LEVERAGING LARGE LANGUAGE MODELS FOR FEW-SHOT KPI EXTRACTION FROM FINANCIAL REPORTS

FRAUNHOFER PUBLICA

Tobias Deußer*[1,2], Cong Zhao [1,2], Daniel Uedelhoven [2], Lorenz Sparrenberg [1], Lars Hillebrand [2], Christian Bauckhage [1,2], and Rafet Sifa [1,2]
[1]University of Bonn, Bonn, Germany
[2]Fraunhofer IAIS, Sankt Augustin, Germany

ABSTRACT

We explore the use of Large Language Models (LLMs) for automating the extraction of Key Performance Indicators (KPIs) from diverse financial reports without any additional fine-tuning. We focus on evaluating various proprietary and open-source LLMs to address the joint named entity recognition and relation extraction tasks essential for accurately linking KPIs to their corresponding values and attributes. Our study highlights the technical challenges involved in the extraction process and presents a comprehensive evaluation of the models' effectiveness. Our results reveal significant insights into handling these LLMs in such a crucial environment and showcase the transformative potential of LLMs in enhancing financial analysis and decision-making.

Keywords Relation Extraction · Named Entity Recognition · Natural Language Processing · Machine Learning · Finance

1 Introduction

In the landscape of financial analysis, the extraction and interpretation of Key Performance Indicators (KPIs) from vast and ever-expanding repositories of financial documents have emerged as a cornerstone for investors, analysts, and regulators alike [1, 2, 3, 4, 5, 6, 7]. These documents, rich with critical data, are dispersed across numerous international databases, each serving as a vital source of insights for investors, analysts, and regulatory bodies worldwide. The task of manually extracting and analyzing Key Performance Indicators (KPIs) from this extensive collection of financial reports is not only time-consuming but also prone to errors, highlighting a pressing need for solutions that are automated, precise, and capable of operating at scale.

The advent of large language models (LLMs) offers a promising avenue for revolutionizing how financial data is extracted, analyzed, and interpreted. With their unparalleled capacity to understand and generate human-like text, LLMs present a novel opportunity to automate KPI extraction from financial documents efficiently.

In this context, we explore the potential of LLMs to redefine the landscape of financial KPI extraction. Through our comprehensive methodology, we demonstrate how these models can be applied to the KPI-EDGAR [8] dataset, resulting in significant advancements in both the accuracy and efficiency of KPI extraction, an application of joint named entity recognition and relation extraction. Our contributions are twofold: First, we provide an in-depth analysis of the challenges involved in extracting KPIs from financial reports and how LLMs can be leveraged to overcome these challenges. Second, we benchmark 9 different LLMs, both open-source and proprietary, to give practitioners and researchers a clear overview of available solutions.

By bridging the gap between LLMs and financial data analysis, this study not only sets a new benchmark for automated financial document analysis but also opens up avenues for future research and practical applications that could transform the financial industry's approach to data-driven decision-making.

*tdeusser@uni-bonn.de, ORCID-ID: 0000-0003-4685-0847

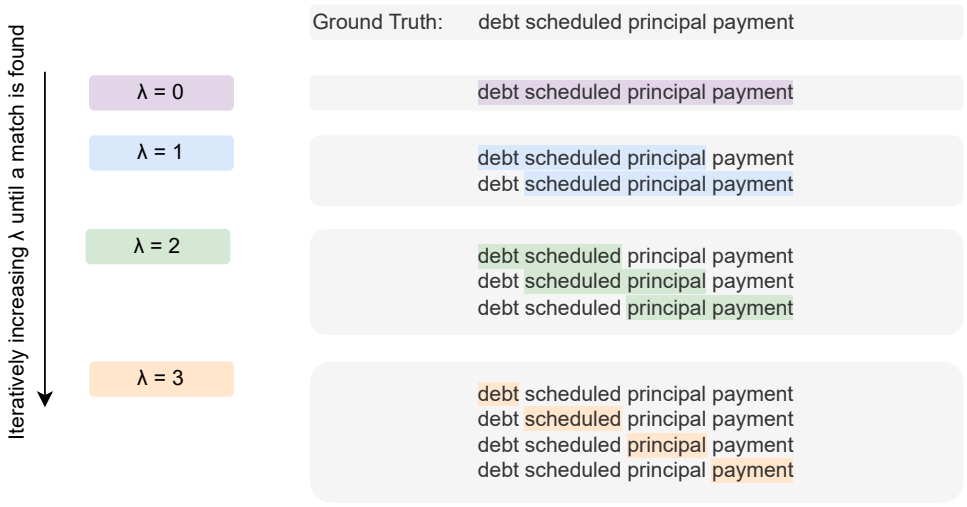

Figure 1: Illustration of range selection for possible generated entity mutations of "debt scheduled principal payment": Each iteration is depicted as a separate block, distinguished by different colors representing varying selection lengths.

2 Related Work

Language models such as BERT, GPT, and Llama have significantly impacted computational linguistics, excelling in understanding semantic and syntactic relationships within extensive text corpora [9, 10, 11]. These models are adept at complex NLP tasks including language translation, named entity recognition (NER), and sentiment analysis, all critical for financial text processing. Recent advancements also include Conditional Variational Autoencoders (CVAEs), which have demonstrated remarkable text generation capabilities for structured financial data [12]. In the financial domain, ensuring the accuracy and integrity of extracted information is paramount. Techniques such as Numerical Cross-Checking (NCC) and linguistic rule-based methods leverage standards like XBRL to verify financial claims, enhancing the reliability of analyses [13, 14]. Furthermore, LLMs have been employed to detect contradictions in financial texts, a task pivotal for assessing the consistency of reported data [15, 16]. For more specialized financial applications, models like FinBERT and its derivatives, including FinBERT-FOMC, have been tailored to perform sentiment analysis, outperforming conventional machine learning approaches [17, 18]. Addressing the need for quality training datasets, initiatives like FinEntity, FiNER, and KPI-EDGAR provide annotated financial entities and performance indicators, crucial for training domain-specific models [19, 20, 8]. Multilingual and diverse financial datasets also support global financial tasks, ensuring models are effective across various languages and financial systems [21, 22, 23].

Essential for linking KPIs to pertinent values and attributes in financial reports are domain-specific NER and relation extraction techniques. Nuanced models such as CBCP, KPI-BERT, or iNERD have been developed to tackle this challenge [24, 25, 26, 27]. These systems not only ensure accurate data retrieval but also link KPIs efficiently within financial documents. Additionally, retrieval-augmented frameworks have been introduced to boost LLMs' performance by integrating external, reliable contexts, further enhancing the precision and reliability of information extraction from financial texts [28].

This study seeks to leverage advancements in KPI extraction from financial documents, particularly focusing on the few-shot capabilities of LLMs for KPI extraction, which none of the above-mentioned works study. These capabilities offer transformative potential for automating and enhancing financial analysis and decision-making.

3 Methodology

Our main approach involves crafting suitable prompts to query multiple models and preserving the generated relational information in JSON format. We then conduct evaluations comparing predicted and ground truth relation. This process involves the specific methods of appropriate information extraction and evaluation criteria.

3.1 Generation of JSON output

During the generation phase, only three selected models (GPT-3.5 Turbo [29], GPT-4 [30], Mistral Large [31]) can enforce JSON outputs with their API calls. For models without this capability, we have to rely on the language model to produce a valid JSON output.

If the model fails to generate a valid JSON format meeting our extraction criterion, or if the content does not adhere to the relationship requirements of the KPI-EDGAR dataset (see [8]), we will penalize it with an F_1 score of 0.

3.2 Extraction of Ranges from Generated Entities

In our extraction and annotation process, our focus lies on precise matching rather than approximating agreement in meaning. Therefore, we undertake a meticulous word-by-word search for matches, aiming for exact correspondence.

Initially, we perform a one-to-one matching based on the length l of the model's predicted entity. Starting with the first word, we compare it with the corresponding word of the ground truth. If a match is found, we proceed to check the subsequent words in both the model's generation and the original text. This process continues until the end of the predicted entity is reached. If a complete match is achieved, we return the range of the model's answer within the original sentence.

In cases where a complete match is not found, we gradually reduce the length l to search for potential partial matches. We create entity subsets with length $l - \lambda$ out of the model's predicted entity, starting from $\lambda = 1$ and iteratively increase λ until $\lambda = l - 1$. In those subsets, we search for matching ranges word-by-word, as described above. This iterative process is illustrated in Figure 1. If we cannot find even an approximate match of the entity, we penalize once again with an F_1 score of 0.

3.3 Fuzzy Word Matching

When comparing words against words, we must also account for special circumstances to ensure that important information is preserved without compromising accuracy. Ideally, two strings are considered equal when they are precisely identical in every aspect. However, there are instances where generated entities or ground truth entities are noisy. This might include an enumeration, a different tense, or inconsistent numerical formatting like a different floating point precision or adding a numerical multiplier (e.g. 1.5 *million* instead of 1,500,000). In these cases, we still consider the string matching successful, as crucial information is retained.

3.4 Prompt Design

In the prompt, we explicitly outlined our requirements for extracting key performance indicators (KPIs) and their corresponding values from the provided text and formatting them in JSON. Then, we describe the named entity classes present in the dataset. Meanwhile, we add several example sentences along with the ground truth entities. Additionally, we ensure that each named entity category is exemplified at least once, enriching the model's learning experience by providing specific instances for each category, thus reinforcing both conceptual understanding and practical application.

Throughout our experiment, we iteratively adjusted the prompt using insights from the generated model outputs to enhance performance further. This includes prioritizing numerical formatting in the output to align with ground truth, emphasizing the extraction of precisely the original text content, underscoring the one-to-one relationship requirement, and reinforcing the JSON format as the answer at the end of the prompt. The prompt satisfying these requirements is available in the Appendix.

4 Results

In the evaluation of nine diverse large language models on the KPI-EDGAR [8] dataset, as depicted in Table 1, we observed a broad range of results. First, proprietary models outpaced their open-source competitors by a significant margin, when it comes to their few-shot performance. Mistral-Large [31] demonstrated the highest F_1-score of 30.75. GPT-4 [30] and GPT-3.5-turbo [29] followed with F_1-scores of 26.91 and 23.89, respectively. Open-source models like Llama-3 [33] and Mixtral [34] displayed lower performance, with F1-scores of 21.30 and 14.48, respectively. These results show that the current state of open-source large language models still lags behind proprietary models when a task like KPI extraction is considered. Interestingly, these results do not align with the current rankings of LLMs[1], indicating that the conversational performance of LLMs does not directly translate to their capability in other tasks.

[1] See, for example, the "LMSYS Chatbot Arena Leaderboard" [37, 38].

Table 1: Few-Shot Performance

Model	Size	Open-Source	Precision	Recall	F_1
Few-shot					
GPT-3.5-turbo [29]	N.A.	No	30.41	19.68	23.89
GPT-4 [30]	N.A.	No	33.03	22.71	26.91
Llama-2 [32]	70B	Yes	19.23	3.78	6.31
Llama-2 [32]	13B	Yes	26.90	9.91	14.49
Llama-2 [32]	7B	Yes	10.32	9.83	10.07
Llama-3 [33]	70B	Yes	23.75	19.31	21.30
Mistral-Large [31]	N.A.	No	**35.00**	**27.42**	**30.75**
Mixtral [34]	8x7B	Yes	26.70	9.94	14.48
Zephyr-β [35]	7B	Yes	18.78	8.19	11.40
Full fine-tuning					
KPI-BERT* [25, 8]	0.1B	Yes	-	-	40.70
KPI-BERT-wCR* [36]	0.1B	Yes	-	-	43.67

Evaluation of the few-shot performance of various Large Language Models on the test set of KPI-EDGAR [8]. Scores are calculated as described in [8]. Models marked with a star* are *not* few-shot and are, therefore, only to a limited extent comparable to the other models. The results of these are taken from their respective studies.

A further observation is that even though open-source large language models lag behind their proprietary counterparts, they in turn can easily be beaten by much smaller open-source models [25, 8, 36], but one has to fine-tune them on the train set, making the comparison to the few-shot evaluation conducted in this study a difficult one.

5 Conclusion

In this study, we evaluated the capabilities of nine large language models (LLMs) on the KPI-EDGAR [8] dataset, providing insights into how well a task like KPI extraction, previously only tackled with fine-tuning models on labeled data, can be solved in a few-shot setting. Notably, proprietary models such as Mistral-Large [31] and GPT-4 [30] outperformed their open-source counterparts. These findings indicate that proprietary models may be better optimized for specific tasks like KPI extraction, which remains a challenge for many open-source LLMs like Llama-3 [33].

However, our results also suggest that model performance in few-shot settings does not universally translate across different types of tasks. Despite their smaller size, open-source models like KPI-BERT [25, 8], when fully fine-tuned, surpassed even the best-performing LLMs in this setting. This underscores the potential of fine-tuning as a critical factor for enhancing model performance on specialized tasks, albeit at the cost of generalizability and the requirement of extensive task-specific data, which might not always be available.

These observations lead us to recommend that developers and researchers consider both the size and the training methodology of language models based on their specific needs and constraints. For tasks requiring high precision and specialization, like the one at hand, fine-tuning smaller models might prove more beneficial, while few-shot capabilities of larger models could be leveraged for broader, less specific applications.

The next step could be investigating alternative ways to leverage LLMs for the relation extraction task, like the methodology proposed in [27], which studied how LLMs can be fine-tuned to do named entity recognition, which traditionally is the first step in extracting relations. Another potential avenue for further research is applying additional pre-training with vast databases like EDGAR to existing LLMs, so that they improve their performance on finance-specific datasets, which will likely lead to an increase in "downstream" tasks like KPI extraction.

Acknowledgments

This research has been partially funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

References

[1] Ganna Demydyuk. Optimal financial key performance indicators: evidence from the airline industry. *Accounting & Taxation*, 3(2):39–51, 2012.

[2] Hany Elzahar, Khaled Hussainey, Francesco Mazzi, and Ioannis Tsalavoutas. Economic consequences of key performance indicators' disclosure quality. *International Review of Financial Analysis*, 39:96–112, 2015. ISSN 1057-5219.

[3] Peter J Harris and Marco Mongiello. Key performance indicators in european hotel properties: general managers' choices and company profiles. *International Journal of Contemporary Hospitality Management*, 13:120–128, 2001.

[4] Brian P. McCullough and Galen T. Trail. Assessing key performance indicators of corporate social responsibility initiatives in sport. *European Sport Management Quarterly*, 23:82–103, 2023.

[5] Sparsh Johari Neelu Nandan Vibhakar, Kamalendra Kumar Tripathi and Kumar Neeraj Jha. Identification of significant financial performance indicators for the indian construction companies. *International Journal of Construction Management*, 23:13–23, 2023.

[6] David Parmenter. *Key performance indicators: developing, implementing, and using winning KPIs*. John Wiley & Sons, 2015.

[7] Satish Sharma, Mikhail Shebalkov, and Andrey Yukhanaev. Evaluating banks performance using key financial indicators—a quantitative modeling of Russian banks. *The Journal of Developing Areas*, pages 425–453, 2016.

[8] Tobias Deußer, Syed Musharraf Ali, Lars Hillebrand, Desiana Nurchalifah, Basil Jacob, Christian Bauckhage, and Rafet Sifa. KPI-EDGAR: A novel dataset and accompanying metric for relation extraction from financial documents. In *Proc. ICMLA*, pages 1654–1659, 2022.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proc. NAACL-HLT*, pages 4171–4186, 2019.

[10] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023.

[12] Ziao Wang, Yunpeng Ren, Xiaofeng Zhang, and Yiyuan Wang. Generating long financial report using conditional variational autoencoders with knowledge distillation. *Transactions on Artificial Intelligence*, 2024.

[13] Yixuan Cao, Hongwei Li, Ping Luo, and Jiaquan Yao. Towards automatic numerical cross-checking: Extracting formulas from text. In *Proc. WWW*, 2018.

[14] Sachin Pawar, Manoj Apte, Aditi Pawde, Sushodhan Vaishampayan, Girish Palshikar, and Akshada Shinde. Dig here! extracting and using knowledge from financial audit reports. In *Proc. FLAIRS*, volume 36, 2023.

[15] Tobias Deußer, Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. Contradiction detection in financial reports. In *Proc. NLDL*, volume 4, 2023. doi:10.7557/18.6799.

[16] Tobias Deußer, David Leonhard, Lars Hillebrand, Armin Berger, Mohamed Khaled, Sarah Heiden, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. Uncovering inconsistencies and contradictions in financial reports using large language models. In *Proc. BigData*, pages 2814–2822. IEEE, 2023.

[17] Allen H. Huang, Hui Wang, and Yi Yang. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841, 2023.

[18] Sandro Gössi, Ziwei Chen, Wonseong Kim, Bernhard Bermeitinger, and Siegfried Handschuh. FinBERT-FOMC: Fine-tuned finbert model with sentiment focus method for enhancing sentiment analysis of fomc minutes. In *Proc. ICAIF*, pages 357–364, 2023.

[19] Yixuan Tang, Yi Yang, Allen Huang, Andy Tam, and Justin Tang. FinEntity: Entity-level sentiment classification for financial texts. In *Proc. EMNLP*, pages 15465–15471, 2023.

[20] Agam Shah, Ruchit Vithani, Abhinav Gullapalli, and Sudheer Chava. FiNER: Financial named entity recognition dataset and weak-supervision model, 2023.

[21] Ali Jabbari, Olivier Sauvage, Hamada Zeine, and Hamza Chergui. A French corpus and annotation schema for named entity recognition and relation extraction of financial news. In *Proc. LREC*, pages 2293–2299, 2020.

[22] Haoyu Wu, Qing Lei, Xinyue Zhang, and Zhengqian Luo. Creating a large-scale financial news corpus for relation extraction. In *Proc. ICAIBD*, pages 259–263. IEEE, 2020.

[23] Rasmus Jørgensen, Oliver Brandt, Mareike Hartmann, Xiang Dai, Christian Igel, and Desmond Elliott. MultiFin: A dataset for multilingual financial NLP. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 894–909, 2023.

[24] Lang Cao, Shihua Zhang, and Juxing Chen. CBCP: A method of causality extraction from unstructured financial text. In *Proc. NLPIR*, pages 135–140, 2021.

[25] Lars Hillebrand, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. KPI-BERT: A joint named entity recognition and relation extraction model for financial reports. In *Proc. ICPR*, pages 606–612, 2022.

[26] Lars Hillebrand, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. Towards automating numerical consistency checks in financial reports. In *Proc. BigData*, pages 5915–5924. IEEE, 2022.

[27] Tobias Deußer, Lars Hillebrand, Christian Bauckhage, and Rafet Sifa. Informed named entity recognition decoding for generative language models, 2023.

[28] Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proc. ICAIF*, page 349–356, 2023.

[29] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proc. NeurIPS*, volume 33, pages 1877–1901, 2020.

[30] OpenAI. GPT-4 technical report, 2023.

[31] Mistral AI Team. Mistral large, February 2024. URL `https://mistral.ai/news/mistral-large/`. Accessed: 2024-04-25.

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

[33] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[34] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts, 2024.

[35] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment, 2023.

[36] Tobias Deußer, Cong Zhao, Wolfgang Krämer, David Leonhard, Christian Bauckhage, and Rafet Sifa. Controlled randomness improves the performance of transformer models. In *Proc. ICMLA*, pages 1805–1810, 2023.

[37] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[38] LMSys Team. Lmsys chatbot arena leaderboard, May 2024. URL `https://arena.lmsys.org/`. Accessed: 2024-05-07.

Appendix

Final Prompt

Formatted KPI Extraction Prompt

Extract Key Performance Indicators and their corresponding numerical value from the following sentence in JSON format. The format is: `{"relation_id": {"named_entity_class": "named_entity_value", "named_entity_class": "named_entity_value"}, ...}`. Here `relation_id` can be multiple, such as 0, 1, 2, 3, 4, 5, etc. Ensure that there are exactly two key-value pairs in `{"named_entity_class": "named_entity_value", "named_entity_class": "named_entity_value"}` The value of a key-value pair can be primarily numbers, and if that's not possible, then strings.
Examples are:
1. Example
"Includes $ 6.7 billion of revenue recognized in 2021 that was included in deferred revenue as of September 26, 2020, $ 5.0 billion of revenue recognized in 2020 that was included in deferred revenue as of September 28, 2019, and $ 5.9 billion of revenue recognized in 2019 that was included in deferred revenue as of September 29, 2018." from which you should extract: {"0": {"KPI": "revenue", "value current year": 6.7}, "1": {"KPI": "revenue", "value two years ago": 5.9}, "2": {"KPI": "revenue", "value previous year": 5.0}}.
... (additional examples continue in the same format) ...
Named entity classes are only allowed from the following range: "KPI", "value current year", "value previous year", "value two years ago", "current year value increase", "previous year value increase", "current year value decrease", "previous year value decrease", "thereof", "attribute", "KPI-Coreference".
Here are the descriptions for the named entity classes:

- "KPI": Key Performance Indicators expressible in numerical and monetary value, e.g. revenue or net sales
- "value current year": Current Year monetary value of a KPI
- "value previous year": Prior Year monetary value of a KPI
- "value two years ago": 2 Year Past Value of a KPI
- "current year value increase": Increase of a KPI from the previous year to the current year
- "previous year value increase": Analogous to increase, but from value two years ago to value previous year
- "current year value decrease": Decrease of a KPI from the previous year to the current year
- "previous year value decrease": Analogous to decrease, but from value two years ago to value previous year
- "thereof": Represents a subordinate KPI, i.e., if a KPI is part of another, broader KPI
- "attribute": Attribute that further describes a KPI
- "KPI-Coreference": A co-reference to a KPI mentioned in a previous sentence

Please follow the examples and description to extract Key Performance Indicators and generate a consistent JSON format.