Baltic J. Modern Computing, Vol. 13 (2025), No. 2, pp. 357–365
https://doi.org/10.22364/bjmc.2025.13.2.03

Using LLM-s for Zero-Shot NER for Morphologically Rich Less-Resourced Languages

Agris ŠOSTAKS[1], Sergejs RIKAČOVS[1], Artūrs SPROĢIS[1], Oskars MĒTRA[1], Uldis LAVRINOVIČS[2]

[1] Institute of Mathematics and Computer Science, University of Latvia
[2] LETA, Latvian Information Agency

agris.sostaks@lumii.lv, sergejs.rikacovs@lumii.lv, arturs.sprogis@lumii.lv, oskars.metra@gmail.com, uldis.lavrinovics@leta.lv

ORCID 0009-0003-5987-1644, ORCID 0009-0003-8989-0942, ORCID 0000-0002-2320-0887, ORCID 0009-0002-3105-9059, ORCID 0009-0009-8378-2020

Abstract. Developing Named Entity Recognition (NER) solutions for morphologically rich but low-resource languages like Latvian is a complex task. Most state-of-the-art methods rely on deep learning models like BERT, which require substantial expertise in architectures, methods, and access to extensive computational resources and data. In this study, we explore the potential of using popular large language models (LLMs) in a zero-shot setting without additional training. We evaluate their performance on the publicly available Latvian dataset (Gruzitis, et.al., 2018) using the F1-score and find that their results are comparable to state-of-the-art methods. Moreover, LLMs offer a simpler, more resource-efficient alternative for NER tasks.

Keywords: LLM, zero-shot, NER

1 Introduction

Natural Language Processing (NLP) focuses on several key directions, including language understanding, generation, and transformation. Language understanding involves parsing, named entity recognition, and sentiment analysis, which aim to extract meaning and structure from text. Emerging trends include few-shot learning, large pre-trained models like transformers (BERT and LLM-s), and integrating multi-modal data, such as text and images, to enhance understanding and generation.

We focus on the named entity recognition (NER) task, which involves identifying and classifying entities in text into predefined categories, such as names of people, organizations, locations, dates, and more. For example, in the sentence ”Barack Obama

was born in Hawaii," an NER system would identify "Barack Obama" as a person and "Hawaii" as a location.

NER is crucial in several areas of NLP, including information extraction, where it helps convert unstructured text into structured data by identifying key entities like people, organizations, and locations. In search engines, NER improves the relevance of results by recognizing important entities in queries. It also enhances question-answering systems by identifying entities to provide more precise answers. In sentiment analysis, NER associates emotions or opinions with specific entities, such as brands or products. Additionally, NER supports machine translation by ensuring proper handling of named entities across languages, and it plays a role in text summarization by highlighting important entities to generate more informative summaries.

We focus on the automatic recognition of named entities, including people, organizations, and geographical locations. Our work involves extracting information from unstructured, low-quality texts, such as social media posts. Specifically, we analyze data in Latvian, a morphologically rich yet less-resourced language.

At LETA, Latvia's leading news and media monitoring agency, our practical work revolves around sentiment and propaganda analysis, where accurately identifying these named entities is essential. Since LETA has limited computational resources, we explore efficient alternatives to traditional, resource-intensive methods for tackling this task.

With the rise of LLMs, we explore their potential for our task in a zero-shot setting. The accessibility, versatility, and ease of integration of LLMs enable rapid development and incorporation into information extraction systems. Our research focuses on evaluating the quality of outputs from different LLMs in such settings. We designed a single prompt and tested it on several models, including Llama-3.1-405b, GPT-4o-mini, Gemma-2-9b-it, Llama-3.1-8b, and Chat-GPT-4o. These represent popular LLM families, such as open-source Llama models and commercial Chat-GPT systems. We compare models with large and smaller parameter sizes to provide insights into using LLMs both as external services and as locally deployed components. While we have worked with additional models (e.g., Chat-GPT-3.5, Gemini-1.5, LLAMA3-8b-chat), we excluded them due to poor initial results or obsolescence with newer versions.

We evaluated the models on the named entity annotation layer of the publicly available Latvian Multilayer Corpus (FullStack-LV dataset) (Gruzitis et al., 2018) to compare their performance with previous work. The evaluation used the F1-score, which balances precision and recall to measure accuracy. LLMs demonstrated F1-scores close to state-of-the-art, even in zero-shot settings. The best-performing model, Llama-3.1-405b, achieved an F1-score of 81.33, nearly matching the highest known score of 82.6 reported by (Znotiņš and Barzdins, 2020) on the same dataset. Smaller models showed lower performance, with GPT-4o-mini scoring 65.0 and Gemma-2-9b-it scoring 60.2.

The primary contribution of this paper is verifying a seemingly simple yet fundamental question: To what extent can LLMs replace task-specific NER tools in low-resource settings? We evaluate the popular and accessible LLMs for zero-shot Named Entity Recognition (NER) in a morphologically rich, low-resource language, specifically, Latvian. The study focuses on the most common named entity types: person (individual or group names), geopolitical entity and location (representing countries,

cities, regions, and geographical places), and organization (including company and institution names). One might expect pre-trained, fine-tuned models to be significantly superior, yet our findings reveal that the performance gap between the best LLMs and state-of-the-art tools is surprisingly small. This unexpected result highlights an important message for the NLP research community working with less-resourced languages like Latvian. Just as deep learning and transformer models revolutionized NLP, LLMs are now reshaping the field once again—even in low-resource scenarios. Results show that LLMs achieve performance close to state-of-the-art while requiring significantly less effort and resources compared to traditional methods.

2 Related Work — Fast Changing State-of-the-Art

Numerous methods are available for performing NER, with deep learning approaches being the current mainstream. Most commonly used datasets are designed for large languages such as English, Chinese, and Arabic (Hu et al., 2024).

Early deep learning approaches used Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, to capture sequential information from text. Another important method involves Convolutional Neural Networks (CNNs), which are commonly used at the character level to capture morphological features such as prefixes and suffixes, improving entity recognition even in morphologically rich languages. CNNs are often combined with RNNs to improve performance, with CNNs processing local information at the character level and RNNs handling word-level sequences. The introduction of Transformers, particularly models like BERT (Bidirectional Encoder Representations from Transformers), revolutionized NER by enabling the model to capture the bidirectional context in text. Unlike RNNs, which process input sequentially, transformers can access all positions in a sequence simultaneously, allowing them to better understand the context. BERT-based models have achieved state-of-the-art performance on various NER tasks by fine-tuning them on labelled NER datasets. Moreover, Conditional Random Fields (CRFs) are often used on top of RNN or BERT architectures to model the dependencies between output labels, ensuring that the sequence labelling respects entity boundaries. However, building such models demands considerable effort and resources. LLMs have transformed NER by utilizing deep contextual understanding, enabling fine-tuning for specific tasks, supporting few-shot learning through prompting, and transferring knowledge across languages and domains. This makes LLMs powerful and versatile tools for modern NER systems.

Fine-tuning LLMs for NER adapts a pre-trained model to accurately recognize and classify entities within a specific dataset. This process begins with a base LLM pre-trained on large-scale textual data and fine-tunes it using a labeled NER dataset, where each token or word is annotated with its entity type. For example, SLIMER (Zamai et al., 2024) is a fine-tuned Llama-2-7b model designed specifically for NER tasks.

Few-shot learning for NER with LLMs leverages their ability to generalize from minimal labeled data, avoiding the need for extensive task-specific fine-tuning. This approach involves providing the LLM with a small set of labeled examples during inference, illustrating how text tokens correspond to specific entity types. The model uses its pre-trained contextual knowledge and these examples to identify and classify entities in

unseen text. Prompt engineering plays a key role, with carefully designed prompts guiding the model's performance. Notable examples include methods to optimize prompt structure and example selection (Cheng et al., 2024), GPT-NER for labeling entities using few-shot learning (Wang et al., 2023), and PromptNER, which integrates entity definitions and examples directly in the prompt (Ashok et al., 2023). Few-shot learning is especially valuable in domain-specific or low-resource settings, as it reduces the need for extensive annotated datasets and computational power. Techniques like in-context learning (Jiang et al., 2024) and retrieval-augmented generation further enhance performance by improving the model's understanding of the NER task.

LLMs' ability to capture complex word dependencies and contextual relationships suggests they could perform well in zero-shot settings without specific pre-training or additional examples. Promising results in this area (Xie et al., 2023) have been achieved using techniques like syntactic prompting combined with tool augmentation. While such approaches have shown success for major languages, we aim to evaluate their effectiveness in zero-shot settings for less-resourced languages, specifically Latvian.

The first attempt to address NER for Latvian was made with the TildeNER toolkit (Pinnis, 2012), which uses a supervised conditional random field classifier enhanced with heuristic and statistical refinement methods. TildeNER achieved an F1-score of approximately 60 on a manually created dataset containing 881 named entities. The authors focused on three NER entity types: locations, persons, and organizations.

Vīksna and Skadina (2020) introduced a pre-trained BERT model trained on large Latvian corpora, achieving an F1-score of 81.91 across 9 NER types. Meanwhile, Znotins and Barzdins (2020) developed LVBERT, a BERT-based model fine-tuned specifically for Latvian to enhance performance on Latvian NLP tasks. LVBERT reached a state-of-the-art F1-score of 82.6 on the FullStack-LV dataset.

Next, Vīksna and Skadina (2022) investigated the performance of various multilingual NER models within the state-of-the-art natural language processing framework, Flair. They found that for Latvian, the more specialized LitLat BERT model achieved the best F1-score of 81.97 on the FullStack-LV dataset. Therefore, BERT-based fine-tuned models currently deliver the best results for morphologically rich, less-resourced languages like Latvian. However, creating such models is a complex and resource-intensive process.

3 Prompt Engineering

We use LLMs to address NER tasks. While LLMs can be fine-tuned for specific tasks using deep learning methods, this requires significant resources, including large datasets. As an alternative, zero-shot prompt-based methods are used to guide the model without the need for extensive fine-tuning. These prompts direct LLMs to perform specific tasks by providing clear, structured instructions within the input. Instead of retraining the model, prompts leverage its existing knowledge by specifying the task, input format, and output requirements.

For our task, we create prompts that instruct the LLM to extract mentions of different named entity types from Latvian text. We design a separate prompt for each entity type, as adjusting a single prompt for all entity types is challenging due to differing

errors. Since the structure of the prompts is similar for each type, we illustrate the case of named entities for persons. Here is a step-by-step breakdown of the prompt:

Task definition: LLM needs to act as an NLP expert and apply NER techniques to identify all mentions of individuals (persons) in the provided text. This reduces ambiguity and ensures the model works within the intended scope of NER.

```
1 Act as a NLP researcher performing Named Entity Recognition (
    NER).
2 Analyze the following text fragment labeled TextToAnalyze.
3 From that fragment, extract a list with named entity mentions
     that represent persons (named individuals).
```

Clarification of the task: LLM has to exclude generic terms. The prompt emphasizes that only specific individuals should be listed. Generic roles or titles like ”teacher,” ”president,” or ”doctor” should be excluded. This ensures that the list only includes names of actual people and not their job positions or generic designations. This aligns the model’s attention with the task and minimizes false positives.

```
1 Ensure that named entities representing persons refer to
    specific individuals by excluding generic terms such as
    titles or roles.
2 If a named entity refers to a role or position, exclude it
    from the list of people.
3 Before giving the answer analyze the list you created and
    exclude from this list items that are not referring to
    named individuals.
```

Output definition: the extracted person entities must be returned in two forms: a) the original form as it appears in the provided text; b) the name of the person converted to the nominative case (which is the default grammatical case for the subject in Latvian, like ”John” instead of ”John’s”). This ensures consistency in the representation of named entities, even if they appear in different grammatical forms in the text. It enhances the usability of the output by providing the proper ”base” form of names, which is crucial for downstream tasks like database matching or reference alignment. The final output must be a JSON object with the key *persons*, storing an array of the extracted named person entities. If no person entities are found, the JSON should return an empty list. The instruction explicitly states, ”Do not give any additional explanation,” forcing the LLM to stick to the task at hand and focus on generating the output in the desired format without unnecessary verbosity or commentary, improving the response’s efficiency and clarity.

```
1 Return JSON object. This object should have field persons,
    containing extracted person mentions.
2 For each item in this list provide both latvian text labeled
    as lv, and same text but in nominative case labeled lv_nc
    .
3 If there are no entity mentions to return - return empty list
    .
4 Do not give any additional explanation.
5 Ensure that you are returning valid JSON.
```

Input:

```
TextToAnalyze:
"Gleznas attēlo , kā Jānis Bērziņš paraksta laulību līgumu ar
     Annu Kalniņu."
```

Output Example:

```
{
  "persons": [
    {
      "lv": "Jānis Bērziņš",
      "lv_nc": "Jānis Bērziņš"
    },
    {
      "lv": "Annu Kalniņu",
      "lv_nc": "Anna Kalniņa"
    }
  ]
}
```

4 Experiment

We use the FullStack-LV dataset (Gruzitis et al., 2018) to evaluate LLMs. This Latvian corpus is designed for broad applications, including natural language understanding (NLU), abstractive text summarization, and knowledge base population. It features hierarchical named entity annotations with both outer and inner (nested) entities. The dataset includes 3947 paragraphs of text, containing 9697 outer entities and 944 inner entities, categorized into nine types: geopolitical entities (GPE), person, time, location, product, organization, money, event, and a general ”entity” category. It adopts a simplified CoNLL-2003 format with BIO (Begin, Inside, Outside) labeling.

The FullStack-LV dataset is commonly used to train and evaluate NER models, including multilingual transformers and other machine learning approaches. Experiments with multilingual transformers have shown strong results, though F1-scores vary depending on the model and training parameters. The dataset's hierarchical structure presents an additional challenge, particularly for models not optimized for nested entity recognition.

We focus on three named entity categories: persons, locations and GPEs, and organizations. The subset of the dataset used for evaluation includes 3,104 person entities, 2,031 GPEs and locations, and 1,847 organization entities, making up 72% of all outer entities in the dataset. This subset is sufficient for evaluation since results vary similarly across entity types, as shown in previous research (Pinnis, 2012), (Vīksna, 2020), and in practical applications, prompts would need to be tailored for each type separately.

It is important to note that our evaluation differs from previous research on Latvian NER. Traditionally, tokenization is performed first; for example, LVBERT uses LVTagger (Paikens et al., 2013) for sentence tokenization. In such cases, the model classifies tokens directly, labeling each token individually. This simplifies F1-score calculation,

as results are straightforward to interpret by comparing gold labels with classified labels to identify true and false positives.

In our approach, the process is more complex. Extracted entities are compared to gold-standard entities and their labels. The gold data includes two forms: the exact string from the text and its nominative (base) form. Models are asked to extract both forms, and we perform cross-comparison. If either extracted form matches the gold data, it is counted as a true positive; otherwise, it is a false positive. Additionally, false negatives—entities missed by the models—must be accounted for.

This method introduces room for errors. For instance, quotation marks in organization names can cause mismatches when models omit them. Simple data cleansing, such as removing quotation marks and trimming leading or trailing whitespace from the gold data, significantly improves results.

We evaluated five models: Llama-3.1-405b, chat-gpt-4o, gpt-4o-mini, gemma-2-9b-it, and Llama-3.1-8b. Access to these models was provided via online services using their respective APIs. The aggregate F1-scores for all evaluated NER types on the FullStack-LV dataset (Gruzitis et.al., 2018) are presented in Table 1. The first row presents the baseline result achieved by LVBERT (Znotins and Barzdins, 2020). The second result, shown in parentheses (81.3), was obtained after applying cleansing procedures to the gold data.

Table 1. The results of LLM tests for the NER task.

Model	F1-Score
LVBERT (Baseline)	82.6
Llama_3.1_405b	76.6 (81.3)*
gpt_4o	71.9
gpt_4o_mini	65.0
gemma-2-9b-it	60.2
Llama_3.1_8b	16.6

Larger models, such as Llama-3.1-405b and chat-gpt-4o, achieve performance close to state-of-the-art, while smaller models like gpt-4o-mini and gemma-2-9b-it perform worse, with Llama-3.1-8b showing even significantly lower results.

It is important to note that LLMs exhibit considerable variation in performance across different NER types. For instance, Llama-3.1-405b achieves a high F1-score of 91.0 for person entities but only 50.0 for organizations, which improves to 69.0 after applying data cleansing. Similar disparities are observed with other LLMs, aligning with discrepancies noted in previous research (Pinnis, 2012), (Vīksna, 2020).

5 Conclusion

LLMs represent a transformative technology for NER tasks, even for morphologically rich and less-resourced languages like Latvian. Despite not being specifically trained

for Latvian—e.g., Llama 3 contains only 5% non-English data—their ability to process texts in Latvian with near state-of-the-art quality is remarkable. What truly sets LLMs apart is their out-of-the-box usability, eliminating the need for pre-training or fine-tuning. This greatly simplifies and broadens the application of NER technology for less-resourced languages.

However, challenges remain in using LLMs for private data that cannot be processed via third-party servers or transmitted over the Internet. Smaller LLMs, which demand fewer computational resources, currently lack the required quality for effective NER tasks. Conversely, deploying larger LLMs requires significant investment unless sufficient computational infrastructure is available.

Looking ahead, we anticipate ongoing advancements in LLM quality for NER as models continue to evolve. Techniques like prompt engineering could further improve the extraction of specific NER types, while fine-tuning and few-shot learning are likely to enhance overall performance even further. We expect LLMs to soon surpass state-of-the-art NER methods. However, a more in-depth performance and error analysis is necessary, as F1-scores vary significantly across different NER types. This analysis will provide a clearer understanding of the limitations and best-use scenarios for LLMs in NER tasks.

Acknowledgements

The research leading to these results has received funding from the research project "Competence Centre of Information and Communication Technologies" of the EU Structural funds, contract No.5.1.1.2.i.0/1/22/A/CFLA/008 signed between IT Competence Centre and Central Finance and Contracting Agency, Research No. 2.6 "Applications of Large Language Models in Analyzing Large Text Corpora".

References

Ashok, D., and Lipton Z. (2023) "PromptNER: Prompting for Named Entity Recognition." arXiv preprint arXiv:2305.15444.

Cheng, Q., Liqiong, C., Zhixing, H., Juan, T., Qiang, X., Binbin, N. (2024) "A Novel Prompting Method for Few-Shot NER via LLMs." Natural Language Processing Journal, Volume 8, 2024, 100099. ISSN 2949-7191. https://doi.org/10.1016/j.nlp.2024.100099.

Gruzitis, N., Pretkalnina, L., Saulite, B., Rituma, L., Nespore-Berzkalne, G., Znotins, A., Paikens, P. (2018) "Creation of a Balanced State-of-the-Art Multilayer Corpus for NLU." Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC).

Hu, Z., Hou, W., Liu, X. (2024) "Deep Learning for Named Entity Recognition: A Survey." Neural Comput & Applic 36, 8995–9022. https://doi.org/10.1007/s00521-024-09646-6.

Guochao, J., Zepeng, D., Yuchen, S., Deqing, Y. (2024) "P-ICL: Point In-Context Learning for Named Entity Recognition with Large Language Models." arXiv preprint arXiv:2405.04960.

Paikens, P., Rituma, L., Pretkalnina, L. (2013) "Morphological Analysis with Limited Resources: Latvian Example." In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), pp. 267–277.

Pinnis, M. (2012) "Latvian and Lithuanian Named Entity Recognition with TildeNER." Seed 40: 37.

Vīksna, R., and Skadiņa, I. (2020) "Large Language Models for Latvian Named Entity Recognition." Human Language Technologies–The Baltic Perspective. IOS Press, pp. 62-69.

Vīksna, R., and Skadiņa, I. (2022) "Multilingual Transformers for Named Entity Recognition." Baltic Journal of Modern Computing, Volume 10, Issue 3.

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Wang, G. (2023) "GPT-NER: Named Entity Recognition via Large Language Models." arXiv preprint arXiv:2304.10428.

Xie, T., Li, Q., Zhang, J., Zhang, Y., Liu, Z., Wang, H. (2023) "Empirical Study of Zero-Shot NER with ChatGPT." In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 7935–7956.

Zamai, A., Zugarini, A., Rigutini, L., Ernandes, M., Maggini, M. (2024) "Show Less, Instruct More: Enriching Prompts with Definitions and Guidelines for Zero-Shot NER." arXiv preprint arXiv:2407.01272.

Znotiņš, A., and Barzdiņš, G. (2020) "LVBERT: Transformer-Based Model for Latvian Language Understanding." Human Language Technologies–The Baltic Perspective. IOS Press, pp. 111-115.

Received January 13, 2025 , revised March 11, 2025, accepted April 7, 2025