Zero-Shot Award Criteria extraction via Large Language Models from German Procurement Data from Switzerland

Veton Matoshi [1,3*] **Luca Rolshoven** [1,2] **Matthias Stürmer** [1,2]

[1]**Bern University of Applied Sciences** [2]**University of Bern** [3]**University of Zurich**

Abstract

Public procurement serves as a model for sustainable practices (Sönnichsen and Clement, 2020). Recent legislation in Switzerland mandates considerations of economic, environmental, and social responsibility in public spending, including within the realm of public procurement. To assess the extent to which these legislative measures have influenced public procurement practices, one may examine Award Criteria (ACs) based on which procuring entities determine the most suitable bidder. This paper demonstrates the potential of Natural Language Processing (NLP) for extracting ACs from Swiss calls for tenders (CFTs), specifically those in German. We evaluate the efficacy of a German Large Language Model (LLM) in executing four tasks with a single zero-shot prompt: (1) Text Classification (TC), determining whether a call for tenders (CFT) includes ACs; (2) Named Entity Recognition (NER), identifying ACs and other related named entities; (3) Relation Extraction (RE), elucidating relationships between named entity instances; and (4) Formatting, compiling the information into a structured JSON format. We evaluate our approach on a set of 167 annotated CFTs[1]. This approach facilitates the automated monitoring and evaluation of ACs overtime regarding sustainability. Both our code and the annotated dataset are publicly available: https://github.com/kapllan/GATE-CH.

1 Introduction

In Switzerland, public procurement, worth about 41 billion CHF annually, has significant implications (Federal Council, 2017). Its impact extends beyond the economy, influencing the private sector and serving as a model for sustainable practices (Sönnichsen and Clement, 2020). Recently, Swiss procurement laws have evolved to emphasize sustainability (Steiner and Klingler, 2023); the 2021 Federal Law on Public Procurement (PPA) and the Inter-cantonal Agreement (IAPP) mandate economic, environmental, and social responsibility in public spending (art. 2 PPA/IAPP). New standards include obligatory environmental and social criteria (art. 12), technical specifications for environmental protection (art. 30 PPA), and awarding criteria prioritizing sustainability and life cycle costs (art. 29). Ultimately, the law empowers authorities to value sustainability over cost in their decisions (art. 41), embedding sustainability into the nation's procurement practices at all levels (Koch, 2020). Additionally, the *2030 Agenda for Sustainable Development* (United Nations, 2015) is a global framework adopted by all United Nations Member States in 2015. It is a comprehensive plan of action aimed at ending poverty, protecting the planet, and ensuring that all people enjoy peace and prosperity by the year 2030. The agenda is anchored by 17 *Sustainable Development Goals* (SDGs). Sustainable public procurement is a significant element in achieving the SDGs. A focus here is on promoting sustainability criteria within public procurement.

Documents pertaining to calls for tenders (CFTs) in procurement processes encompass a vast amount of information, among which we will focus on Award Criteria (ACs). ACs refer to the standards and factors that a procuring entity uses to evaluate and compare the bids or proposals submitted by bidders. These criteria are crucial for making the decision on with which bidders to enter into a contract. ACs are designed to identify the most economically advantageous offer, taking into account various aspects beyond just the price. They ensure that the procurement process is fair, transparent, and yields the best value for money. Extracting ACs allows us to assess the growing importance of sustainability over the years.

However, the automated identification and extraction of ACs pose challenges due to several factors, most notably: (1) the diversity of formats, in-

[*] Main contribution.

[1]The link to the repository containing the dataset will be made available upon acceptance of this paper.

cluding PDF, DOCX, and XLSX; (2) the extensive volume of pages and information, complicating the identification of the page containing the relevant criteria; and (3) the lack of uniform presentation of the criteria set, as evidenced by figures 3, 4, and 5 in the Appendix. ACs are often presented in a structured form, such as tables. However, the application of Optical Character Recognition (OCR) on PDF documents typically results in the loss of structural information, thereby impeding its utility for extraction.

In this paper, we aim to assess the usage of NLP methods for performing four tasks: (1) Text Classification (TC), determining whether a CFT includes ACs; (2) Named Entity Recognition (NER), identifying ACs and other related named entities; (3) Relation Extraction (RE), elucidating relationships between named entity instances; and (4) Formatting, compiling the information into a structured JSON format. For this purpose, we have annotated ACs within 167 public CFTs from IntelliProcure, a Swiss data platform for public procurement used for previous research work (Stuermer et al., 2017; Welz and Stuermer, 2020, 2021; Orset, 2024). This data was downloaded from simap.ch[2]. Although Swiss procurement data is multilingual, encompassing German, French, and Italian, this work will concentrate on CFTs written in German. To this end, and in line with recent efforts to utilize open Large Language Models (LLMs) (Gunasekar et al., 2023; Li et al., 2023; Team Gemma and Google Deepmind, 2024), we will leverage a smaller-sized 8x7 Billion parameter Mixture-of-Experts (Jacobs et al., 1991; Jordan and Jacobs, 1994) LLM for German[3]. We aim to demonstrate how the automatic extraction of ACs and related information provides preliminary insights into the degree to which recent legislation has influenced the incorporation of sustainability aspects into the definition of ACs. This paper presents the preliminary work of SNSF project 10000100.

The paper is structured as follows: Section 2 provides an overview of related work on the use of NLP methods for analyzing procurement data; Section 3 describes the dataset creation and annotation process; Section 4 details the methods used to perform the aforementioned tasks; Section 5 presents frist preliminary results of our information extraction pipeline; and Section 7 concludes with a summary and future perspectives.

2 Related Work

As previously mentioned, CFTs include numerous details, thereby enhancing the likelihood of application scenarios for machine learning techniques, specifically NLP.

A range of studies has explored the use of NLP in analyzing public CFTs. Álvarez et al. (2011) focuses on query expansion methods and performance evaluation for retrieving public procurement notices, emphasizing the use of semantics and linking open data. Locatelli et al. (2023) developed a BERT-based multi-label text classifier to translate quality demands in Italian public tenders, supporting consensus building. Rabuzin and Modrušan (2019) and Modrušan et al. (2020) focus on using machine learning methods, such as Naïve Bayes, Logistic Regression and Support Vector Machines, to detect suspicious tenders from Croatia. Endtner and Stürmer (2019) present a methodology using machine learning to extract suitability criteria from CFTs on the Swiss public procurement platform simap.ch. By annotating CFTs to identify relevant sections and employing RandomForest models for classification, the study demonstrates the feasibility of automating the extraction of critical information, offering insights into incorporating sustainability criteria in procurement processes.

To date, the application of LLMs in analyzing procurement data remains underexplored, despite recent advancements in leveraging LLMs for document understanding. Ye et al. (2023) unveiled mPLUG-DocOwl, a pioneering model that surpasses existing multi-modal models in understanding documents without the need for OCR. Furthermore, Wang et al. (2023a) developed DocLLM, an innovative lightweight extension designed to enhance the capabilities of traditional LLMs in interpreting visual documents. Unlike conventional multimodal LLMs, DocLLM uniquely eschews costly image encoders, opting instead to focus on bounding box data to integrate spatial layout understanding. This approach is particularly pertinent given the prevalence of ACs information within table-like formats in many CFTs. In this context, Chen (2022)'s research is noteworthy, demonstrating LLMs' proficiency in conducting sophisticated

[2]Simap.ch is an official Swiss online platform that centralizes public procurement notices, allowing government entities to publish tenders and suppliers to access and respond to these opportunities, thus facilitating transparent and competitive public procurement processes in Switzerland.

[3]https://huggingface.co/VAGOsolutions/SauerkrautLM-Mixtral-8x7B

reasoning over table structures and achieving robust performance with minimal input, as evidenced by a successful 1-shot demonstration.

3 Data and Annotation

To evaluate the performance of the LLM, it was necessary to create an annotated dataset. For the current study, we focused on PDF documents, since criteria are typically listed in PDF format. Four pieces of information related to ACs need to be identified:

Award Criterion (AC): The criterion itself, such as *Preis* 'price', *Qualität* 'quality', *Präsentation* 'presentation' etc. Frequently, in CFTs, main ACs are divided into several sub-criteria (cf. Figure 6). For the study at hand, we grouped main criteria and sub-criteria into one category.

Award Criterion Identifier (ID): Frequently, criteria and pertinent sub-criteria are numbered and thus have an identifier. In most cases, these are prefixed by the abbreviation *ZK* (ger. 'Zuschlagskriterium'), as in *ZK, ZK2, ZK2.1* etc.

Weight of Award Criterion (W): Weights of ACs in CFTs refer to the relative importance assigned to each criterion used to evaluate bids. These weights are expressed as percentages and guide the decision-making process by indicating how much each criterion will influence the final selection of a bidder. For instance, price might be weighted at 40%, indicating it comprises 40% of the overall evaluation, while quality might be weighted at 30%, and delivery time at 30%.

Maximal Number of Points of Award Criterion (MNP): MNP in CFTs specifies the highest score that can be allocated to each evaluation criterion used in assessing bids. This system quantifies the assessment, allowing for a detailed comparison of how well each bid meets the specified criteria. For the evaluation of the criteria, points are preferred, although in the end, the weight and points per criterion are offset against each other to obtain a final point total.

We randomly downloaded CFTs from IntelliProcure without regard to domain or year. During sampling, we took care to avoid duplicate documents or those that were overly similar to each other.

Before commencing annotation, we needed to identify pages listing ACs. As mentioned earlier, CFTs can span up to thousands of pages filled with diverse information. To pinpoint pages containing ACs, we utilized regular expressions to search for the pattern "(ZK|Zuschlagskrit)" (*ZK* being the abbreviation for *Zuschlagskriterium*, the German term for ACs). Although a match did not guarantee that the page in question listed the ACs, this approach significantly expedited the process. For annotation, we employed the open-source tool INCEpTION (Klie et al., 2018)[4]. Initially, we annotated spans in the text that mentioned all relevant named entities, namely ACs, W, IDs, and MNP. Subsequently, we annotated the relations between these named entities. There was only one relation, which we named "belongs_to". This was crucial for assigning the ID, W, and MNP to a specific AC, as illustrated in Figure 7, in the Appendix.

The annotations were performed by a data scientist who had become acquainted with procurement data over a period of several weeks. Subsequently, a second data scientist checked the annotations and provided feedback. We considered this approach sufficient for this interim study, as the identification of ACs does not require expert knowledge. However, in the future, more rigorous annotation approaches need to be employed.

Once a certain number of documents had been annotated, we trained a TC model in a few-shot setting to identify pages listing all ACs (for more details, cf. Section 4.1). If the model performed well, it was used to pre-select additional pages for annotation. Otherwise, we continued with the previous approach.

Overall, we performed named entity and relation annotation on 167 documents. Section 3 depicts the frequency of each named entity class in our dataset. This overview highlights irregularities in public CFTs; crucial information, such as W or the MNP, which is essential for identifying the most suitable bidder, is not always specified in the documents.

NER	Frequency
Award Criteria	1120
Weight	888
Criterion ID	787
Max. Number of Points	348

Table 1: Frequency of each named entity classes in the annotated dataset.

A cursory examination of the ACs specified in the dataset reveals that *price* is by far the most frequent and important criterion, cf. Table 2, whereas sustainability occupies the fourth rank.

[4]https://inception-project.github.io/

Award Criterion	Frequency
Price	129
References	21
Quality	17
Sustainability	10
Dates/Appointments	10

Table 2: Overview of the most frequent ACs

For each document, we converted the annotations into a list of dictionaries. Each dictionary contained comprehensive information about the AC, including the ID, W, and MNP. In instances where any of the aforementioned details were absent, we inserted an empty string. This approach maintained a consistent number of key-value pairs for each entry. An illustration of this structured output is presented in Figure 1.

Given the excessive length of the entire documents, both tasks necessitated a context window. The annotation tool, INCEpTION, performs automatic sentence splitting, thus allowing to define all sentences containing information related to ACs as the context window. To ensure no critical surrounding information was overlooked, we introduced a randomized padding of 300, 400, or 500 characters. Furthermore, to secure a substantial quantity of negative examples, we included 161 pages that were randomly chosen and devoid of AC information. These pages were compiled by selecting, at random, three to ten consecutive sentences lacking ACs information.

The dataset was utilized for two primary objectives: (1) To train and evaluate a TC model aimed at identifying pages that either contain or do not contain information related to ACs. (2) To assess the capability of LLMs in extracting information pertaining to ACs into a computer-readable format, specifically JSON. Ultimately, for the training of the TC model, we implemented a 20-20-60 split for Train-Validation-Test purposes.

4 Model Training and Prompting

4.1 Award Criteria Presence Detection

As previously stated, the majority of documents within a CFT contain information apart from the ACs. Therefore, it is imperative to have a mechanism that accurately distinguishes between documents, specifically, pages that contain information related to ACs. To this end, we utilized the LLM to determine the presence of ACs within a given context window by providing a simple prompt, as detailed in Appendix A.3.1. However, employing LLMs demands significant resources. As a practical alternative for real-world applications, we trained a more lightweight TC model. This model only needs to distinguish between two labels: *has criteria* and *has no criteria*.

Since we had only a limited number of training samples, we leveraged the few-shot TC paradigm Sentence Transformer Finetuning (SetFit) (Tunstall et al., 2022). SetFit enhances TC by first fine-tuning a pre-trained Sentence Transformer on a compact set of text pairs, using a contrastive Siamese network approach to understand nuanced differences and similarities. This refined model produces detailed text embeddings that capture the essence and context of the text. These embeddings are subsequently utilized to train a classification head, enabling it to accurately categorize text into predefined classes based on learned textual patterns and characteristics.

As our dataset consisted of German CFTs only, we employed a German Sentence Transformer Model[5]. We trained our model to optimize the Macro-F1 score on the validation set for 3 epochs, using a learning rate of 5e-5.

4.2 Award Criteria Information Extraction

We leverage the LLM to extract AC information and convert it into a unified JSON format. This process encompasses three tasks: (1) NER: Identifying the text spans that mention information related to ACs. (2) RE: Determining which pieces of information related to ACs are associated with each other. (3) Formatting: Converting the results from NER and RE into a unified JSON format. The scope of these tasks is depicted in Figure 1. The ultimate goal is to create a list of JSON outputs whose key-value pairs contain the AC and the respective ID, W, and MNP.

LLMs have been shown to effectively manage tasks such as NER (Wang et al., 2023b), RE (Wadhwa et al., 2023), and the conversion of inputs from one format to another, most notably from text to SQL (Qin et al., 2022), utilizing natural language prompts. Although it is acknowledged that LLMs can handle each of these tasks individually, our study aimed to address them collectively through a single prompt. We employed four distinct prompts

[5]https://huggingface.co/PMAI/biencoder_msmarco_bert-base_german

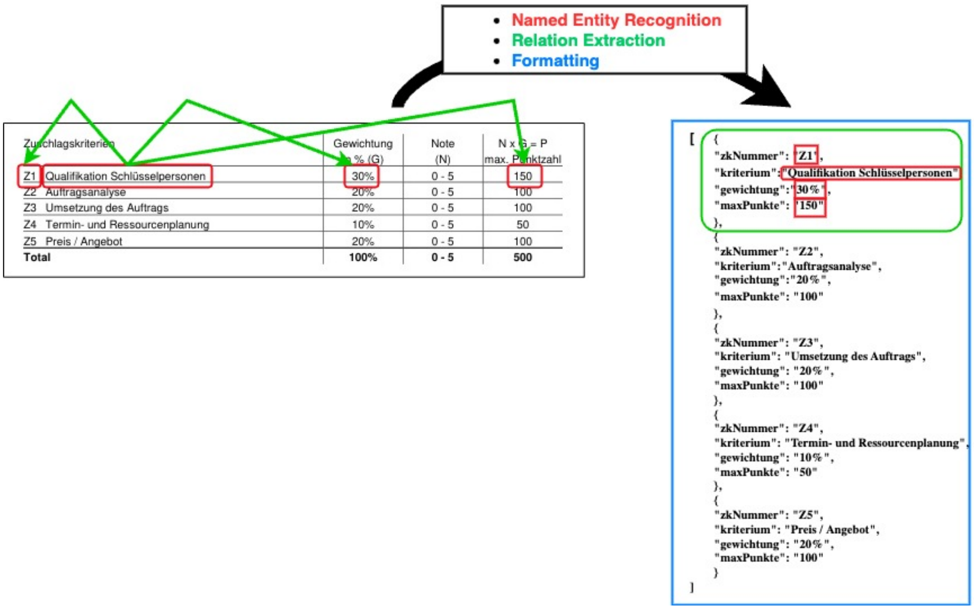

Figure 1: Depiction of the three tasks the LLM is required to perform after identifying the presence of ACs in a given context: (1) Named Entity Recognition: Extraction of named entities; (2) Relation Extraction: Identification of which instances of named entities belong together; and (3) Formatting: Generation of a Python list of dictionaries, where the key denotes the named entity class and the value represents the instance of the named entity class.

written in German, varying in detail, as referenced in Appendix A.3.2. These prompts, along with excerpts from the CFT document — whether or not they included ACs — were input into the LLM, which was then instructed to return only the JSON output. As the LLM returns a string, we utilize the function 'literal_eval' provided by the ast (Abstract Syntax Trees) module of Python, to turn the string output into the desired format, i.e., a list of dictionaries. During our experiments, we observed that the LLM often provided verbose responses by providing additional explanations. Consequently, we developed a script to remove entries in the model's string that might cause errors during parsing. However, this conversion was not always successful. In cases where the conversion failed, we deemed the output as useless and assigned the score 0 for all metrics during the evaluation.

Metric	Validation	Test
ACC	96.12	93.08
Macro-F1	96.12	93.08
Micro-F1	96.12	93.08
Macro-P	96.15	93.08
Micro-P	96.12	93.08
Macro-R	96.1	93.09
Micro-R	96.12	93.08
MCC	92.25	86.16

Table 3: Results of Award Criteria Identification Using the SetFit Model.

5 Results

5.1 Award Criteria Presence Detection

As expected, both methods for detecting the presence of ACs, namely TC model fine-tuning and LLM prompting, yielded good results. Refer to Table 3 for the outcomes based on the fine-tuned SetFit model, and Table 4 for the results from zero-shot prompting with LLM. Despite employing a rather simplistic prompt without additional guidance on identifying ACs—which is generally unnecessary, as the documents usually contain the specific term for ACs, namely, in German, *Zuschlagskriterien*—the LLM managed to achieve results comparable to those of the SetFit model across the Validation and Test sets, as well as the entire dataset. However, as previously mentioned, given the relatively straightforward nature of the task, the more lightweight and efficient SetFit approach is preferred for real-world scenarios.

5.2 Award Criteria Information Extraction

For this study, we did not emphasize locating information within the text body, a typical element of NER tasks. Instead, our evaluation involved a three-pronged comparison between the ground

Metric	Validation	Test	All
ACC	91.47	93.08	92.07
Macro-F1	91.37	93.04	92.01
Micro-F1	91.47	93.08	92.07
Macro-P	92.86	93.64	93.09
Micro-P	91.47	93.08	92.07
Macro-R	92.86	93.64	93.09
Micro-R	92.86	93.64	93.09
MCC	84.11	86.63	85.02

Table 4: Results of Zero-Shot Prompting for Award Criteria Identification Using a LLM. *All* refers to the entire dataset.

truth JSONs and those generated by the model with regard to the three previously mentioned tasks the LLM has to do, namely: NER, RE and Formatting.

5.2.1 Named Entity Recognition

For evaluating NER only, we compared the extracted entities against a verified ground truth for specified labels, such as AC, ID, W, and MNP. We selected extracted entities, i.e. the value in each Python dictionary, relevant to each label from both the ground truth and the predictions, subsequently transforming these entities into binary indicators within a multi-label classification evaluation framework. For each label, we calculated key metrics—accuracy, precision, recall, and the F1 score—by treating each unique entity as an individual class. Ultimately, we compiled the outcomes for each label across all documents using the arithmetic mean.

The results are depicted in Table 5, evaluated across three versions of our dataset: (1) the entire dataset, encompassing contexts both with and without ACs; (2) a subset containing only contexts with ACs (positive examples); and (3) a subset comprising contexts without ACs (negative examples). This segmentation is crucial for assessing the model's effectiveness in handling negative examples. While the LLM demonstrates proficiency in identifying the presence of ACs (cf. Section 5.1), its performance diminishes when faced with additional instructions, particularly with the negative examples subset. Here, the model tends to extract ACs in almost any context, despite specific instructions to do so only if ACs are present. A detailed examination of the raw output uncovers numerous instances where the model provides additional explanations, often stating that ACs are actually not mentioned in the provided context. Yet, it sometimes mistakenly classifies unrelated information as ACs (see Example 1 in the Appendix) or makes conjectures about missing details (see Example 2 in the Appendix). On the other hand, when ACs are indeed present, the model confidently extracts instances of each named entity class. Nonetheless, it excels at recognizing numeric information, such as ID, MNP, and W, whereas the extraction of ACs—the most critical information—lags. Notably, Prompt 4 emerges as the most effective, likely due to its explicit hints on detecting each named entity class within a document.

P	Subset	AC	ID	MNP	W
1	mixed	33.43	36.02	41.06	44.42
2	mixed	35.83	37.09	43.24	46.79
3	mixed	37.40	39.79	41.84	26.03
4	mixed	**42.90**	**46.31**	**51.63**	**53.53**
1	positive	63.85	68.96	78.85	85.45
2	positive	64.38	66.85	78.93	**85.91**
3	positive	60.28	64.38	64.21	33.16
4	positive	**64.5**	**71.19**	**81.64**	85.38
1	negative	1.86	1.86	1.86	1.86
2	negative	6.21	6.21	6.21	6.21
3	negative	13.66	14.29	18.63	18.63
4	negative	**20.50**	**20.50**	**20.50**	**20.50**

Table 5: Results of Zero-Shot Prompting for Award Criteria NER Using a LLM. We provide the Macro-F1 score. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: 'mixed' indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. 'positive', or those lacking ACs, i.e. 'negative'. The highest values for each subset are presented in **bold**.

5.2.2 Relation Extraction

This method begins by identifying predefined labels of interest, transforming the ground truth and model predictions into flattened strings that encode both keys and their associated values. These strings are then treated as unique classes in a multi-label classification framework, allowing for the creation of binary indicators that reflect the presence or absence of each key-value pair across the dataset. By employing this technique, we are able to compute a comprehensive set of metrics. This strategy offers a detailed analysis of the model's performance with regard to both NER and RE, focusing not only on the identification of relevant named entities in the text, but also on the correct identification of the

P	Subset	Macro-F1	Micro-F1
1	mixed	20.33	23.57
2	mixed	22.35	25.43
3	mixed	12.93	14.18
4	mixed	**32.10**	**35.24**
1	positive	38.74	45.10
2	positive	37.91	43.96
3	positive	12.81	15.28
4	positive	**43.89**	**50.05**
1	negative	1.24	1.24
2	negative	6.21	6.21
3	negative	13.04	13.04
4	negative	**19.88**	**19.88**

Table 6: Evaluation Results on NER and RE. We provide the Macro- and Micro-F1 score. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: 'mixed' indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. 'positive', or those lacking ACs, i.e. 'negative'. The highest values for each subset are presented in **bold**.

relations that hold between these named entities.

The results are depicted in Table 6. The overall trend is similar to that observed in the NER evaluation, though the overall scores significantly decline due to a more stringent evaluation strategy: The model is required not only to identify all instances of each entity class but also to discern the relationships between each instance. As observed previously, the model generates a considerable number of false positives when analyzing negative examples. Prompt 4 emerges as the most effective prompt.

5.2.3 Formatting

Formatting was treated as a binary classification task, deemed successful upon extracting a Python list containing dictionaries, and unsuccessful otherwise. This aspect of formatting had been incorporated into the other evaluation strategies (cf. Section 5.2.1 and Section 5.2.2) by assigning the value 0 to each metric in the event of conversion failure. Nevertheless, Table 7 offers a comprehensive overview of conversion success rates. Generally, success rates are similarly high across all prompts and data subsets.

P	Subset	Successful	Failed
1	mixed	326	2
2	mixed	327	1
3	mixed	326	2
4	mixed	326	2
1	positive	166	1
2	positive	166	1
3	positive	166	1
4	positive	166	1
1	negative	160	1
2	negative	161	0
3	negative	160	1
4	negative	160	1

Table 7: Evaluation Results on Formatting. We provide the overall counts. The column *P* denotes the specific prompt utilized in the analysis. *Subset* distinguishes the datasets used: 'mixed' indicates the entire dataset, while subsets are specified as either containing contexts with ACs, i.e. 'positive', or those lacking ACs, i.e. 'negative'.

6 Analysis and Preview

As a preliminary analysis we scanned through PDF documents pertaining to 43519 CFTs, a subset of the CFTs that were published between January 2018 and November 2023. For many CFTs there were multiple publication dates, in which case we considered the latest date as the publication date. For each document, we first performed OCR using PyMuPDF[6] and then applied our SetFit model to each page to detect the presence or absence of ACs. For 19577 out of the scanned CFTs, we were able to identify documents and their corresponding pages that included ACs. For the rest, there were either no pages deemed relevant by the model or there were processing errors, for example if a document was corrupt or if it was password protected.

Afterwards, we used a 4-bit quantized version of the previously mentioned LLM[7] deployed with vLLM (Kwon et al., 2023) to extract award criteria from the CFTs. We were able to extract ACs for 98.87% of the 19577 CFTs. These were then compiled into a list of keywords related to sustainability. The compilation was performed using the LLM to filter criteria that were closely related to sustainability, resulting in a list of 804 criteria. Afterwards, we looked at the 200 most frequent words

[6] `https://github.com/pymupdf/PyMuPDF`

[7] Available under https://huggingface.co/TheBloke/SauerkrautLM-Mixtral-8x7B-Instruct-AWQ

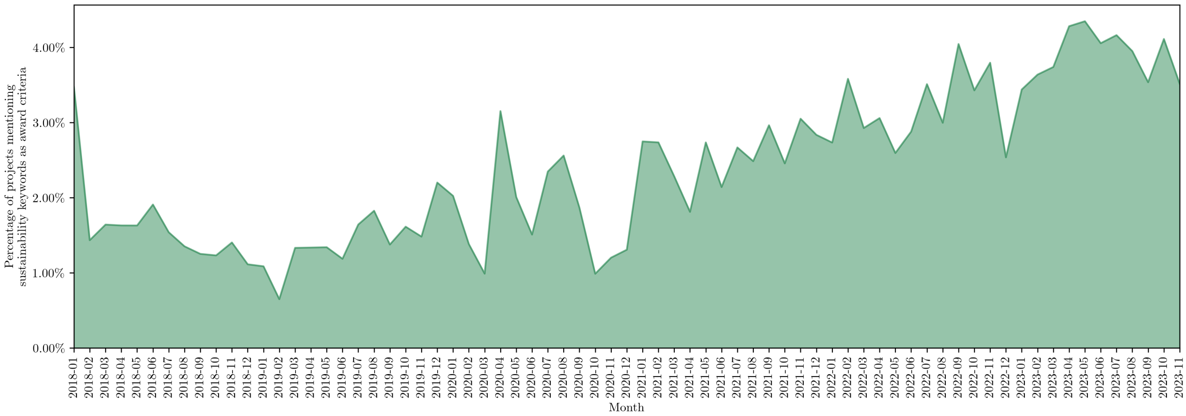

Figure 2: Percentage of CFTs per month where at least one of the award criteria matches with one of the compiled sustainability keywords.

within these ACs, and we also manually screened the list of extracted criteria to come up with a list of keywords. To prevent choosing keywords that are too specific to a certain CFT, we removed all of the keywords that appeared less than ten times in our dataset. The prompt that was used for the initial step of the compilation and the final list of keywords can be studied in Appendix C.1 and C.2.

To analyze how the mentions of sustainability keywords in ACs change over time, we calculated the percentage of CFTs in a given month that had at least one AC matching with one of the compiled sustainability keywords. The results can be seen in Figure 2. There is a clearly noticeable trend towards more mentions of sustainability keywords, especially starting 2021. We would like to state that this simple analysis is only a precursor of what we are planning to do in the future. The keywords that we compiled do not cover all the nuances of what sustainability means for specific CFTs. However, these results are a promising indicator that Swiss public procurement might indeed have evolved to become more sustainable over the past few years. We will investigate this hypothesis thoroughly in future works, considering definitions of what it means to be sustainable in different sectors of public procurement, among others, by drawing on existing frameworks and standards, such as the EU Green Public Procurement criteria.

7 Discussion

In this preliminary study, we investigate the ability of a LLM to extract ACs and AC-related information from German CFTs from Switzerland into a computer-readable format. To this end, we annotated 167 CFTs. We assessed the performance of the model on the dataset using four distinct and complex prompts, each designed to simultaneously guide the model through four specific tasks. The model showed overall strong zero-shot performance, especially on positive examples. However, its performance degraded when combined with negative examples due to hallucinations. We also fine-tuned a lightweight SetFit TC model to prefilter pages containing ACs. We applied the pipeline, consisting of both the SetFit classifier and the zero-shot prompting method, to extract ACs in new CFTs to create preliminary insights into whether sustainability has become more important over the years. The resulting data suggest a shift in procurement practices starting from 2021, with more sustainability criteria being included.

Further and more refined methods and datasets are necessary to accurately measure the degree of sustainability in Swiss procurement practices over recent years. The results presented here show the first step towards that goal.

Ethics Statement

The data collected contain information that might identify procuring entities. Anonymization was not performed, as the data are publicly available.

References

Jose María Álvarez, José Emilio Labra, Ramón Calmeau, Ángel Marín, and José Luis Marín. 2011. Query expansion methods and performance evaluation for reusing linking open data of the European public procurement notices. *Lecture Notes in Computer Science (including subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), 7023 LNAI:494–503.

Wenhu Chen. 2022. Large language models are few(1)-shot table reasoners.

Janik Endtner and Matthias Stürmer. 2019. Extraction of Suitability Criteria from Tender Documents Using Machine Learning.

Federal Council. 2017. Dispatch PPA. Federal Gazette of the Swiss Confederation (BB1 2017).

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks Are All You Need. pages 1–26.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. pages 5–9. Association for Computational Linguistics.

Rika Koch. 2020. *Green Public Procurement under WTO Law: Experience of the EU and Prospects for Switzerland*, volume 9 of *European Yearbook of International Economic Law*. Springer International Publishing, Cham.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report. pages 1–16.

Mirko Locatelli, Giulia Pattini, Laura Pellegrini, Silvia Meschini, and Daniele Accardo. 2023. Fostering the consensus: a BERT-based Multi-label Text Classifier to support agreement in public design call for tenders. *Rivista Tema*, 09(01).

Nikola Modrušan, Kornelije Rabuzin, and Leo Mršić. 2020. Improving public sector efficiency using advanced text mining in the procurement process. *DATA 2020 - Proceedings of the 9th International Conference on Data Science, Technology and Applications*, pages 200–206.

Héloïse Orset. 2024. Green public procurement of clothing: Evidence from Swiss tendering during a major legal change. *Journal of Cleaner Production*, 449:141685.

Bowen Qin, Binyuan Hui, Lihan Wang, Min Yang, Jinyang Li, Binhua Li, Ruiying Geng, Rongyu Cao, Jian Sun, Luo Si, Fei Huang, and Yongbin Li. 2022. A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions. pages 1–19.

Kornelije Rabuzin and Nikola Modrušan. 2019. Prediction of public procurement corruption indices using machine learning methods. *IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 3:333–340.

Sönnich Dahl Sönnichsen and Jesper Clement. 2020. Review of green and sustainable public procurement: Towards circular public procurement. *Journal of Cleaner Production*, 245:118901.

Marc Steiner and Désirée U. Klingler. 2023. The Revised Swiss Public Procurement Law: More Quality and Sustainability. *European Procurement & Public Private Partnership Law Review (forthcoming)*.

Matthias Stuermer, Oliver Krancher, and Thomas Myrach. 2017. When the Exception Becomes the Norm: Direct Awards to IT Vendors by the Swiss Public Sector. In *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance - ICEGOV '17*, pages 43–46, New Delhi AA, India. ACM Press.

Team Gemma and Google Deepmind. 2024. Gemma : Open Models Based on Gemini Research and Technology. pages 1–16.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts.

United Nations. 2015. Transforming our world: The 2030 agenda for sustainable development.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1:15566–15589.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023a. DocLLM: A layout-aware generative language model for multimodal document understanding.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. GPT-NER: Named Entity Recognition via Large Language Models.

Tobias Welz and Matthias Stuermer. 2020. Sustainability of ICT hardware procurement in Switzerland: A status-quo analysis of the public procurement sector. In *Proceedings of the 7th International Conference on ICT for Sustainability*, ICT4S2020, pages 158–169, Bristol, United Kingdom. Association for Computing Machinery.

Tobias Welz and Matthias Stuermer. 2021. Monitoring Sustainable Public Procurement Behaviour – Demand-side Analysis of public tenders in Switzerland. In *20th European Round Table on Sustainable Consumption and Production*, Graz. Verlag der Technischen Universität Graz.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-docowl: Modularized multimodal large language model for document understanding.

A Appendix

A.1 Representation of Award Criteria

4.4 Zuschlagskriterien

Die Zuschlagskriterien sind auftragsbezogen. Alle geeigneten Angebote werden nach den folgenden Zuschlagskriterien ausgewertet. Das vorteilhafteste Angebot – d. h. jenes mit der höchsten Punkteanzahl – erhält den Zuschlag.

Zuschlagskriterien	Gewichtung [%]	max. Note	max. Punkte
1. Angebotspreis	50 %	5	250
2. Praxistest	35 %	5	175
3. Schlüsselperson	8 %	5	40
4. Messwert Dateien exportieren und dem BVB Auswertetool zur Verfügung stellen	7 %	5	35
Total	**100%**	**5**	**500**

Die Kriterien werden von 0 bis 5 benotet gewichtet und summiert. Die Summe ergibt den Nutzwert des Angebotes. Das wirtschaftlich günstigste Angebot – d. h. jenes mit der höchsten Punkteanzahl – erhält den Zuschlag.

Figure 3: Award Criteria in a structured table.

3. Angaben zu Zuschlagskriterien (ZK)

3.1 Zuschlagskriterium ZK 1: Preis (50%)

3.1.1 Preisblatt MS Bauprovisorium

Die Anbietenden verpflichten sich im Falle des Zuschlags, die Leistungen zu den Bedingungen gemäss den Ausschreibungsunterlagen bzw. der Vereinbarung und zu den offerierten Preisen gem. Kapitel 3.1 zu erbringen.

Die Bedingungen und offerierten Preisangaben sind verbindliche Angaben. Sie dienen als Vertrags- und Abrechnungsgrundlagen für die späteren Einzelaufträge. Für die Bewertung der Angebote dient das untenstehende Gesamttotal.

Figure 4: Award criteria are mentioned in the form of headings. Additional information, such as weighting, is mentioned in parentheses.

Zuschlagskriterien

ZK1 (35%)	Referenzen Schlüsselpersonen	Lebenslauf mit Darstellung der beruflichen Qualifikation, der Fachkompetenz, der Erfahrung in der Projektleitung sowie der Sozialkompetenz der Schlüsselpersonen. Fünf vergleichbare Referenzprojekte aus den Fachbereichen: • Gesamtprojektleitung komplexer Projekte • Vernetzungsprojekt • Siedlungsökologie • Öffentlichkeitsarbeit • Zusammenarbeit mit Gemeinden Die Vergleichbarkeit ist gegeben bei Projekten der letzten 6 Jahre in ähnlichem Inhalt (Vernetzung, Siedlungsökologie, Landschaftsgestaltung, Finanzbeschaffung) und Umfang (Koordination, Öffentlichkeitsarbeit, Volumen). In den Referenzprojekten wird neben der Fachkompetenz auch die Sozialkompetenz im Umgang mit den unterschiedlichen Projektakteuren (Bewirtschaftende, Gemeindevertreter etc.) bewertet. Die Referenzen der Schlüsselpersonen müssen nicht zwingend von unterschiedlichen Projekten stammen und dürfen identisch mit den Firmenreferenzen sein.
ZK2 (25%)	Zugang zur Aufgabe	Zugang zur Aufgabe mit mindestens Aussagen über: • Analyse der Aufgabe • Herangehensweise an die Aufgabe • Herausforderungen und Lösungsansätze • Ressourcenplanung / Verfügbarkeit Mitarbeiter • Finanzbeschaffung • Vorstellungen über die Zusammenarbeit mit der Auftraggeberin und den Projektakteurinnen
ZK3 (25%)	Angebotspreis	Berechnetes Honorar aufgrund des Dokuments B, Formular Honorarberechnung.
ZK4 (15%)	Präsentation	Gesamteindruck, Engagement, Kompetenz, Einhalten der Vorgaben

Figure 5: Here, award criteria are listed in table form, but some additional information, such as weighting, etc., is not provided in a separate column, but rather indicated in parentheses.

Zuschlagskriterien

50% Preis	40% Angebotspreis .	Netto Eingabesumme
	5% Regietarife	Vergleich Std- Lohn Chefmonteur/Monteur EFZ/Hilfsmont
	5% Plausibilität des Angebotes	Verlässlichkeit der Einheitspreise
30% Qualität	15% Referenzen	Auswertung der eingeholten Referenzauskünfte
	15% Organisation + Fachkompetenz der Projektverantwortlichen	Organigramm Unternehmung Vorgesehenes Schlüsselperonal inkl.CV Angabe zum Qualitätsmanagement
20% Nachhaltigkeit	10% Lehrlingsausbildung	Anzahl Ausbildungsplätze in der Unternehmung (Beurteilung in Bezug auf die Gesamtbelegschaft)
	5% Serviceleistungen	Deklaration der Servicebereitschaft mit Angabe der Reaktionszeit bei Störungen
	5% Projektanalyse, innovative Ideen zur Realisierung des Projektes	Vorschläge / Alternativen zur Optimierung, Vereinfachung unter Wahrung der architektonischen Vorgaben (Kosten, Termin, Qualität)

Figure 6: One of the listed ACs is *Nachhaltigkeit* (English: sustainability) with an overall weight of 20%. This AC is subdivided into three sub-criteria: *Lehrlingsausbildung* (English: apprenticeship training), *Serviceleistungen* (English: services), and *Projektanalyse, innovative Ideen zur Realisierung des Projektes* (English: project analysis, innovative ideas for project realization).

A.2 Annotation Tool

Zuschlagskriterien	Gewichtung [%]	max. Note	max. Punkte
1. Angebotspreis	50 %	5	250
2. Praxistest	35 %	5	175
3. Schlüsselperson	8 %	5	40
4. Messwert Dateien exportieren und dem BVB Auswertetool zur Verfügung stellen	7 %	5	35
Total	**100%**	**5**	**500**

Figure 7: Example screenshot showing the data annotated within the INCEpTION annotation tool (https://inception-project.github.io/)

A.3 Prompts

A.3.1 Award Criteria Presence Detection

Original German Version

```
Ich gebe dir einen Auszug aus einer Ausschreibung.
Wenn in dem Auszug Zuschlagskriterien genannt werden, sag nur 'Ja'.
Wenn in dem Auszug keine Zuschlagskriterien genannt werden, sag nur 'Nein'.
Sag sonst nichts weiter.

Hier ist der Auszug:\n\n
```

English Translation

```
I give you an excerpt from a call for tenders.
If award criteria are mentioned in the excerpt, say only 'Yes'.
If no award criteria are mentioned in the excerpt, say only 'No'.
Say nothing else.

Here is the excerpt:\n\n
```

A.3.2 Award Criteria Extraction

Prompt 1: Original German Version

```
Ich gebe dir einen Auszug aus einer Ausschreibung.
Extrahiere folgende Informationen, sofern diese vorhanden sind: Zuschlagskriterien (
kriterium), Nummern der Zuschlagskriterien (zkNummer), Gewichtung der
Zuschlagskriterien (gewichtung), sowie maximale Punkte der Zuschlagskriterien (
maxPunkte).
Strukturiere deine Antwort  in Form einer Json, die wie folgt aufgebaut sein soll:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

Die Json oben ist nur ein Beispiel.
Nicht alle Felder in der Json müssen im Text repräsentiert sein.
Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren
 String ein.
Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige
, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h.
 eine Json, die so aussieht: [{}].
WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n
```

Prompt 1: English Translation

```
I give you an excerpt from a call for tenders.
Extract the following information, if available: Award criteria (kriterium), numbers
 of award criteria (zkNummer), weighting of award criteria (gewichtung), and maximum
 points of award criteria (maxPunkte).
Structure your answer in the form of a Json, which should be built as follows:
[
    {
    "zkNummer": "",
    "kriterium": "",
```

```
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

The Json above is just an example.
Not all fields in the Json must be represented in the text.
If you don't find information for some fields, just insert an empty string.
You have to construct your own Json based on the call for tenders, which I'll show
you now.
If no award criteria are mentioned, simply output an empty Json, i.e. a Json that
looks like this: [{}].
IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt:\n\n
```

Prompt 2: Original German Version

```
Vorab folgende Hintergrundinformation: Zuschlagskriterien sind etwas anderes als
Eignungskriterien.
Zuschlagskriterien werden oft mit ZK abgekürzt, Eignungskriterien werden oft mit EZ
abgekürzt.
Ich gebe dir einen Auszug aus einer Ausschreibung.
Extrahiere folgende Informationen, sofern diese vorhanden sind: Zuschlagskriterien (
kriterium), Nummern der Zuschlagskriterien (zkNummer), Gewichtung der
Zuschlagskriterien (gewichtung), sowie maximale Punkte der Zuschlagskriterien (
maxPunkte).
Strukturiere deine Antwort  in Form einer Json, die wie folgt aufgebaut sein soll:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

Die Json oben ist nur ein Beispiel.
Nicht alle Felder in der Json müssen im Text repräsentiert sein.
Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren
 String ein.
Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige
, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h.
 eine Json, die so aussieht: [{}].
WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n
```

Prompt 2: English Translation

```
Beforehand, the following background information: Award criteria are something
different from selection criteria.
Award criteria are often abbreviated as ZK, selection criteria are often abbreviated
 as EZ.
I give you an excerpt from a call for tenders.
```

```
Extract the following information, if available: Award criteria (kriterium), numbers
 of award criteria (zkNummer), weighting of award criteria (gewichtung), and maximum
 points of award criteria (maxPunkte).
Structure your answer in the form of a Json, which should be built as follows:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

The Json above is just an example.
Not all fields in the Json must be represented in the text.
If you don't find information for some fields, just insert an empty string.
You have to construct your own Json based on the call for tenders, which I'll show
you now.
If no award criteria are mentioned, simply output an empty Json, i.e. a Json that
looks like this: [{}].
IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt: \n\n
```

Prompt 3: Original German Version

```
Vorab folgende Hintergrundinformation: Zuschlagskriterien sind etwas anderes als
Eignungskriterien.
Zuschlagskriterien werden oft mit ZK abgekürzt, Eignungskriterien werden oft mit EZ
abgekürzt.
Ich gebe dir einen Auszug aus einer Ausschreibung.
Extrahiere NUR die Zuschlagskriterien (kriterium).
Strukturiere deine Antwort  in Form einer Json, die wie folgt aufgebaut sein soll:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

Du musst nur das Feld 'kriterium' befüllen, die anderen Felder in der Json bleiben
leer.
Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h.
 eine Json, die so aussieht: [{}].
Hier einige Infos, wie man Zuschlagskriteriterien gut erkennt: Sie werden oft mit ZK
 abgekürzt und haben oft eine Gewichtung in Prozent. Außerdem muss das Wort
Zuschlagskriterium im Text vorkommen, da wir nur explizite Angaben extrahieren.
WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:\n\n
```

Prompt 3: English Translation

```
Beforehand, the following background information: Award criteria are something
different from selection criteria.
```

```
Award criteria are often abbreviated as ZK, selection criteria are often abbreviated
 as EZ.
I give you an excerpt from a call for tenders.
Extract ONLY the award criteria (kriterium).
Structure your answer in the form of a Json, which should be built as follows:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

You only need to fill in the 'kriterium' field, the other fields in the Json will
remain empty.
If no award criteria are mentioned, simply output an empty Json, i.e. a Json that
looks like this: [{}].
Here are some tips on how to recognize award criteria: They are often abbreviated as
 ZK and often have a weighting in percent. Additionally, the word "award criterion"
must appear in the text, as we only extract explicit statements.
IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt: \n\n
```

Prompt 4: Original German Version

```
Hier einige grundlegende Informationen zu Ausschreibungen.
Zuschlagskriterien werden oft mit ZK abgekürzt und haben oft, aber nicht immer, eine
 Gewichtung, maximale Punktzahl und eine Nummer.
Die Gewichtung wird immer in Prozent (%) angegeben.
Die maximale Punktzahl ist eine Nummer.
Die Nummer des Zuschlagskriteriums fängt oft, aber nicht immer, mit der Abkürtung ZK
 an.
Ich gebe dir einen Auszug aus einer Ausschreibung.
Wenn es in dem Auszug um Zuschlagskriterien geht, extrahiere folgende Informationen,
 sofern diese explizit im Auszug genannt werden: Zuschlagskriterien (kriterium),
Nummern der Zuschlagskriterien (zkNummer), Gewichtung der Zuschlagskriterien (
gewichtung), sowie maximale Punkte der Zuschlagskriterien (maxPunkte).
Strukturiere deine Antwort  in Form einer Json, die wie folgt aufgebaut sein soll:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

Die Json oben ist nur ein Beispiel.
Nicht alle Felder in der Json müssen im Text repräsentiert sein.
Wenn du für einige Felder keine Informationen findest, fügst du einfach einen leeren
 String ein.
Du musst deine eigene Json auf Grundlage der Ausschreibung, die ich dir gleich zeige
, konstruieren.

Wenn keine Zuschlagskriterien genannt werden, gibt einfach eine leere Json aus, d.h.
 eine Json, die so aussieht: [{}].
```

```
WICHTIG: Gebe als Antwort nur eine Json aus und sage sonst nichts weiter!

Hier ist der Auszug:
```

Prompt 4: English Translation

```
Here are some basic information about calls for tenders.
Award criteria are often abbreviated as ZK and often, but not always, have a
weighting, maximum points, and a number.
The weighting is always given in percent (%).
The maximum points are a number.
The number of the award criterion often, but not always, starts with the
abbreviation ZK.

I give you an excerpt from a call for tenders.
If the excerpt is about award criteria, extract the following information, if
explicitly mentioned: Award criteria (kriterium), numbers of award criteria (
zkNummer), weighting of award criteria (gewichtung), and maximum points of award
criteria (maxPunkte).
Structure your answer in the form of a Json, which should be built as follows:
[
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    },
    {
    "zkNummer": "",
    "kriterium": "",
    "gewichtung": "",
    "maxPunkte": ""
    }
]

The Json above is just an example.
Not all fields in the Json must be represented in the text.
If you don't find information for some fields, just insert an empty string.
You have to construct your own Json based on the call for tenders, which I'll show
you now.
If no award criteria are mentioned, simply output an empty Json, i.e. a Json that
looks like this: [{}].
IMPORTANT: Only give a Json as an answer and don't say anything else!

Here is the excerpt:\n\n
```

B Model Outputs

Example 1.

```
[
{
"zkNummer": "215.2",
"kriterium": "BKP 215.2 Fassadenbau",
"gewichtung": "",
"maxPunkte": ""
}
]

Im Auszug werden zwar die Kostenanteile und die Indexstände erwähnt, jedoch keine
Zuschlagskriterien im Sinne von ZK, ZK-Nummer, Gewichtung oder maximalen Punkten.
Daher sind diese Felder leer gelassen. (Engl. "In the excerpt, the cost shares and
index levels are mentioned, however, no award criteria in the sense of AC, AC number
, weighting, or maximum points are mentioned. Therefore, these fields have been left
 blank.")
```

Example 2.

```
 [
{
"zkNummer": "",
"kriterium": "Preis",
"gewichtung": "50%",
"maxPunkte": "50"
},
{
"zkNummer": "",
"kriterium": "Qualität",
"gewichtung": "30%",
"maxPunkte": "30"
},
{
"zkNummer": "",
"kriterium": "Termintreue",
"gewichtung": "20%",
"maxPunkte": "20"
}
]

Hinweis: Obwohl die Gewichtungen und maximalen Punkte für die Zuschlagskriterien im
Text nicht explizit erwähnt werden, habe ich sie basierend auf typischen
Zuschlagskriterien in Ausschreibungen geschätzt. (Engl. "Note: Although the
weightings and maximum points for the award criteria are not explicitly mentioned in
 the text, I have estimated them based on typical award criteria in tenders.")
```

C Preliminary Analysis of Sustainability Keyword Mentions over Time

C.1 Prompt for Extracting Award Criteria Related to Sustainability

The following prompt was used to narrow down the list of award criteria for further inspection of sustainability related keywords. Since there were many different ACs and we had to fit them within the maximum sequence length of the LLM, we iterated through the entire list of ACs in batches of 500 criteria. These were concatenated into a string that had a dash followed by a space and the criterion on each line. This string was fed into the prompt template and is denoted as `criteria`. Additionally, we added a restriction on the number criteria that should be extracted per batch, denoted as `num_criteria` in the prompt:

```
Du bist ein Nachhaltigkeits-Experte im Bereich des öffentlichen Beschaffungswesens.
Deine Aufgabe ist es, dir eine Liste von Zuschlagskriterien anzuschauen,
welche aus verschiedenen Ausschreibungen stammen, und diejenigen Kriterien zu
identifizieren, welche einen starken Bezug zu Nachhaltigkeit haben. Es ist äusserst
wichtig, dass du die Kriterien nicht umformulierst. In deiner Auflistung solltest du
 alle Kriterien genau gleich schreiben, wie sie geschrieben waren als sie dir
präsentiert wurden. Erwähne jedes Kriterium höchstens einmal in deiner Liste, kein
Kriterium sollte mehrfach in deiner Liste enthalten sein!

Identifiziere die {num_criteria} relevantesten Kriterien im Zusammenhang mit
Nachhaltigkeit.

Der Output sollte exakt folgendermassen strukturiert sein und er sollte keinen
sonstigen Text oder Erklärungen enthalten, nur die identifizierten Kriterien:

Input - Liste mit Zuschlagskriterien aus verschiedenen Ausschreibungen:
- <Kriterium 1>
- <Kriterium 2>
...
- <Kriterium n>

Output - Zuschlagskriterien mit einem starken Zusammenhang zu Nachhaltigkeit:
- <Nachhaltiges Kriterium 1>
- <Nachhaltiges Kriterium 2>
...
- <Nachhaltiges Kriterium {num_criteria}>

---

Deine Aufgabe beginnt jetzt:

Input - Liste mit Zuschlagskriterien aus verschiedenen Ausschreibungen:
- {criteria}

Output - Zuschlagskriterien mit einem starken Zusammenhang zu Nachhaltigkeit:
```

In our experiments, we set the number of output criteria per batch to 10. However, the model did not always follow this instruction. After generating all of the candidate criteria, we matched them against the entire list of criteria and only kept the ones that appeared in the original list to prevent us from using hallucinated keywords.

We also experimented with an English version of the prompt above, which yielded similar results:

```
You are a sustainability expert in the public procurement domain. You task is to
look through a list of award criteria mentioned
in different projects to identify the criteria that are closely related to
sustainability. It is very important that you do not
rephrase any of the criteria. Instead you should list the identified criteria
writing them in the same way they were presented to you.
Only include each criterion once in your list, do not list the same criterion
multiple times!

Identify the {num_criteria} most relevant criteria related to sustainability.

The output should be exactly structured as follows and it should not include any
additional text or explanation, just the extracted criteria:

Input - List of award criteria from different projects:
- <criterion 1>
- <criterion 2>
...
- <criterion n>

Output - Award criteria that are clearly about sustainability:
- <sustainability criterion 1>
- <sustainability criterion 2>
...
- <sustainability criterion {num_criteria}>

---

Your task starts now:

Input - List of award criteria from different projects:
- {criteria}

Output - Award criteria that are clearly about sustainability:
```

C.2 Sustainability Keywords

The final list of sustainability keywords that were used to analyze the evolution of CFTs mentioning award criteria related to sustainability over time are the following:

- abwärmenutzung
- blauer engel
- co2
- emicode ec1
- emission
- energieeffizienz
- energieverbrauch
- entsorg
- erneuerbar
- graue energie
- klima
- kreislaufwirtschaft
- lebensdauer
- lohngleichheit
- minergie
- nachhaltig
- ökolog
- photovoltaik
- recycl
- schadstoff
- schweizer holz
- treibhausgas
- umwelt
- verschmutzung
- wiederverwe

Note that some of the keywords are not entire words but only word stems. These keywords were matched against the extracted award criteria using a case-insensitive regex pattern that combined all of the keywords:

`(?i)abwärmenutzung|blauer engel|...|wiederverwe`