Baltic J. Modern Computing, Vol. 13 (2025), No. 3, pp. 551–566
https://doi.org/10.22364/bjmc.2025.13.3.01

Improving NER State-of-the-Art for Latvian with LLM and Few-Shot Learning

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 complex. Most state-of-the-art methods rely on deep learning models like BERT (Bidirectional Encoder Representations from Transformers), which require substantial expertise in architectures, methods, and access to extensive computational resources and data. This study shows how to use popular large language models (LLMs) in a few-shot setting, without fine-tuning to surpass state-of-the-art results. We evaluate their performance on the publicly available Latvian dataset (Gruzitis, et al., 2018). We analyze the performance of LLMs on the recognition of different NER types.

Keywords: LLM, few-shot, NER, Latvian

1 Introduction

Natural Language Processing (NLP) encompasses several fundamental areas, including language understanding, generation, and transformation. Language understanding involves tasks such as parsing, named entity recognition (NER), and sentiment analysis, all of which aim to extract meaning and structure from text. Recent advancements in the field include few-shot learning, and large-scale pre-trained models like transformers (e.g., BERT and LLMs). **We research the options to use the LLMs for the NER task for morphologically rich low-resourced languages, in particular, Latvian.**

LLMs have been already used for NLP tasks in Latvian. The performance of LLMs varies between nearly perfect for some tasks and unsatisfactory for others. (Dargis et al.,

2024) investigates the application of LLMs to natural language understanding (NLU) multiple-choice questions in Latvian, with GPT-4o achieving an accuracy of 82%, surpassing average human performance. However, the study also highlighted notable deficiencies in natural language generation (NLG) tasks across all models, underscoring persistent challenges in generating coherent and contextually appropriate text analyses in low-resource languages like Latvian. Therefore, while the researchers acknowledged the advancements in NLU performance, they emphasized the need for further LLM improvements in NLG capabilities for these languages. (Kostiuk et al, 2025a) evaluates LLMs for question-answering (QA) tasks in Latvian, reporting varying outcomes, with GPT-4o attaining the highest accuracy at 89%. The same authors (Kostiuk et al, 2025b) assess LLMs on Wikipedia-based QA in Latvian, demonstrating near-perfect results, indicating the models' proficiency in this context. (Purvins et al., 2024) explores the use of LLMs for sentiment analysis in Latvian, introducing a novel dataset derived from Reddit data. By employing prompt engineering with the GPT-3.5-turbo model, the study achieved 82% accuracy, significantly surpassing previous results on the Latvian Tweet Sentiment Corpus. Thus, the use of LLMs seems appropriate for various NLP tasks in Latvian.

Our focus is on NER, a task that involves detecting and categorizing entities in text into predefined classes, such as names of individuals, organizations, locations, and dates. For instance, 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 plays a vital role in multiple NLP applications. For information extraction, it helps structure unstructured text by identifying key entities such as people, organizations, and locations. Search engines leverage NER to improve result relevance by recognizing significant entities in queries. Similarly, in question-answering systems, NER enhances response accuracy by pinpointing relevant entities. For sentiment analysis, NER associates opinions or emotions with specific entities, such as brands or products. Additionally, NER contributes to machine translation by ensuring proper handling of named entities across languages and supports text summarization by emphasizing key entities to generate more informative summaries.

Our research focuses on automatically identifying named entities, including individuals, organizations, and geographical locations, particularly in unstructured, low-quality text, such as social media posts. Specifically, we work with Latvian, a morphologically complex yet under-resourced language. At LETA, Latvia's leading news and media monitoring agency, our practical work involves sentiment and propaganda analysis, where accurate identification of named entities is crucial. Given LETA's limited computational resources, we explore efficient alternatives to traditional, resource-intensive approaches to address this challenge effectively.

We saw that LLMs perform close to the state-of-the-art (Šostaks, et al.,2025). We experimented with a zero-shot setting where Llama-3.1-405b, achieved an F1-score of 0.8133, nearly matching the highest known score of 0.826 reported by (Znotiņš and Barzdins, 2020) on the same FullStack-LV dataset (Gruzitis, et al., 2018). The setting of the experiment was close enough to conclude the potential of LLMs, but now we show that LLMs (OpenAI o3-mini in particular) exceed the previous state-of-the-art by reaching 0.84 F1-score. We did it using the few-shot prompting method using a

single prompt on the cleaned subset of the FullStack-LV dataset. We asked the LLM to annotate the given text using CoNLL-2003 NER type annotations and the IOB2 tagging scheme. Thus, LLM performs both - tokenization and NER.

The **contribution of this paper** is the following:

- we show how LLMs with few-shot learning might be used to achieve the state-of-the-art NER for Latvian, morphologically rich but low-resourced language;
- we highlight and provide qualitative analysis of problems with NER datasets in Latvian and why it is the reason our and previous results are not entirely reliable;
- we show that different NER types are recognized with different accuracy, and discuss the possible reasons;
- we sketch the possible use of LLMs to improve the quality of NER datasets.

The paper is organized as follows. Section 2 describes the state-of-the-art of NER for Latvian. Section 3 contains a description of prompt engineering used in the experiment, and Section 4 states the settings, steps, and results of the experiment. This section also contains an error analysis of the gold data, an analysis of the typical mistakes made by LLMs, and remarks on the experiments with LLMs in zero-shot setting. The paper has conclusions and an appendix where the full prompt can be seen.

2 State-of-the-Art of NER for Latvian

Numerous methods are available for performing NER, with deep learning approaches being the current mainstream. The most commonly used datasets are designed for large languages such as English, Chinese, and Arabic (Hu et al., 2024). Currently, the trending large language models (LLMs) are showing remarkable results for various NLP tasks. 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 have been achieved using techniques like syntactic prompting combined with tool augmentation (Xie et al., 2023) and few-shot learning (Wang et al., 2023). While these approaches have shown success for major languages, we aim to evaluate their effectiveness for less-resourced languages, specifically Latvian. Much less explored is the performance of LLMs for morphologically rich less-resourced languages, such as Latvian with 1.5 million native speakers. Our latest findings suggested that LLMs in zero-shot settings performed almost as well as state-of-the-art methods (Znotiņš and Barzdins, 2020) for 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 F-score of approximately 0.60 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 0.8191 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 0.826 on the FullStack-LV dataset (Gruzitis, et al., 2018). 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 F-score of 0.8197 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.

Let's discuss how the state-of-the-art results (Znotins and Barzdins, 2020) have been achieved and what was the experiment setting. After the monolingual transformer model was trained on a 500-million-token Latvian corpus, the researchers used a FullStack-LV dataset containing named entities such as persons, organizations, locations, and events, ensuring a balanced train-test split to prevent document overlap. For a full list of NER types in the dataset see Table 1. The input text was tokenized using a custom

Table 1. Named Entity Types in the FullStack-LV Dataset with Latvian and English Examples

Entity Type	Description	Example (Latvian / English)
PER (Person)	Names of individuals, both real and fictional	Jānis Bērziņš / John Smith
ORG (Organization)	Companies, institutions, government agencies, and organized groups	Latvijas Universitāte / NASA
GPE (Geopolitical Entity)	Political and administrative regions, such as countries and cities	Latvija / European Union
LOC (Location)	Geographical places that are not geopolitical entities, such as mountains and rivers	Baltijas jūra / Carpathian Mountains
PROD (Product)	Names of products, services, and other artefacts	Aldaris Gaišais / Windows 11
EVE (Event)	Names of significant occurrences, including wars, festivals, and sports events	Pasaules kauss / Olympic Games 2024
MON (Money)	Monetary values, salaries, prices, and financial figures	100 eiro / $5 million
TIM (Time)	Specific dates, time intervals, and event-relevant timing expressions	2023. gada 10. novembris / three years ago

model, generating a 32,000-token vocabulary optimized for Latvian, reducing fragmentation issues seen in Multilingual BERT (mBERT). The Bidirectional LSTM with a CRF layer was applied for sequence labelling, utilizing the IOB2 tagging scheme to identify multi-token entities. The IOB2 tagging scheme is a standard method for labelling named entities in sequence labelling tasks like NER. It assigns one of three tags to each token: B- (Beginning) for the first token of an entity, I- (Inside) for subsequent tokens within the same entity, and O (Outside) for tokens that do not belong to any entity. This approach ensures that multi-word entities are correctly identified while maintaining clear boundary distinctions. For example, in the sentence "Barack Obama visited New York", the tags would be: Barack (B-PER), Obama (I-PER), visited (O), New (B-LOC), York (I-LOC).

F1-score was the primary evaluation metric, and LVBERT achieved the highest score (0.826), outperforming all other models. The researchers primarily reported overall NER performance using the F1-score, but they did not provide a detailed breakdown of performance for different named entity types (e.g., Person, Organization, Location, etc.).

3 Prompt Engineering

In this section, we describe the prompt engineering used for our experiment. The full prompt is given in the Appendix.

This prompt is a carefully structured instruction set designed to guide an LLM in performing NER using the IOB2 tagging scheme. It employs multiple prompt engineering techniques, including explicit task definition, structured guidelines, few-shot learning, output formatting constraints, and step-by-step processing. These techniques ensure high accuracy, consistency, and adaptability in the model's responses. The input for the prompt is the text to be analyzed, the output is JSON containing all the tokens (words, punctuation signs, etc.) annotated using CoNLL labels and IOB2 schema.

The prompt starts with a clear task definition, explicitly stating that the model will receive a block of text and must analyze it according to the CoNLL dataset labelling rules. By specifying the dataset standard, the prompt ensures that the model follows established NER conventions. Furthermore, the instruction that the output must be formatted in JSON enforces structured, machine-readable results, making the response directly useful for downstream applications. This structured approach helps prevent hallucinations and ensures that the model adheres to predefined annotation rules.

A critical part of this prompt is the detailed labelling guidelines, which specify the IOB2 scheme. To remind, the B- (Beginning) tag is assigned to the first word in a named entity, the I- (Inside) tag is given to subsequent words within the same entity, and the O (Outside) tag is used for words that do not belong to any named entity. This format ensures that multi-word entities are correctly segmented. For example, the name "Einšteina kungs" is labelled as B-PER and I-PER, correctly capturing its entity boundaries. Without this explicit segmentation, a model might misinterpret whether words belong to separate or single entities.

The prompt further strengthens entity classification by defining eight distinct entity types: PER (Person), ORG (Organization), GPE (Geopolitical Entity), LOC (Location), TIM (Time), PRO (Product), EVE (Event), and MON (Money). Each category is accompanied by precise definitions and special cases to prevent misclassification. For instance, "Latvijas dzelzceļš" (Latvian Railways) must be tagged as B-ORG, I-ORG, ensuring that multi-word organizations are properly labelled. Similarly, "London city" is classified as B-LOC, I-LOC, differentiating it from geopolitical entities such as "Latvia," which falls under B-GPE. These fine-grained distinctions prevent ambiguous interpretations and improve classification accuracy.

Another powerful technique in the prompt is example-driven few-shot learning, where multiple entity annotation cases are explicitly provided. These include numeric entities like event numbers ("15. olimpiskās spēles" → B-EVE, I-EVE, I-EVE), quoted names (""Apple"" → B-PRO, I-PRO, I-PRO), and temporal expressions ("since last

year" → B-TIM, I-TIM, I-TIM). By exposing the model to various cases, it learns the correct labelling without requiring additional context. This transforms the prompt into a few-shot learning approach, as the model generalizes from the provided examples.

To ensure output consistency, the prompt enforces structured JSON formatting. The response must be a list of objects where each word is mapped to its corresponding label. The example for "NATO contract" illustrates this well, ensuring that the model generates an output like:

```
[
    { "word": "NATO", "label": "B-PRO" },
    { "word": "contract", "label": "I-PRO" }
]
```

By explicitly defining this format, the prompt minimizes hallucinations, response inconsistencies, and incorrect structural outputs, making the results easily machine-parsable.

Finally, the instruction "Respond step by step." serves as a mechanism to enhance logical processing and reasoning accuracy. Instead of generating all outputs at once, the model is encouraged to break down the task, improving its ability to process entities sequentially and avoid tagging errors. This incremental reasoning technique, often used in chain-of-thought prompting, helps the model resolve complex entity boundaries and ambiguous cases.

Overall, this prompt is an example of how structured instruction, example-based learning, and enforced output formatting can be leveraged to optimize LLM performance in NER tasks. By combining clear guidelines, diverse labelled cases, JSON constraints, and step-by-step reasoning, it ensures high accuracy, consistency, and adaptability.

4 Experiment

The goal of the experiment was to match or even exceed the previous state-of-the-art (baseline) result, LVBERT, (Znotins and Barzdins, 2020) by using out-of-the-box LLMs and few-shot learning. To achieve this goal we tried to make an experiment as close to the baseline settings as possible. Although the NER task is formulated the same way - LLM should annotate the tokens with the same set of annotations, we have included the tokenization in the task (LVBERT used a custom tokenizer). It has been done indirectly without explicitly stating the tokenization task in the prompt. In early stages of prompt development, we observed that the LLMs produced inconsistent tokenization, which led to annotation errors and instability in results. However, these issues were resolved as we iteratively refined the prompt, adding more structured examples and clearer instructions. We hypothesize that this improvement is due to the LLM's emergent ability to focus more accurately on the intended task when given richer context and guidance, which aligns with findings on prompt sensitivity in large language models (Zhou et al., 2023).

After the first experiments of annotation task we conducted with several LLMs (GPT-4o, Llama-3.1-405, Llama-3.3-70, DeepSeek-V3), we noticed that the quality of the FullStack-LV dataset's NER-annotated layer was poor. We tested LLMs on the small data subsets and manually examined errors made by them. It turned out that there were cases, when mistakes were made not by LLMs, but by human annotators.

Further analysis of errors in the FullStack-LV dataset's NER layer gold data reveals several notable inconsistencies and misclassifications. One of the primary issues is attention-related errors, where certain entities are inconsistently tagged. For instance, in a sentence mentioning multiple individuals—Jānis, Pēteris, and Aivars—only Jānis and Pēteris are annotated as entities, while Aivars is omitted, suggesting a lack of annotation consistency. Another significant inconsistency is observed in the labelling of identical terms starting with capital letters across similar contexts. For example, the word Birojs (Office) is annotated as an entity in one sentence but left untagged in another structurally similar sentence. This suggests annotation subjectivity, where the same term is interpreted differently depending on the annotator or the context in which it appears. Such errors likely arise due to multiple annotators working on the dataset, leading to annotation discrepancies. Additionally, there are cases of conceptual misinterpretation, particularly concerning geopolitical and organizational entities. A notable example is Eiropas Savienība (European Union), which, despite being an organization, is sometimes incorrectly classified as a geographical location rather than an entity in the Organization (ORG) category. Another issue involves time-related entities, which frequently go untagged. This omission may stem from the cognitive difficulty of simultaneously identifying multiple entity types during annotation. It is plausible that when annotators focus on recognizing persons and organizations, they inadvertently overlook temporal expressions, leading to systematic gaps in annotation. This phenomenon raises questions about cognitive load in multi-entity tagging, which could benefit from further investigation in cognitive linguistics. Entities such as persons and organizations are generally easier to identify, likely due to the orthographic feature of capitalization, which provides a strong visual cue. However, the lack of a formalized definition for each entity type may contribute to inconsistencies. If entity definitions exist within the dataset guidelines, they are not readily accessible, making it difficult for annotators to apply consistent classification criteria.

We sampled 2200 sentences out of 13691 sentences in the FullStack-LV dataset taking into account different sources the dataset consists of (the dataset is a balanced 10-million-word text corpus: 60% news sources, 20% fiction, 10% legal texts, 5% spoken, 5% miscellaneous) achieving as full coverage as possible. This amount is comparable to the test set volume for LVBERT. First, we corrected the errors in the gold data. The process begins with an LLM labelling named entities in the text, followed by a systematic manual review of false positives (incorrect entity classifications) and false negatives (missed entities). We corrected these errors, ensuring contextual correctness, consistency in entity tagging, and proper boundary detection. However, the final quality depends on several factors, including the accuracy of the initial LLM-generated labels, the expertise of human annotators, and the consistency of annotation guidelines. We think, that the quality of the data can be improved even further, taking into account the resources and time we were able to invest in this task. The amount of work needed to correct the data was the main reason to evaluate the subset and not the full dataset. Next, we evaluated the OpenAI o3-mini and GPT-4o models using OpenAI API on the selected sentences. We have chosen these models because they showed the best results in preliminary experiments. The F1-score over all NER types was 0.84 and 0.79 accordingly (see Table 2).

Table 2. The results of LLM tests for the NER task are based on the FullStack-LV dataset (Gruzitis et al., 2018). The first row presents the baseline result achieved by LVBERT (Znotins and Barzdins, 2020).

Model	F1-Score
LVBERT (Baseline)	82.6
OpenAI o3-mini	84
GPT-4o	79

While it may be tempting to describe our few-shot prompting approach as achieving a new state-of-the-art result for Latvian NER, such a claim would be misleading due to inconsistencies in dataset quality across prior work. Our experiment uses an improved and corrected version of the dataset, whereas earlier models—including LVBERT—were trained on the original version, which contains annotation errors and inconsistencies. As such, direct comparisons are not entirely fair. Nonetheless, LVBERT remains a useful baseline, as it was trained on the same underlying data and represents the strongest previously reported result using it.

The presence of noisy or inconsistently annotated data (”dirty” data) in the dataset likely imposed an upper bound on achievable model performance. Models trained on such data are not only limited in their ability to learn correct entity boundaries and labels but may also internalize incorrect patterns or biases introduced by annotation errors. As a result, their evaluation metrics, such as the F1 score, can give a misleading impression of real world capability. Use of a corrected and higher-quality dataset offers the potential to train models that better capture the true structure of the task. We hypothesize that if models like LVBERT or newer transformer-based systems were retrained on this improved dataset, they could surpass previously reported results, as the cleaner data provides a more learnable signal and less noise, enabling more accurate learning and evaluation.

We want to point out the problems with drawing strict conclusions from our (and all previous) results. There are two main issues: 1) the results differ significantly across different NER types (see Table 3 and Table 4). We use resampling-based method (Efron and Tibshirani, 1993) for calculation of confidence intervals (CI); 2) the quality of the Latvian NER dataset must be improved to assess the performance of different models.

4.1 Cross-Type Performance Analysis

Looking at Person (PER) entities, both models demonstrate exceptional performance, with high precision and recall, resulting in F1-scores above 0.90. This indicates that the models are highly effective in identifying personal names, whether real or fictional. The ability to consistently recognize named individuals with minimal false positives or false negatives suggests that person entity recognition is a well-learned task, likely due to distinct linguistic patterns and frequent occurrences in training data.

For Location (LOC) entities, both models perform well, though with a slightly lower recall compared to precision. This suggests that while they accurately classify locations,

Table 3. Class-wise and Overall NER Metrics for OpenAI o3-mini

Class	Precision	Recall	TP	FP	FN	F1-Score [95% CI]
PER	0.92	0.92	447	37	38	0.92 [0.906, 0.940]
LOC	0.90	0.86	321	37	51	0.88 [0.853, 0.903]
ORG	0.78	0.86	233	64	39	0.82 [0.785, 0.849]
TIM	0.87	0.75	244	38	82	0.80 [0.768, 0.837]
PRO	0.70	0.62	88	37	54	0.66 [0.591, 0.726]
MON	0.41	0.90	9	13	1	0.56 [0.357, 0.722]
EVE	0.58	0.52	26	19	24	0.55 [0.414, 0.660]
Overall	0.85	0.83	-	-	-	0.84

Table 4. Class-wise and Overall NER Metrics for GPT-4o

Class	Precision	Recall	TP	FP	FN	F1-Score [95% CI]
PER	0.94	0.96	409	28	15	0.95 [0.934, 0.964]
LOC	0.78	0.87	251	69	36	0.83 [0.791, 0.861]
ORG	0.70	0.80	182	77	46	0.75 [0.699, 0.787]
TIM	0.68	0.65	150	71	81	0.66 [0.611, 0.712]
PRO	0.56	0.57	70	55	52	0.57 [0.487, 0.633]
MON	0.31	1.00	8	18	0	0.47 [0.267, 0.667]
EVE	0.49	0.46	22	23	26	0.47 [0.329, 0.594]
Overall	0.76	0.81	-	-	-	0.79

they may miss some valid location mentions. The challenge here likely arises from ambiguity between locations and geopolitical entities (GPE), as well as the presence of less common location names that may not have been sufficiently represented during training.

In Organization (ORG) recognition, both models exhibit good but not perfect performance, with a moderate balance between precision and recall. A notable trend is that organizations are sometimes over-identified, leading to false positives. This could stem from difficulty in distinguishing between organization names and other proper nouns, especially in cases where multi-word entities or abbreviations are involved. The tendency to err on the side of classification rather than omission suggests that organization detection is challenging due to the diverse structures of organizational names.

When handling Time (TIM) entities, the models show greater inconsistency, with recall being lower than precision. This indicates that while they are able to recognize time expressions with high accuracy, they often fail to detect all relevant instances. The difficulty likely comes from the wide range of temporal expressions, including relative time references like ”five months ago” or ”yesterday”, which require a deeper understanding of context rather than just lexical recognition.

Product (PRO) and Event (EVE) entity recognition presents a notable challenge, as both categories have lower precision and recall compared to other entity types. The models often misclassify general nouns as products or events, leading to high false positive rates. Additionally, products and events are highly domain-specific, meaning that their representation in training data may be insufficient for robust generalization. The tendency to over-predict in these categories suggests that entity boundaries are

harder to establish, especially when dealing with complex multi-word product names or event titles.

Money (MON) entities show the most imbalanced performance, with recall being much higher than precision. This means that the models are effective in capturing all monetary values but struggle with differentiating between actual monetary mentions and unrelated numerical values. The high false positive rate suggests that non-monetary numerical expressions are frequently misclassified, which could be a problem in financial or legal applications where precision is critical.

4.2 Typical Errors Made by LLMs

Lastly, let's go through the qualitative examples of common mistakes made by LLMs. One issue is the failure to recognize personal names with uncommon or short forms, such as Sī (Xi), particularly when they appear at the beginning of a sentence. In these cases, the model struggles to distinguish whether the capitalization is due to sentence-initial position or an actual named entity, leading to misclassification or omission. This suggests that LLMs rely heavily on capitalization cues without deeper syntactic or contextual understanding, which is particularly problematic in a morphologically rich language like Latvian.

Another notable challenge is the confusion between literary works and personal names. For instance, in the phrase "Jānis Pērā Gintā labi spēlēja" (John acted well in Peer Gynt), the model may misinterpret Pērs Gints (a play title) as part of the personal name, leading to incorrect tagging. This type of error indicates that LLMs may lack a strong internalized knowledge base of Latvian cultural references and named entities, resulting in an inability to correctly differentiate between works of art and individuals. Similar misclassifications occur in other domains, such as music groups, films, and concerts, where an entity can belong to multiple categories depending on context. For example, "Only one" (a song) can be misclassified as a product, an organization, or an event depending on sentence structure, requiring a more advanced context-aware disambiguation strategy.

A frequently observed error involves incorrect classification of geopolitical entities (GPE) and organizations. In cases such as "Latvijas Republikas Valsts ieņēmumu dienests" (Republic of Latvia, State Revenue Service), the model often misclassifies "Latvijas Republika" as a location (GPE) while recognizing only "Valsts ieņēmumu dienests" as an organization. This error suggests that the model fails to correctly interpret hierarchical organizational structures, instead splitting multi-word entities into separate, independent labels. Improving multi-token entity recognition and ensuring that models can process complex administrative names as unified entities would help address this issue.

LLMs also exhibit systematic misclassification of numerical values as monetary amounts, particularly when isolated numbers appear in a sentence. This suggests an overgeneralization of patterns commonly associated with financial data, leading to false positives in monetary entity detection. Similarly, temporal expressions are inconsistently recognized. While the model correctly tags complex time-related phrases in some cases, it inexplicably fails to recognize basic time entities such as "vakar" (yesterday)

in certain contexts. This inconsistency indicates that the model may struggle with implicit temporal cues or prioritize more explicitly structured date expressions over simple adverbial time references.

Another category of errors arises from incorrect entity classification based on context. An example is Dailes teātris (Dailes Theatre), which can be classified as either a location or an organization depending on usage. In ”Dailes teātrī dod garšīgu kafiju”, it should be a location, while in ”Dailes teātris pieņēma darbā aktieri”, it functions as an organization. Similar issues are observed with brand names like ”Audi”, which can refer to a product, a company, or even a location (e.g., a car interior in a figurative sense). These errors highlight the model's difficulty in applying entity type flexibility based on sentence-level context, suggesting a need for improved semantic reasoning in ambiguous cases.

Finally, LLMs struggle with long, complex named entities and acronyms. While shorter and more well-known acronyms are often recognized, uncommon or multi-word place names are frequently misclassified or truncated. For instance, ”Aizkraukles novada Daugavas labās pietekas aizsargjoslas krastā” (on the shore of the protective zone of the right tributary of the Daugava River in Aizkraukle Municipality) presents significant difficulty, with the model failing to correctly tag the entire phrase as a location entity. This issue suggests that current LLMs may have limitations in handling extended named entities that do not fit into familiar patterns. Addressing this would require better long-span entity detection and improved handling of nested or hierarchically structured locations.

Overall, these findings emphasize the need for improved entity disambiguation, better handling of context-dependent classifications, and enhanced recognition of complex multi-word named entities in Latvian. Incorporating context-aware entity resolution, improved hierarchical NER strategies, and more robust linguistic knowledge about cultural and administrative terms would be essential to improve LLM performance in Latvian NER tasks.

4.3 Do Few-Shots Matter?

We also conducted experiments in zero-shot prompting for the NER task. In the zero-shot setting, no examples were provided in the prompt (see the Appendix, lines 27-80). We found that models struggled to produce correctly structured outputs without examples: approximately 2% of responses (each response conforms to a single sentence) were malformed, typically due to incorrect JSON formatting. When these malformed outputs were excluded from the evaluation, the performance of the remaining valid outputs was still consistently lower than that of few-shot prompting. For instance, the o3-mini model achieved only 0.72 F1-score, while o4 performed even worse, with an F1-score of 0.64 in the zero-shot configuration.

We think that the superior performance of few-shot prompting in our experiments arises from its ability to better ground the model in task expectations through concrete examples. When LLMs are given a few illustrative input-output pairs, they more effectively infer both the semantic labeling requirements and the expected output structure, in this case, JSON-formatted NER annotations. In contrast, zero-shot prompts often leave room for ambiguity, especially for tasks involving structured outputs or multiple

constraints. This lack of guidance likely contributes to the significantly higher rate of malformed outputs observed in the zero-shot setting. By seeing examples, the model can better align its generation behavior with the desired schema, reducing both format errors and label inconsistencies. This aligns with previous findings that LLMs benefit from inductive bias provided by demonstrations, particularly in tasks that require compositional reasoning or adherence to strict formatting (Brown et al., 2020).

5 Conclusion

LLMs leveraging few-shot learning techniques **might achieve state-of-the-art performance for NER in Latvian**, surpassing previous benchmarks set by LVBERT. The experiment shows that OpenAI o3-mini attained an F1-score of 0.84 exceeding the prior best score of 0.826 on the cleaned version of the same dataset. We emphasize the potential of LLMs in handling morphologically rich and low-resourced languages like Latvian, reducing the need for resource-intensive fine-tuned transformer models.

Our experiment and also the research of (Dargis et al., 2024) **highlight significant challenges in developing high-quality language models for Latvian**, primarily due to the lack of well-annotated and comprehensive datasets. The poor quality of existing Latvian datasets, as observed in our analysis, further exacerbates this issue, leading to inconsistencies in NER and broader NLP tasks. Without reliable benchmarking datasets, it becomes difficult to accurately evaluate model performance, refine annotations, and draw significant conclusions on the generalization and practical implications of the results. To advance Latvian NLP research, there is a critical need for higher-quality, standardized datasets that account for linguistic nuances, reduce annotation errors, and provide a more representative foundation for training and evaluation.

Our **experiment hints that LLMs can be used to improve the quality of NER datasets**. For example, multiple LLMs can be leveraged to both annotate NER datasets and automatically identify and resolve errors within them. By utilizing multiple models with varying architectures, training data, and biases, it is possible to compare their outputs and detect inconsistencies in entity labelling. A practical approach involves running different LLMs independently on the same dataset, allowing them to recognize named entities and observing where they agree and where their classifications diverge. Areas of high agreement among models suggest greater confidence in correct entity classification, while areas of disagreement indicate potential annotation errors, ambiguities, or edge cases requiring closer inspection.

This comparative approach allows for a semi-automated error detection pipeline, where LLM consensus can be used to refine NER datasets. In cases where models consistently misclassify certain entities, it may highlight systematic biases in model training data or deficiencies in the dataset itself, such as missing entity definitions or ambiguous annotation guidelines. By identifying these discrepancies, annotators can prioritize manual verification for problematic cases rather than reviewing the entire dataset.

Furthermore, an ensemble of LLMs could be used to propose corrections for misclassified entities. If multiple models label an entity differently, a confidence-based resolution strategy can be implemented, where entities with low inter-model agreement are flagged for human review. Over time, this iterative process can improve dataset quality

by reducing annotation errors, improving consistency, and refining named entity type definitions. By integrating LLMs as annotation assistants rather than relying solely on human annotators, NER dataset development can become more efficient, scalable, and less prone to human annotation errors, ultimately leading to more robust and generalizable models in downstream NLP applications. Similar ideas, leveraging LLMs for NER annotations, have been already approbated, e.g., (Naraki et al., 2024) or (Bogdanov et al., 2024).

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.

Bogdanov, S., Constantin, A., Bernard, T., Crabbé, B., Bernard, E. (2024). NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 11829-11841).

Brown, T., et al., (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Darģis, R., Barzdins, G., Skadiņa, I., Saulīte, B. (2024). Evaluating Open-Source LLMs in Low-Resource Languages: Insights from Latvian High School Exams. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities* (pp. 289-293).

Efron, B., Tibshirani, R.J. (1993). An Introduction to the Bootstrap. *Monographs on statistics and applied probability*, 57(1), 1-436.

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.

Kostiuk, Y., Vitman, O., Gagała, Ł., Kiulian, A. (2025a). Towards Multilingual LLM Evaluation for Baltic and Nordic languages: A study on Lithuanian History. *arXiv e-prints*, arXiv-2501.

Kostiuk, Y., Vitman, O., Gagała, Ł., Kiulian, A. (2025b). The Veln(ia)s is in the Details: Evaluating LLM Judgment on Latvian and Lithuanian Short Answer Matching. *arXiv e-prints*, arXiv-2501.

Naraki, Y., Yamaki, R., Ikeda, Y., Horie, T., Naganuma, H. (2024). Augmenting NER Datasets with LLMs: Towards Automated and Refined Annotation. *arXiv e-prints*, arXiv-2404.

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

Purvins, P., Urtans, E., Caune, V. (2024) Using large language models to improve sentiment analysis in Latvian language. *Baltic Journal of Modern Computing*, *12*(2), 165-175.

Šostaks, A., Rikačovs S., Sproģis A., Mētra O., Lavrinovičs U. (2025) Using LLM-s for Zero-Shot NER for Morphologically Rich Less-Resourced Languages. Baltic Journal of Modern Computing Vol. 13, No. 2, pp. 357–365

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.

Zhou, C., et al., (2023). Lima: Less is more for alignment. Advances in Neural Information Processing Systems, 36, 55006-55021.

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

6 Appendix - Prompt

```
You will be provided with a block of text, labeled as **Text**.
 Your task is to analyze the text and label each word according to the CoNLL dataset labeling rules.
 Then, output the labeled words in JSON format.

 Labeling Guidelines:
 1. Each word in the text should be labeled with one of the following tags:
 - B- (Beginning): Marks the beginning of a named entity.
 - I- (Inside): Marks a word inside the same named entity.
 - O (Outside): Marks words that are not part of any named entity.

 2. The named entities can belong to the following types:
 - PER: Person, make sure to extract only named people including fictional like gods, but not positions, job titles, pronouns, references, etc.
 - ORG: Organization, including named companies, institutions, state organizations, units, military organizations, divisions, etc. and named complex multi name organizations.
 - GPE: Geopolitical Entity, like administrative territories including states, countries, cities, towns, villages, etc. Make sure the EU, the Sovient Union, the Commonwealth of Independent States, Eastern Europe are also GPE.
 - LOC: Location, including non-GPE locations, mountain ranges, named rivers, named lakes, in buildings, full
```

```
addresses, continents, regions like Scandinavia, Baltics,
 etc., but not countries, cities.
- TIM: Time, like dates, time intervals, specific timing
relevant to an event, etc. Some examples are "Now", "
Yesterday", "five month ago", "10 days later", "since
last year", "before Christmas", etc.
- PRO: Things like monuments, laws, statements, documents
, pacts, portals, diseases, medicine, chemicals, rock
bands, albums, movies, plays, services and other
artifacts, including and named complex multi named
entities.
- EVE: Event, like named hurricanes, battles, wars,
sports, events, elections, etc. and named complex multi
name events.
- MON: Money, like monetary values, salaries, prices, etc
.

Expected JSON Output:
The output should be a JSON object where:
- there is a key named result, representing list of
objects with keys : word (original word) and
corresponding label of that word.

The JSON object should be a list of these word objects.

Example:
For the text: "Einšteina kungs ir fiziķis". Note that the
 person contains multiple words.
The output should be:
[
{{ "word": "Einšteina", "label": "B-PER" }},
{{ "word": "kungs", "label": "I-PER" }},
{{ "word": "ir", "label": "O" }},
{{ "word": "fiziķis", "label": "O" }},
]

If the entity contains multiple words and some of them
belong to location or GPE
or ORG, then assign them to the entity's class.

For example, "Latvijas dzelzceļš" has to be tagged as:
[
{{ "word": "Latvijas", "label": "B-ORG" }},
{{ "word": "dzelzceļš", "label": "I-ORG" }}
]

or "15. olimpiskās spēles"

[
{{ "word": "15.", "label": "B-EVE" }},
```

```
    {{ "word": "olimpiskās", "label": "I-EVE" }},
    {{ "word": "spēles", "label": "I-EVE" }}
    ]

    or "NATO contract"

    [
    {{ "word": "NATO", "label": "B-PROD" }},
    {{ "word": "contract", "label": "I-PROD" }},
    ]

    If there is a location, for example, "London city", then tag both words like:
    [
    {{ "word": "London", "label": "B-LOC" }},
    {{ "word": "city", "label": "I-LOC" }}
    ]

    It there is a quotes or other symbols as part of the entity, for example, "Apple",
    then include the symbols in the name as well. For example,
    [
    {{ "word": """, "label": "B-PRO" }},
    {{ "word": "Apple", "label": "I-PRO" }},
    {{ "word": """, "label": "I-PRO" }}
    ]

    Timing has to contain all the words. For example, "since last year":
    [
    {{ "word": "since", "label": "B-TIM" }},
    {{ "word": "last", "label": "I-TIM" }},
    {{ "word": "year", "label": "I-TIM" }}
    ]


    Make sure to correctly distinguish between locations and GPE.

    Text: '''{text}'''

    Respond step by step.
    Return the response as a JSON object.
```

Received March 24, 2025 , revised June 26, 2025, accepted July 17, 2025