FEW-SHOT EVENT EXTRACTION IN LITHUANIAN WITH GOOGLE GEMINI AND OPENAI GPT

Arūnas Čiukšys and Rita Butkienė

Department of Information Systems, Kaunas University of Technology, Kaunas, Lithuania

ABSTRACT

Automatic event extraction (EE) is a crucial tool across various domains, allowing for more efficient analysis and decision-making by extracting domain-specific information from vast amounts of textual data. In the context of under-resourced languages like Lithuanian, the development of EE systems is particularly challenging due to the lack of annotated datasets. This study investigates and evaluates the event extraction capabilities of two large language models (LLMs): OpenAI's GPT and Google Gemini, using few-shot prompting. We propose novel methodologies, including a combined approach and a layered prompting approach, to improve the performance of these models in identifying two specific event types. The models were benchmarked using various performance metrics, such as accuracy, precision, recall, and F1-score, against a manually annotated gold-standard corpus. The results demonstrate that LLMs achieve satisfactory performance in extracting events in Lithuanian, though model accuracy varied depending on the prompting methodology. The findings underscore the potential of LLMs in addressing event extraction challenges for under-resourced languages, while also pointing to opportunities for improvement through enhanced prompt strategies and refined methodologies.

KEYWORDS

Event Extraction, LLMs, Few-Shot Prompting, Gemini, GPT, Layered Prompting, Combined Prompting

1. INTRODUCTION

Automatic event extraction (EE) is a necessary tool across various domains [1] [2][3][4][5]enabling more efficient analysis and decision-making used for extracting domain-specific information. In the contemporary digital era, an overwhelming quantity of textual data is generated daily. Automatic EE systems can quickly analyze this data to identify relevant events, thereby saving time and resources compared to manual processing. These automated systems can even operate in real time [6], which is particularly advantageous for applications necessitating immediate insights, such as financial markets, news monitoring, and social media analysis. By providing timely and precise information, automated event extraction empowers decision-makers to respond swiftly and effectively to emerging trends, potential threats, and opportunities. This technology allows organizations to reallocate human resources from repetitive data extraction and analysis tasks to more strategic activities, thereby enhancing overall productivity.

David C. Wyld et al. (Eds): CSITAI, MATHCS, ITCAU, COMSAP, ACINT - 2024
pp. 173-184, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142513

1.1. Automatic Event Extraction for Under-Resourced Languages

Automatic EE systems earlier had been crafted using pipeline [7]or pattern-matching methodology[8] and lately were outperformed by machine learning (ML) based models[9][10][11]for this task. The main problem when creating these types of models for under-resourced languages such as Lithuanian[12], is that labeled datasets do not exist. For various NLP tasks[13], including automatic event extraction, large language models (LLMs) come into play due to their capabilities to efficiently process large volumes of text and extract the desired information [14] [15] [2] [16][17] in zero-shot or few-shot prompting [1][2][18].

The Lithuanian language presents unique challenges for NLP due to its rich morphology and relatively limited digital linguistic resources. These factors complicate the development of robust NLP systems, including those for event extraction. Despite these challenges, advancements in language technology and the availability of pre-trained language models like OpenAI's GPT (GPT) or Google Gemini (GEMINI) offer new opportunities to enhance NLP capabilities for the Lithuanian language. The advent of large language models [16], particularly GPT [19] and Gemini[20],with the ability to work without additional model fine-tuning or training, has generated significant interest in their applicability to specific natural language processing tasks.

1.2. Event Types

There is only one gold-standard corpus created for Lithuanian[21], in which two event types, based on the MUC [22] framework, have been annotated. These event types were therefore selected for this research.

Contact.Meet. This event refers to situations where two or more individuals come together physically at a specific location. The primary purpose is face-to-face communication or interaction. Such events are usually planned, like meetings, appointments, or encounters.

Contact.Phone– Write. This event describes a non-face-to-face communication where individuals interact through written or spoken forms. This encompasses phone calls as well as written communications, including emails, text messages, and letters.

1.3. Objective

The objective of this study is to investigate and assess the event extraction capabilities of GPT and GEMINI for the Lithuanian language. Specifically, we aim to analyze their performance using various few-shot prompting methodologies, including the introduction of a novel approach. We will assess and compare the performance of the latest GPT and GEMINI models in extracting events from Lithuanian texts.

The remainder of the paper is structured as follows: Section 2 provides an overview of the related work. Section 3 presents the methodology for event extraction using LLMs in Lithuanian, including two novel approaches: a combination of both models with different rules for integration, and a layered approach. Section 4 covers the experiments and discusses the results. Finally, the conclusions are presented, along with directions for future work.

2. BACKGROUND AND RELATED WORK

Event extraction is a critical subfield of natural language processing that has evolved significantly with advancements in machine learning and large language models.

2.1. Events

The definition of an event varies depending on the specific domain or task. For this research, we adopt the definition from the Automatic Content Extraction (ACE) 2005 evaluation, which defines an event as "*a specific occurrence involving participants.*"In this context, an event is represented by a sentence that details the event, and an event trigger is a word or phrase that best expresses the occurrence of an event.

For example, consider the sentence: "*During the meeting in Paris between U.S. President Biden and the President of France, a breakthrough agreement on climate action was signed by several key nations.*"Here, the event type is CONTACT, with the subtype MEET. The entire sentence functions as an event mention, and the word "meeting" serves as the event trigger.

2.2. Event Extraction

Event Extraction[23] is a subfield of NLP that involves the identification and extraction of specific events from textual data. This task includes different elements such as event triggers, participants, and the temporal and spatial context. Traditional event extraction techniques have relied heavily on machine learning and pattern-matching systems, which require extensive annotated data and domain-specific knowledge. However, LLMs can be harnessed zero-shot and play an important role in small languages. The primary objective of event extraction is to transform unstructured text into structured information by identifying events and their components, adhering to guidelines such as those outlined in ACE 2005. In this study, we classify sentences based on the events they represent and pinpoint the event trigger attributes.

2.3. Large Language Models

Large language models have showcased exceptional performance across a wide range of NLP tasks[14] [15] [2] [16][17][1][2][18].

These models leverage vast amounts of data and computational power to generate human-like text and understand complex linguistic patterns. Furthermore, they can be prompted in many different languages including small ones.LLMs have been effectively applied to a variety of tasks, including text generation, machine translation, information retrieval, and event extraction.

2.4. Google Gemini and OpenAI GPT

In this research, we utilized two of the most recent models from Google and two from OpenAI, totaling 4 models (Table 1).

Google's Gemini 1.5 Pro represents the forefront of multimodal AI models, leveraging the Mixture of Experts (MoE) architecture to seamlessly process and integrate multiple data modalities, including text, images, and audio, simultaneously. This architecture ensures efficient resource allocation by selectively engaging sub-networks (experts) depending on the task, making it particularly well-suited for complex, cross-modal tasks like event extraction and data analysis. In contrast, Google Gemini Flash 1.5 is a more streamlined version designed for real-

time applications, offering lower latency while maintaining the ability to handle multimodal inputs, albeit with reduced depth in processing.

On the other hand, OpenAI GPT-4o is a text-based model that uses transformer architecture and self-attention mechanisms to perform complex natural language processing tasks. The model excels in high-accuracy tasks such as text generation, reasoning, and language-based event extraction. Meanwhile, the GPT-4o mini version is a streamlined, agile variant optimized for real-time, low-latency applications, providing quick responses while maintaining a high level of accuracy. While Google Gemini excels in multimodal applications, GPT-4o models are optimized for text-based tasks, providing specialized performance in their respective domains.

Table 1: LLMs comparison

LLM	Description	Architecture / Modality
Google Gemini [24]1.5 Pro	The most capable model from Google for complex tasks.	Multimodal model + Mixture of Experts (MoE) [25]architecture. This architecture allows it to process multiple data types simultaneously.
Google Gemini Flash 1.5	Lightweight high-performance version of Gemini Pro. For real-time lower-latency tasks.	
OpenAI GPT-4o[26]	The most capable model from OpenAI for complex tasks.	Transformer model [27] (self-attention mechanism) with text-based modality.
OpenAI GPT-4o mini	Lightweight high-performance version of GPT-4o. For real-time lower-latency tasks.	

2.5. Prompting Techniques

In the domain of NLP, prompting techniques (Table 2) play a critical role in optimizing the performance of LLMs for various tasks. Zero-shot prompting [18]is a technique where the model is prompted to perform a task without being provided with any task-specific examples.The model relies solely on its pre-trained general knowledge to generate responses. This approach is particularly useful in scenarios where labeled data is scarce or unavailable. However, due to the lack of context, zero-shot prompting can sometimes yield lower accuracy, especially in tasks requiring nuanced understanding or complex reasoning. In contrast, few-shot prompting [18]provides the model with a limited number of task-specific examples, helping it better grasp the context and produce more accurate responses.This technique is particularly effective in cases where providing a few relevant examples helps the model generalize to new tasks. Few-shot prompting has gained widespread adoption for its ability to improve model performance with minimal training data, making it especially useful for tasks such as event extraction and summarization.

Chain-of-thought prompting[28]is another advanced technique that guides the model to explicitly articulate its reasoning process before reaching a final answer. This approach encourages the model to break down complex problems into manageable steps, enhancing performance in tasks requiring multi-step reasoning, such as problem-solving and question-answering. Self-consistency prompting [29]builds upon this by generating multiple reasoning paths and selecting the most consistent outcome as the final answer. These techniques have been shown to improve accuracy in complex reasoning tasks by encouraging the model to self-assess its reasoning.

Lastly, instruction-based prompting [30]provides explicit instructions to guide the model's task execution, ensuring clarity and structure in the model's responses. This approach is highly effective in tasks where precise instructions are needed, such as text generation or code completion. Meta-prompting[31], on the other hand, focuses on iteratively refining the quality of prompts. By learning from previous interactions, meta-prompting enhances the model's ability to

generate high-quality outputs over time, making it a dynamic and adaptive approach to prompt generation.

Table 2: Prompting Techniques Comparison

Prompting Technique	Description
Zero-Shot Prompting[18]	No task-specific examples are provided. The model is prompted to perform the task based on general understanding.
Few-Shot Prompting[18]	A few task-specific examples are provided to guide the model's responses.
Chain-of-Thought Prompting[28]	The model is prompted to articulate its reasoning step by step before providing a final answer.
Self-Consistency Prompting[29]	The model generates multiple reasoning paths and selects the most consistent one as the final answer.
Instruction-Based Prompting[30]	The model is provided with explicit instructions on how to complete the task.
Meta-Prompting[31]	Learning to generate better prompts by iteratively improving the quality of prompts given to the model.

3. METHODOLOGY OF USING LLM FOR AUTOMATIC EVENT EXTRACTION

Our EventGPT application, initially designed for gold corpus creation and synthetic data generation in Lithuanian, has been expanded to support the objectives of this research. We implemented automatic event extraction in Lithuanian using few-shot prompting methodologies, leveraging both Google Gemini and OpenAI GPT via API. Additionally, we integrated features for the automatic evaluation of each approach, enabling us to easily obtain results. These enhancements utilize a pre-built framework and incorporate a novel technique.

3.1. Framework

The framework designed for event extraction consists of four key components:

Task Description: In this section, the language model is tasked with identifying one of two specific event types, if applicable, and generating a structured, labeled output.

Definitions: Instead of providing exact definitions, a reference is made to the conference where these two event types were described, with the expectation that the LLM will already know this information.
Example: A well-crafted example is given, demonstrating how the structured output should be formatted, serving as guidance for the model.

Task Sentence: This part includes the sentence that needs to be labeled as either an event or not an event.

According to this framework, prompts are automatically generated for each sentence, following the structure outlined in the framework. As illustrated in Table 3, the prompt consists of a task description, relevant references, a sample structured output, and the specific sentence to be classified as an event or non-event.

Table 3: Event Extraction Prompt Example

This task involves identifying one of two event types: Contact.Meeting or Contact.Written-Phone and determining their time. The task text may present only one event. If the text contains one of the required events, provide a structured output following the example. Otherwise, respond with: "unidentified." The definitions of the event and its types are taken from Linguistic Data Consortium (2005). ACE (Automatic Content Extraction) English Annotation Guidelines for Events. Example: Input: Bush and Putin met this week to discuss matters regarding Chechnya. Output:[{"Trigger":"met","Time":"this week","Type":"Contact.Meeting"}] Task text: On Thursday, President Gitanas Nausėda, addressing the leaders of the European Political Community (EPC), emphasized the need to actively defend democratic values and the territorial integrity of countries.

3.2. Combined Approach

A novel LLM combination, referred to as the Combined Approach (CA) for EE, is introduced. In this method, both models are prompted for event extraction on each sentence. Upon receiving outputs from both models, two scenarios can occur: Combined Approach OR (CA OR) and Combined Approach AND (CA AND).

In CA OR, an event is considered identified if either of the models detects it.

In CA AND, the event is only recognized if both models agree on the event type and structure. This combined approach allows leveraging the strengths of multiple models to handle the complexity of event extraction tasks.

3.3. Layered Prompting Approach

A novel layered prompting approach (LPA) for event extraction (Fig. 1) is proposed, where an initial prompt is followed by two subsequent layers applying stressed prompting [32].

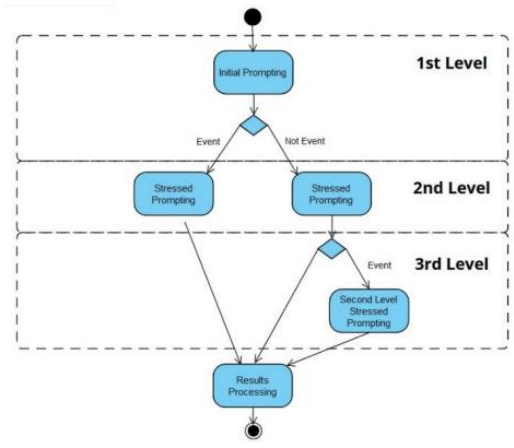

Figure 1: Layered Prompting Approach

First Layer – Initial Prompting (IP): In this stage, the language model is asked to provide an output based on the previous described framework and example in Table 3. The model then classifies the sentence as either an event or not an event.

Second Layer – Stressed Prompting (SP): If the LLM does not classify the sentence as an event in the first layer, the SP provides explicit definitions of the event types and prompts the model to reassess whether the sentence is not an event. An exampleis provided in Table 4. If the LLM classifies the sentence as an event, specific definitions and key points related to the classified event type are highlighted, prompting the model to confirm the accuracy of its decision.An example is in Table 5.

Third Layer – Second Stressed Prompting (SSP): If the sentence was initially classified as not an event but, after the SP, the model changes its decision, a second round of stressed prompting is applied. The model is asked once more to confirm its decision, with a strong emphasis on the key points and definitions related to the event type.

This layered approach ensures a thorough evaluation of each sentence, allowing the model to reconsider its decisions with more detailed information in each subsequent layer.

Table 4: Stress Prompting after Positive EE

You have determined that this sentence: (After Israel announced that its foreign minister met with his counterpart from Libya, despite the fact that the two countries do not maintain official diplomatic relations, protests erupted in Libya, according to Libyan media.) matches the Contact.Meeting type event according to the definition provided below. Please double-check to confirm whether it is clearly stated or can be reasonably inferred that the meeting is indeed happening physically. If so, provide a structured output according to the example. Otherwise, respond with: "unidentified." Event and type definitions: Contact.Meeting: A meeting of 2 or more entities (must be mentioned, e.g., people, countries, companies) at a specific physical location (face-to-face). Includes talks, summits, conferences, visits, and other meetings. For a sentence to match the Contact.Meeting event, it must clearly describe that the meeting is happening physically, i.e., people are meeting face-to-face somewhere. For example, "GM talks to Chrysler about acquiring Jeep" is not considered a meeting event. Contact.Written-Phone: 2 or more people (must be mentioned) contact each other by phone or in writing (e.g., by email) remotely without a physical meeting. "A person told reporters" or "issued a press release" is not this event. Example: Input: Bush and Putin met this week to discuss matters regarding Chechnya. Output:[{"Trigger":"met","Time":"this week","Type":"Contact.Meet"}]

Table 5: Stress Prompting after Negative EE

You have determined that this sentence: (Turkey's leader Recep Tayyip Erdoğan announces that the parliament will ratify Sweden's NATO membership only when the U.S. allows Ankara to sell F-16 fighter jets.) does not match any of the events listed below. Please double-check to confirm whether it indeed does not match any of the definitions provided below. If it does, provide a structured output according to the example. Otherwise, respond with: "unidentified." Event and type definitions: 1. Contact.Meeting: A meeting of 2 or more entities (must be mentioned, e.g., people, countries, companies) at a specific physical location (face-to-face). Includes talks, summits, conferences, visits, and other meetings. For a sentence to match the Contact.Meeting event, it must clearly describe that the meeting is happening physically, i.e., people are meeting face-to-face somewhere. For example, "GM talks to Chrysler about acquiring Jeep" is not considered a meeting event. 2. Contact.Written-Phone: 2 or more people (must be mentioned) contact each other by phone or in writing (e.g., by email) remotely without a physical meeting. "A person told reporters" or "issued a press release" is not this event. Example: Input: Bush and Putin met this week to discuss matters regarding Chechnya. Output:[{"Trigger":"met","Time":"this week","Type":"Contact.Meeting"}]

4. EXPERIMENT

To evaluate the effectiveness of large language models in event extraction tasks for Lithuanian, a series of experiments were conducted using a predefined framework. The focus was on comparing the performance of various prompting methodologies and models to identify their strengths and limitations in this context.

4.1. Experimentation and Evaluation

The experiment benchmarked the performance of the latest language models from Google and OpenAI available at the time of the study. Utilizing the pre-designed framework and the EventGPT application, for automatic prompting and annotation of events according to the output was conducted on the dataset of which we have a gold corpus using different models and approaches. Following the event extraction process, the performance of each model was evaluated against a manually annotated golden corpus to measure its effectiveness.

The evaluation metrics employed in the study include:
Accuracy: The overall correctness of the event classification.
Precision: The proportion of true positive event classifications among all positive classifications made by the model.
Recall: The ability of the model to identify all relevant events within the dataset.
F1-Score: The harmonic mean of precision and recall, providing a balanced measure of a model's performance in event extraction.

These metrics offered a comprehensive assessment of the models' ability to extract and classify events compared to the human-annotated dataset. Benchmarking results are provided in Table 6.

Table 6: LLM Results Comparison

LLM, Methodology	Accuracy	Precision	Recall	F1-Score
GPT-mini	92,92%	74,76%	30,2%	43,02%
GPT-mini LPA	92,61%	63,64%	41,18%	50%
GPT-4o	93,17%	63,32%	56,86%	59,92%
GPT-4o LPA	92.82%	62.29%	49.41%	55.26%
Gemini 1.5 Flash	85,6%	35,79%	76,08%	48,68%
Gemini 1.5 Flash LPA	83,07%	31,72%	76,86%	44,91%
Gemini Pro 1.5	92,29%	57,69%	52,94%	55,21%
Gemini Pro 1.5LPA	92,64%	60%	54,12%	56,91%
CA (GPT-4o + Gemini Pro 1.5) And	93,03%	67,7%	42,75%	52,41%
CA (GPT-4o + Gemini Pro 1.5) OR	92,43%	56,62%	67,06%	61,4%

4.2. Discussion

The accuracy results across all models are generally high, indicating that they perform well in event extraction tasks. Gemini Flash, however, shows slightly lower accuracy compared to the other models. This indicates that while the models are proficient in accurately classifying events, Gemini Flash may face challenges with specific aspects of the task.

When looking at Precision, we see more variability. Precision, which measures the model's ability to avoid false positives, reveals that some models are prone to predicting events that do not exist. Gemini Flash performs notably poorly in this metric, indicating a high rate of false positives. Even after applying the Layered Prompting Approach, Gemini Flash's precision decreases further. Interestingly, GPT-mini achieves the highest precision, implying it is more conservative in labeling events, which may reduce false positives but might miss some actual events.

Recall, which evaluates the model's capability to identify all true events, varies across models. Gemini Flash demonstrates strong recall, indicating its ability to detect a high number of events. However, this strength is offset by reduced precision, as it frequently misclassifies non-events as events. This high recall, paired with low precision, suggests that Gemini Flash is overly aggressive in event identification. The LPA increases recall further but reduces precision, which seems to result in more balanced outcomes. Gemini Pro with LPA shows improvements in both precision and recall, which is an impressive feat, highlighting its ability to better balance event detection and accuracy.

Lastly, the F1-score, which harmonizes precision and recall, serves as a comprehensive metric for comparing model performance. The Combined Approach using GPT-4o and Gemini Pro with the OR condition achieves the best F1-score, meaning that this combined model strikes the best balance between precision and recall. On the other hand, both lightweight models, GPT-4o mini and Gemini Flash, demonstrate the lowest F1-scores, showing that they struggle to balance event identification with avoiding false positives. This suggests that while these models may perform

well on certain tasks, they are less effective for this specific event extraction task compared to their more advanced counterparts.

5. CONCLUSION

The results of our experiment demonstrate that language models achieved satisfactory performance in extracting events in Lithuanian. However, the quality of these models varied based on the prompting methodology employed, highlighting the need for experimentation with different approaches. To further enhance performance, incorporating additional layers of prompting and employing more sophisticated prompt frameworks and methodologies could prove beneficial.

These findings underscore the potential of LLMs, such as OpenAI GPT and Google Gemini, for tackling event extraction tasks in under-resourced languages like Lithuanian. This research suggests that while current models perform well, there is room for improvement through refining prompt strategies and enhancing methodological frameworks, which may result in more accurate and reliable event extraction outcomes.

5.1. Limitations

Despite the promising results in this study, one important limitation needs to be acknowledged. The availability of annotated datasets for Lithuanian is very limited. As the research only utilized two event types from the MUC framework for evaluation, this limits the generalizability of the findings to other event types or domains. Furthermore, although few-shot prompting approaches demonstrated satisfactory performance, their effectiveness is highly dependent on the complexity and clarity of the task. They may still lag behind fully supervised models fine-tuned on extensive datasets.

5.2. Future Work

To overcome these limitations, future work could explore several directions. Firstly, expanding the range of event types and datasets used for training and evaluation would enable a more thorough analysis of the models' generalization capabilities. Creating more gold-standard corpora for would be a significant step forward.

Finally, future research could also experiment with more advanced prompting techniques, such as dynamic or adaptive prompting, where the model refines its prompts based on its own performance and feedback. Additionally, integrating more layers of prompting or multi-step reasoning techniques like chain-of-thought prompting could further enhance the performance, particularly for more complex event types or scenarios where the context is less straightforward.

REFERENCES

[1] C. Lou, J. Gao, C. Yu, W. Wang, H. Zhao, W. Tu and R. Xu, "Translation-Based Implicit Annotation Projection for Zero-Shot Cross-Lingual Event Argument Extraction," in SIGIR '22: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.

[2] Z. Sun, G. Pergola, B. Wallace and Y. He, "Leveraging ChatGPT in Pharmacovigilance Event Extraction: An Empirical Study," in Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), 2024.

[3] T. Wu, F. Shiri, J. Kang, G. Qi, G. Haffari and Y.-F. Li, "KC-GEE: knowledge-based conditioning for generative event extraction," World Wide Web (Springer), vol. 26, p. 3983–3999, 2023.

[4] C. Saetia, A. Thonglong, A. Thonglong, T. Amornchaiteera, T. Amornchaiteera, T. Chalothorn, T. Chalothorn, S. Taerungruang, S. Taerungruang, P. Buabthong and P. Buabthong, "Streamlining event extraction with a simplified annotation framework," Frontiers Artificial Intelligence, vol. 7, 2024.

[5] W. Liu, L. Zhou, D. Zeng, Y. Xiao, S. Cheng, C. Zhang, G. Lee, M. Zhang and W. Chen, "Beyond Single-Event Extraction: Towards Efficient Document-Level Multi-Event Argument Extraction," in Findings of the Association for Computational Linguistics ACL 2024, Bangkok, 2024.

[6] C. Jenkins, S. Agarwal, J. Barry, S. Fincke and E. Boschee, "Massively Multi-Lingual Event Understanding: Extraction, Visualization, and Search," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), Toronto, 2023.

[7] D. Ahn, "The stages of event extraction," in Proceedings of the Workshop on Annotating and Reasoning about Time and Events, 2006.

[8] E. Riloff, "Automatically Generating Extraction Patterns from Untagged Text," in Proceedings of the National Conference on Artificial Intelligence, 1996.

[9] Y. Chen, L. S. and J. Zhao, "Joint event extraction via recurrent neural networks," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017.

[10] J. Liu, Y. Chen, K. Liu, W. Bi and X. Liu, "Event Extraction as Machine Reading Comprehension," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

[11] Y. Zhang, G. Xu, Y. Wang, X. Liang, L. Wang and T. Huang, "Empower event detection with bi-directional neural language model," ELSEVIER Knowledge-Based Systems, vol. 167, pp. 87-97, 2019.

[12] G. A. Melnik-Leroy, J. Bernatavičienė, G. Korvel, G. Navickas, G. Tamulevičius and P. Treigys, "An Overview of Lithuanian Intonation: A Linguistic and Modelling Perspective," Informatica, vol. 33, no. 4, p. 795–832, 2022.

[13] A. M. K1 and B. A. P, "Ambiguities in Natural Language Processing," International Journal of Innovative Research in Computer, vol. II, 2014.

[14] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu and P. Fung, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," in Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2023.

[15] D. Sobania, M. Briesch, C. Hanna and J. Petke, "An Analysis of the Automatic Bug Fixing Performance of ChatGPT," in 2023 IEEE/ACM International Workshop on Automated Program Repair (APR), 2023.

[16] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes and A. Mian, "A Comprehensive Overview of Large Language Models," arXiv preprint arXiv:2307.06435, 2023.

[17] K. Huang, I. Hsu, T. Parekh, Z. Xie, Z. Zhang, P. Natarajan, K. Chang, N. Peng and H. Ji., "A Reevaluation of Event Extraction: Past, Present, and Future Challenges.," CoRR, vol. abs/2311.09562, 2023.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child and A. R. Dan, "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020.

[19] OpenAI, "ChatGPT," 2022. [Online]. Available: https://openai.com/.

[20] T. S. Timothy R. McIntosh, T. Liu, P. Watters and M. N. Halgamuge, "From Google Gemini to OpenAI Q* (Q-Star): A Survey of Reshaping the Generative Artificial Intelligence (AI) Research Landscape," arXiv preprint arXiv:2312.10868, 2023.

[21] "GitHub, KTU_Lithuanian_Events_Dataset," 2024. [Online]. Available: https://github.com/device7/KTU_Lithuanian_Events_Dataset.

[22] B. Sundheim and R. Grishman, "Message Understanding Conference - 6: A brief history," in Volume 1: The 16th International Conference on Computational Linguistics., 1996.

[23] Linguistic Data Consortium, ACE (Automatic Content Extraction) English Annotation Guidelines for Events, 2005.

[24] Google, "Google Blog," Google, 2024. [Online]. Available: https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/.

[25] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton and J. Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," in ICLR 2017, 2017.

[26] OpenAI, "Hellp GPT-4o," OpenAI, 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/.
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
[28] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in 36th Conference on Neural Information Processing Systems (NeurIPS 2022), 2022.
[29] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in ICLR 2023, 2023.
[30] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai and Q. V. Le, "Finetuned Language Models Are Zero-Shot Learners," in ICLR 2022, 2022.
[31] Y. Hou, H. Dong, X. Wang, B. Li and W. Che, "MetaPrompting: Learning to Learn Better Prompts," in Proceedings of the 29th International Conference on Computational Linguistics, 2022.
[32] S. Min, M. Lewis, H. Hajishirzi and L. Zettlemoyer, "MetaICL: Learning to Learn In Context," in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2022.

AUTHORS

ARŪNAS ČIUKŠYS received the B.Sc. and M.Sc. degrees in computer science from the Kaunas University of Technology, in 2012 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Faculty of Informatics. His research interests include machine learning, AI, information system engineering, databases, software, and information extraction.

RITA BUTKIENĖ received the B.Sc., M.Sc., and Ph.D. degrees from the Kaunas University of Technology, in 1993, 1995, and 2002, respectively. She is currently a Professor at the Kaunas University of Technology. Her research interests encompass information system engineering, ontologies, semantic technologies, databases, distributed ledgers, information extraction, and information retrieval.