
Definition + Multiple Masked	0.87	0.93	0.89	0.83	0.83	0.83
Definition + Framed	0.90	0.90	0.90	0.82	0.88	0.85

TABLE II
SINGLE MASKED PROMPT WITH GEMINI

	Precision	*Recall*	*F-Measure*
Person	0.89	0.90	0.90
Location	0.91	0.95	0.93
Organization	0.88	0.86	0.87

TABLE III
DEFINITION + FRAMED PROMPT WITH GEMINI

	Precision	*Recall*	*F-Measure*
Person	0.88	0.88	0.88
Location	0.92	0.92	0.92
Organization	0.92	0.89	0.90

Turkish. We also experimented with two recently published Llama-based [25] and Mistral-based [26] language models specifically trained for Turkish but observed poor and unstable performances. Using each prompt template, we collected model responses given to our randomly selected sentences as query contexts. The named entities classified by LLMs were compared to manually annotated entities and their matching performances were measured in terms of precision, recall, and f-measure as shown in Table I. The most important observations that we made from the experiments are as follows:

- Both models achieved high f-measure scores between 0.80 and 0.90 where Gemini outperformed ChatGPT in all prompt templates.
- Providing task-related information and examples (few-shot setting) enabled models perform better. Nonetheless, Gemini showed a comparable performance in zero-shot setting as compared to that in few-shot setting.
- Giving the definition of entity classes along with example sentences increased the precision or recall scores.
- Masking only a single named entity rather than masking all entities in a sentence resulted in higher scores in both models.
- Models performed better in recognizing location entities.

We also closely analyzed the performances of these language models in recognizing different entity classes. We observed that performances varied remarkably depending on the prompt being used. For instance, entities that correspond to a person were recognized more effectively than the entities that describe an organization once 'Single Masked Prompt' template is used with Gemini as shown in Table II. However with the 'Definition + Framed Prompt' template, the organization related entities were recognized more accurately by Gemini in comparison to person related entities as shown in Table III.

V. CONCLUSION AND FUTURE WORK

Prompt engineering is a powerful tool that has been successfully applied to various natural language processing tasks. It enables more precise and effective use of large language models once the problem is formulated as a machine reading comprehension task. However, the body of work on prompt engineering in Turkish language processing is still developing with few number of applications so far. In this work, we explored the use of two prompting strategies (zero-shot and few-shot) in recognizing Turkish named entities. For this purpose, we crafted different prompt templates and assessed their performances in recognizing person, location, and organization entities by using two well-know large language models. Although this work provided promising results that highlight the potential of prompt engineering in Turkish, several open questions and areas for future research remain. One key area of interest is enhancing this work to other token-based language processing tasks in Turkish such as aspect-based sentiment analysis. Another important area is the exploration of new datasets (collected from different sources such as Wikipedia), entity types, and language models. Performing similar investigations in other morphologically rich languages is another fruitful research direction.

REFERENCES

[1] S. Sekine and C. Nobata, "Definition, dictionaries and tagger for extended named entity hierarchy", in *The Fourth International Conference On Language Resources And Evaluation (LREC'04)*, 2004.

[2] R. Prasanna Kumar, G. Bharathi Mohan, S. Parthasarathy, and R. Venkatakrishnan, "Transformer-based models for named entity recognition: a comparative study", in *14th International Conference On Computing Communication And Networking Technologies (ICCCNT)*, 2023, pp. 1-5.

[3] G. Aras, D. Makaroğlu, S. Demir, and A. Cakir, "An evaluation of recent neural sequence tagging models in Turkish named entity recognition", in *Expert Systems With Applications*. vol. 182, 2021, pp. 115049.

[4] J. Fu, X. Huang, and P. Liu, “SpanNER: named entity re-/recognition as span prediction”, in *The 59th Annual Meeting Of The Association For Computational Linguistics And The 11th International Joint Conference On Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 7183-7195.

[5] A. Shrimal, A. Jain, K. Mehta, and P. Yenigalla, “NER-MQMRC: formulating named entity recognition as multi question machine reading comprehension”, in *The 2022 Conference Of The North American Chapter Of The Association For Computational Linguistics: Human Language Technologies: Industry Track*, 2022, pp. 230-238.

[6] Y. Fei and X. Xu, “GFMRC: A machine reading comprehension model for named entity recognition”, in *Pattern Recognition Letters*. vol. 172, 2023, pp. 97-105.

[7] Y. Zhang, J. Wang, X. Zhu, T. Sakai, and H. Yamana, “NER-to-MRC: named-entity recognition completely solving as machine reading comprehension”, in *arXiv*, 2023.

[8] X. Du, H. Zhao, D. Xing, Y. Jia, and H. Zan, “MRC-based nested medical NER with co-prediction and adaptive pre-training”, in *The 2024 Joint International Conference On Computational Linguistics, Language Resources And Evaluation (LREC-COLING)*, 2024, pp. 11669-11679.

[9] S. Vatsal and H. Dubey, “A survey of prompt engineering methods in large language models for different NLP tasks”, in *arXiv*, 2024.

[10] F. Villena, L. Miranda, and C. Aracena, “llmNER: (zero—few)-shot named entity recognition, exploiting the power of large language models”, in *arXiv*, 2024.

[11] I. S. Rozhkov and N. V. Loukachevitch, “Prompts in few-shot named entity recognition”, in *Pattern Recognition and Image Analysis*, vol. 33, 2023, pp. 122-131.

[12] G. Tür, D. Hakkahni-Tür, and K. Oflazer, “A statistical information extraction system for Turkish”, in *Natural Language Engineering*. vol. 9, 2023, pp. 181-210.

[13] M. Pehlivanoğlu, M. Syakura, and N. Duru, “Enhancing paraphrasing in chatbots through prompt engineering: a comparative study on ChatGPT”, Bing, and Bard”, in *8th International Conference On Computer Science And Engineering (UBMK)*, 2023, pp. 432-437.

[14] F. Akkurt et al., “Evaluating the quality of a corpus annotation scheme using pretrained language models” ,in *The 2024 Joint International Conference On Computational Linguistics, Language Resources And Evaluation (LREC-COLING)*, 2024, pp. 6504-6514.

[15] L. Cui, Y. Wu, J. Liu, S. Yang, and Y. Zhang, “Template-based named entity recognition using BART”, in *Findings Of The Association For Computational Linguistics: ACL-IJCNLP*, 2021, pp. 1835-1845.

[16] A. Liu et al., “QaNER: prompting question answering models for few-shot named entity recognition”, in *arXiv*, 2022.

[17] Y. Shen et al., “PromptNER: prompt locating and typing for named entity recognition”, in *The 61st Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 12492-12507.

[18] R. Ma et al., “Template-free prompt tuning for few-shot NER”, in *The 2022 Conference Of The North American Chapter Of The Association For Computational Linguistics: Human Language Technologies*, 2022, pp. 5721-5732.

[19] D. Ashok and Z. Lipton, “PromptNER: prompting for named entity recognition”, in *arXiv*, 2023.

[20] H. Zeghidi and L. Moncla, “Evaluating named entity recognition using few-shot prompting with large language models“, in *arXiv*, 2024.

[21] S. Song, F. Shen, and J. Zhao, “RoPDA: robust prompt-based data augmentation for low-resource named entity recognition”, in *The AAAI Conference On Artificial Intelligence*, vol. 38, 2024, pp. 19017-19025.

[22] X. Li et al., “A unified MRC framework for named entity recognition”, in *The 58th Annual Meeting Of The Association For Computational Linguistics*, 2020, pp. 5849-5859.

[23] D. Lee et al., “Good examples make a faster learner: simple demonstration-based learning for low-resource NER”, in *The 60th Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2687-2700.

[24] I. Rozhkov and N. Loukachevitch, “Prompts in few-shot named entity recognition”, in *Pattern Recognition And Image Analysis*, vol.33, 2023, pp. 122-131.

[25] H. Kesgin et al., “Introducing cosmosGPT: monolingual training for Turkish language models, in *arXiv*, 2024.

[26] E. Acikgoz, M. Erdogan, and D. Yuret, “Bridging the bosphorus: advancing Turkish large language models through strategies for low-resource language adaptation and benchmarking”, in *arXiv*, 2024.