Evaluating Large Language Models and Prompt Variants on the Task of Detecting Cease and Desist Violations in German Online Product Descriptions

Marian Lambert[1][0000−0001−7347−5734], Nico Döring[1][0009−0004−2018−5417], Thomas Schuster[2][0000−0002−9539−1627], and Norbert Schmitz[2][0000−0002−6683−4564]

[1] XPACE GmbH, Blücherstr. 32, 75177, Pforzheim, Germany
[2] Pforzheim University, Tiefenbronner Str. 65, 75175 Pforzheim, Germany

Abstract. This paper compares different large language models (LLMs) on the task of detecting violations of cease and desist declarations in German online product descriptions as part of the KIVEDU project. We evaluate two proprietary LLMs by OpenAI (`gpt-3.5-turbo-0301` and `gpt-3.5-turbo-0613`) and three open source LLMs by various organizations (`LLaMA2`, `StableBeluga2`, and `Platypus2`) using different variations in text input (prompt) on a dataset of 116 manually labeled pairs of cease and desist declarations and product descriptions. The evaluation aims to explore two research questions: 1) Which LLM is most adept at identifying violations? and 2) How do prompt variations impact model performance? The results show that `StableBeluga2` performed best, achieving the highest accuracy and micro F1 score. It was also the most reliable model with minimal deviations in performance across prompt variants. The `Platypus2` and `gpt-3.5-turbo-0301` models also achieved good results though they displayed greater variability in their performance. The worst-performing model was `LLaMA2`. The results further show that prompting had a significant impact on model performance, with the presence of a step-by-step instruction generally decreasing performance and a "yes"/"no" output format leading to higher performance. However, this was highly dependent on the specific model. Role prompting and providing a longer vs. shorter instruction had a minimal impact on performance across models. Overall, the study demonstrates the potential of LLMs in automating the detection of cease and desist violations in online product descriptions. Further research is needed to evaluate other LLM models and prompt variations, as well as to explore approaches like LLM fine-tuning on domain-specific data to further improve performance.

Keywords: NLP · Large Language Model · Consumer Protection · Consumer Rights Enforcement · Prompt Engineering · LegalTech

1 Introduction

The rapid digitization of commerce has led to an increase in the number of products and services available online. As consumers increasingly shift to e-commerce platforms for their purchasing needs, the volume of online offerings also continues to rise. However, rising alongside the digital growth is the incidence of consumer-rights infringing behaviors, notably in the form of misleading or false product descriptions and deceptive advertising claims.

A study conducted in 2014 emphasized this concern [8]: approximately 37% of the European Union's e-commerce activities were found non-compliant with the Union consumer law. This deviation from legal standards has wrought an estimated annual financial detriment of €770 million upon consumers. While this poses evident financial ramifications, it further exacerbates the competitive market dynamics, putting compliant businesses at an unfair disadvantage.

To counteract these activities in Germany, consumer protection agencies (and other entitled organizations) can request infringing companies to sign a *cease and desist declaration*. This legal document binds the signing company to abstain from involving in similar unfair practices in the future. Nonetheless, ensuring compliance with these declarations, i.e. verifying that the company does not participate in similar infringing behavior, remains a significant challenge. Consumer protection agencies, grappling with constraints in resources, find it burdensome to conduct routine checks for adherence. The manual nature of these verification processes renders them not only time-intensive but also financially costly.

This difficulty in monitoring compliance often provides companies with an opportunity to persist in their unfair practices, infringing upon consumer rights without significant checks. Such unchecked behavior not only endangers consumers but also undermines fair competition in the market, creating an unfavorable environment for competitors who operate ethically.

Recent strides in the realm of natural language processing (NLP), particularly the advent of large language models (LLMs), offer a promising avenue to address this challenge. The capabilities exhibited by these models pave the way for automating the otherwise manual processes, thus offering support in consumer rights enforcement. In a recent publication, we introduced the KIVEDU project which aspires to revolutionize the enforcement of consumer rights in Germany through the use of LLMs and other AI technologies [24].[3] In its current, initial phase, the project focuses on automating the identification of cease and desist violations in German product descriptions.

This technical research paper delves into our preliminary assessment of implementing various LLMs, encompassing both proprietary and open source models, in conjunction with multiple text input (prompt) variations on a manually curated test dataset. At its core, our exploration is guided by two central research questions:

[3] Project website can be found online at `https://www.kivedu-projekt.de`

RQ1: Which of the tested LLMs emerges as the most adept for the specific task of identifying cease and desist violations in German product descriptions?
RQ2: How do variations in the provided prompt influence the overall performance of the LLMs on this task?

The remaining document is structured as follows: In Section 2, we present a short review of related works. This will be followed by a section introducing the theoretical underpinnings, primarily introducing LLMs and their application potential for the challenge at hand. Section 4 outlines our methodology using a zero-shot prompting approach. Further, the experimental framework is presented in Section 5, including details of the tested models, dataset and prompt variations. We then present and discuss our results in Section 6. Lastly, Section 7 gives a conclusive summary and projections for future avenues of research.

2 Related Works

In this section, we present related publications and projects that use AI and Machine Learning (ML) in the area of consumer protection. We conclude this section with a short paragraph on how our problem description differs from the ones presented here.

For years, researchers have studied ML applications in consumer protection law, initially focusing on Terms of Service (ToS) and privacy contracts. A pivotal study by Lippi et al. used ML techniques like support vector machines (SVM) and tree kernels to identify unfair clauses in online contracts, setting a new standard in performance beyond random guessing [12].

Braun and his team (2019) described the potential of legal tech in strengthening consumer rights, showcasing two models that interpret, assess, and summarize the terms of service of German online retailers [3]. They also delved into automated recognition of unlawful sections within German terms and conditions (TaC) [2]. Utilizing a pre-trained German BERT language model [7], they were able to pinpoint illegal content with high accuracy. Based on these results, the CLAUDETTE tool was created. This tool not only enlarges the base dataset to cover 50 ToS, but also employs various ML techniques, including Deep Learning, for a detailed categorization into specific classes of unfairness [13]. This tool has also been used to identify confusing or unfair privacy terms related to the European General Data Protection Regulation [6].

Besides identifying biased or illicit terms in online default agreements, ML has also been utilized by Trappey et al. to analyze and pinpoint judgment documents from US trademark dispute precedent cases [22]. Furthermore, in the realm of patent law, ML has been employed to forecast legal conflicts [14, 10].

Moreover, in 2018, Chakrabarti and colleagues designed an ML-driven system to detect and extract high-risk sections within contracts and then categorize them into different risk levels [4]. They utilized paragraph vectors for training and applied classification methods encompassing multiple versions of SVM and Naive Bayes models, attaining high accuracy rates.

Generally, it can be observed that there is limited research on applications of LLMs towards tasks in the legal domain. To the best of our knowledge, no published work exists for a use case similar to that presented in this paper. Therefore, our project distinguishes itself from the publications presented above in various ways. Firstly, we don't address typical contracts like ToS or privacy policies. Our focus is on cease and desist declarations, which are civil law agreements between businesses and consumer protection agencies or other qualified organizations. These declarations often have unique and highly specific characteristics, presenting obstacles for machine learning training and analysis. Moreover, determining a legal violation is not just dependent on the wording of the cease and desist declaration, but also on the related online product description that may contain the infringement. As a result, we can't label whole sections as unjust, in contrast to the methods used in the cited publications. A comprehensive individual review of the cease and desist declaration, considering both its specific wording and the surrounding context, is essential. For these reasons, and contrary to the cited literature, we are utilising current state-of-the-art LLMs for our specific use case.

3 Theoretical Background

In this section, we introduce the most important theoretical concepts of our research. We begin by explaining the purpose of cease and desist declarations in Germany and the mechanisms for their verification. Subsequently, we explore the challenges of automating the detection of violations in these declarations and discuss the limitations of conventional ML models for this task. We then introduce LLMs, explain how they are trained and emphasize their suitability for addressing this challenge. Finally, we show how LLMs can be applied to our specific problem.

3.1 Cease and Desist Declarations in Germany

Consumer protection organizations (as well as business associations, competitors and the chamber of industry and commerce) in Germany can issue a cease and desist declaration to offending companies when they violate consumer rights by advertising with inaccurate or misleading online product descriptions. This statement, which identifies the precise infringement (for instance, claiming that a food supplement can treat cancer), acts as a legally binding commitment of the company to refrain from similar conduct in the future. If it signs the declaration, the corporation may be subject to fines for repeated infractions. To make sure a company is adhering, consumer protection organizations must continuously check public product descriptions to make sure they don't violate any agreements the company has signed. AI can be used to speed up this time-consuming task, which frequently entails reading through thousands of product descriptions and hundreds of cease and desist declarations. Monitoring can be automated by an AI model that assesses if a violation is present.

Yet, automatically detecting violations is a nuanced task demanding intricate language understanding and good reasoning capabilities. Slight variations in the phrasing or semantic meaning of the declaration or the product description can be pivotal in determining a potential violation. For instance, a firm might be prohibited from promoting a banking account as "the *first* CO2 neutral account", but a statement like "open *your first* CO2 neutral account" might be compliant. Thus, the ability to distinguish subtle differences in linguistic constructions and semantics is highly important.

3.2 Challenges

Traditional machine learning techniques, which were successful in many of the examples we discussed above, face difficulties with this task. They have trouble understanding language and place an excessive reliance on large amounts of training data.

Data Dependency The vast majority of machine learning approaches are heavily reliant on extensive datasets to train effectively. In the context of cease and desist declaration violations, obtaining a vast number of relevant and varied examples is inherently challenging given the specificity and infrequency of such cases, particularly in our situation where we focused on a specific area (online product descriptions) and language (German).

Nuanced Language Understanding Existing models may struggle with comprehending the nuanced differences in terms and phrases, leading to potential oversight in intricate cases. This is due to the fact that they do not have a deeper understanding of the (German) language and thus will fail to identify words or phrases which are semantically similar but did not occur in the training data.

3.3 Large Language Models

Given these limitations of traditional ML approaches, large language models present a promising innovation in the realm of Deep Learning. Large language models refers to a family of neural network models (often based on the *transformer* architecture [23]) which are designed for processing and generating human language. LLMs are pre-trained on massive text data from the internet [26]. Given a text input (prompt), these models generate text as output based on which words are most probable and helpful (text generation). LLMs are characterized by a large number of model parameters, reaching hundreds of billions. Due to their size, LLMs exhibit exceptional understanding of human language and can reliably solve a variety of downstream tasks without any specific training data, such as question answering, translating between multiple languages, classifying sentiments in a text, and summarizing long passages [26].

Large language models are typically trained following a two-tiered process (see [26]). In the initial stage, known as *pre-training*, the model processes expansive text datasets (often sourced from the internet) with the task of learning

to predict subsequent words in sentences. Through this, it familiarizes itself with grammatical structures, real-world facts, and even reasoning skills. The all-rounded model resulting from this stage is often referred to as a *base LLM*. This type of model can reliably predict the next word in a sentence and be used on a variety of tasks. However, for more complex tasks, their usability is limited as they tend to not follow instructions provided in the prompt.

To solve this limitation, the base model can further be refined in a process called *fine-tuning*, where it's trained to follow provided instructions. This often includes an approach called reinforcement learning through human feedback (RLHF) [27]. Essentially, a reward model is trained on human-provided data to determine the usefulness of input-output pairs of the base model. The base model is then fine-tuned through reinforcement learning utilizing the reward model. The result of this process is an *instruction-tuned* LLM, which can reliably follow provided instructions and provide helpful outputs.

3.4 Detecting Cease and Desist Violations with LLMs

The capabilities of LLMs are exceptional. In particular, their ability to solve multiple tasks without requiring unique model architectures and training data for each sets them apart. The concept of transfer learning, where knowledge from one domain can be repurposed for another, negates the necessity for vast labeled datasets across all applications. Given these benefits, this paper presents an evaluation of applying different LLMs on the task of identifying cease and desist violations in German product descriptions. For this use case, LLMs can be applied in two ways.

Prompting This approach hinges on text generation. The fine-tuned LLM is provided with a text input (prompt) encompassing an instruction of what to do, i.e. decide whether a violation is present or not, and the cease and desist declaration as well as product description. It then generates an output, indicating whether a violation is present or not. In the standard case of *zero-shot prompting*, the model is not provided with any specific examples of how a correct input-output pair should look like; it generates its output only from the provided instruction. In the case of *one-* or *few-shot prompting*, the model is additionally provided with one or multiple correct input-output pairs in the prompt, thus giving the model the opportunity to learn from examples.

Classification Head Fine-Tuning In this approach, a classification head is trained on the intermediate outputs of the fine-tuned LLM. This essentially constructs a specialized classification model that is fine-tuned for the specific task at hand. At the same time, it retains the extensive language understanding capabilities inherent in the LLM. While this strategy aims to yield more consistent outputs, it comes with a limitation: it requires a large set of labeled data for training. Such extensive labeled data is often scarce in specialized fields like ours, making this approach less feasible for our particular context. Therefore, our paper chooses to focus on the first strategy, namely the prompting approach, and more specifically on zero-shot prompting.

4 Methodology

Following the zero-shot prompting approach, we developed a process of deciding whether a product description violates a cease and desist declaration which consists of *prompt construction*, *inference* and *output parsing*.

Prompt Construction In this step, we construct the text input which will be passed to the model. This input encompasses a *main prompt* consisting of multiple variable components (role, instruction, step-by-step instruction and output format) and the *target* which contains the relevant passage of the cease and desist declaration and the product description (see Figure 1 for the generic prompt structure and an example prompt). The components of the main prompt are based on general prompt engineering guidelines and have been shown to improve LLM performance. In particular, the *role* component assigns a role to the LLM with the aim of inducing behavior that is more suited for solving the task at hand [19]. The *instruction* component provides the general instruction for the task. The *step-by-step instruction* provides a detailed iterative instruction for solving the task (chain of thought prompting). This has been shown to help LLMs solve more complex tasks [25]. Lastly, the *output format* component provides guidance on how the output of the LLM should look like (e.g. JSON or "yes"/"no"). The specific values we used for each component will be introduced in the next section.

Inference We pass the prompt constructed in the previous step to the LLM and capture the generated output. Depending on the expected output (provided in the output format component), we might limit the number of tokens generated by the LLM to save on inference time and cost.

Output Parsing We parse the generated output to receive a binary decision on whether a violation is present or not. This step also depends on the expected output format.

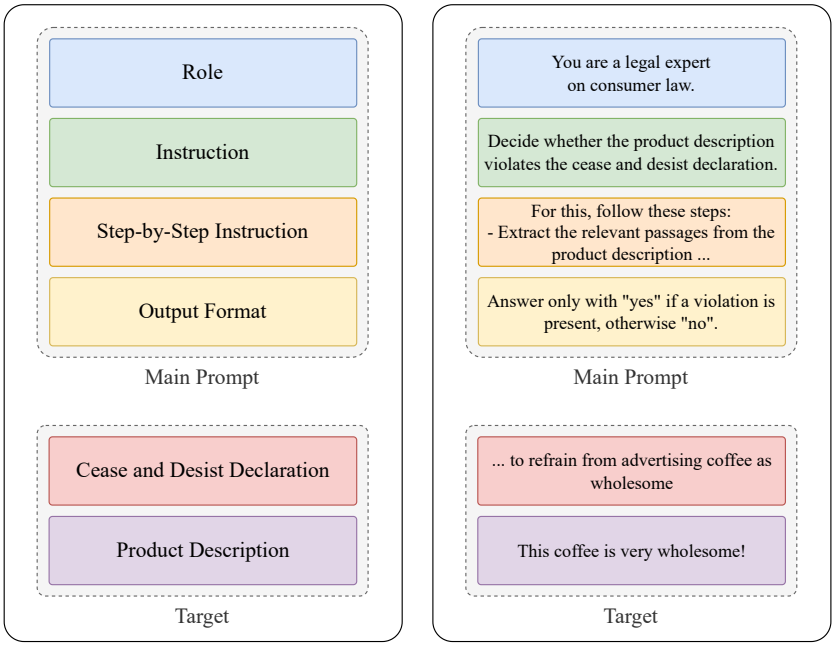
5 Experimental Setup

We evaluated five LLMs using different prompt variations on a manually curated test dataset to provide answers to the question of which LLM is best suited for identifying cease and desist violations in product descriptions (**RQ1**) and how different prompt variants impact model performance (**RQ2**). Our experimental setup consists of *five parts*: the dataset, tested LLMs, prompt variants, evaluation metrics, and evaluation process. We will now introduce each component in detail and explain our evaluation process.

5.1 Dataset

Our dataset consists of 116 manually labeled pairs of cease and desist declarations (81 unique) and product descriptions (106 unique) in German language.

Fig. 1. Prompt Structure (left) and Example Prompt (right)



In 84 cases, the product description contains a violation of the corresponding declaration. The remaining 32 cases contain no violation. The dataset spans a wide range of product types, including supplements, medication, books, food, financial services, beverages and more. All pairs are based on real-world examples, but some were modified to specifically include or not include a violation. The dataset was created and labeled by studied experts in the field of consumer law and e-commerce. Unfortunately, due to data privacy limitations, we cannot publish the dataset.

5.2 Tested LLMs

In our experiments, we evaluated two proprietary and three open source LLMs (see Table 1). They are all instruction-tuned LLMs and were released or last updated within the last six months at the time of writing. Both `gpt-3.5-turbo` models are different versions of the same underlying model (from March 1st 2023 and June 13th 2023 respectively). `StableBeluga2` and `Platypus2` are both either directly or indirectly based on `LLaMA2`. They are ranking in the top ten on the Huggingface Open LLM Leaderboard at the time of writing [1]. We chose these LLMs as they are known to perform very well, have a large enough context size of around 4000 tokens, were also trained on German text and were fairly easy and cheap to use. Other suitable, proprietary LLMs like `Claude2`, `PaLM2` or `GPT-4` were either not available to us or too expensive to evaluate systematically.

Model	Organization	Context Size	# Parameters	License
`gpt-3.5-turbo-0301` [18]	OpenAI	4096	175B	Proprietary
`gpt-3.5-turbo-0613` [18]	OpenAI	4096	175B	Proprietary
`LLaMA2` [21]	Meta	4096	70B	Open Source
`StableBeluga2` [21, 17, 15]	StabilityAi	4096	70B	Open Source
`Platypus2` [21, 9, 11]	garage-bAInd	4096	70B	Open Source

Table 1. Overview of the tested LLMs

5.3 Tested Prompt Variants

As explained in the last section, the main prompt we used consisted of different variable components (see Figure 1). To evaluate the impact of different prompt variations, we came up with different values for each component and evaluated all LLMs with each combination of components on the entire dataset (72 prompts in total). The values we used for each component are presented in Table 2, translated to English. For our experiments, we used the original German variants. The structure of the *target* part of the prompt remained constant except for the actual values of the cease and desist declaration and product description which were added based on the current case from the test dataset.

5.4 Evaluation Metrics

By comparing the parsed outputs from the model with the true labels from our dataset, we can calculate different classification metrics. On top of the traditional metrics like accuracy, F1 scores (both macro, i.e. unweighted mean of the F1 scores for each class, and micro, i.e. taking total true positives, false negatives and false positives across classes into account), precision and recall, we also calculate the *total accuracy* which counts cases where the parsing failed as incorrectly classified examples. Total accuracy is the most important metric to evaluate for us as many LLMs generated invalid outputs which can lead to misleading accuracy and F1 scores. In particular, we give special attention to accuracy and micro F1 scores. This is due to the fact that our dataset is imbalanced, and these metrics combined provide a more comprehensive perspective, ensuring that the model performs well across all classes, not just the majority ones.

5.5 Evaluation Process

We implemented all LLMs using the Python library `langchain` which provides us with a common interface to use for our evaluation process. For parameters, we adhered to each model's default settings as per its documentation. However, we minimized the temperature and limited token generation to 16 for faster inference. We designed prompt variants by computing the cartesian product of each prompt component. Each prompt variant was then systematically run through the dataset for each model. Certain models necessitated slight prompt modifications to align with their specific prompting formats.

ID	Content (translated to English from German)
	Role
none	-
expert	You are an expert for consumer protection and competition law
	Instruction
default	Your task is to find out whether the product description in the triple quotes violates the cease and desist letter in the triple quotes.
short	Check whether the product description contains a violation of the cease and desist letter.
verbose	Your task is to find out whether the product description in the triple quotes violates the cease-and-desist declaration in the triple quotes. A violation occurs if the product description contains formulations or statements that were prohibited in the cease-and-desist declaration. You must decide for yourself whether the exact wording must be present, or whether wording in the spirit is already sufficient.
default_verbose	You will receive a cease-and-desist letter that describes what action is to be refrained from and a description with which a product is advertised online. Check whether the product description contains a violation of the cease-and-desist declaration.
	Step-by-Step
none	-
default	To solve the task, perform the following steps: Step 1: Read the cease and desist letter carefully. The cease-and-desist declaration describes what behavior is to be refrained from. Step 2: Read the product description carefully and look for phrases that could constitute a violation of the cease-and-desist letter. Step 3: Extract the maximum of three most important passages from the product description and the cease-and-desist declaration. If there are no relevant passages, leave the corresponding fields blank. Step 4: Compare the extracted passages from the product description with the extracted passages from the cease and desist letter. Step 5: Write a reason why the product description does or does not violate the cease and desist letter. Step 6: Decide whether or not the product description violates the cease and desist letter.
exact_loose_match_hint	*Similar to default, but contains a hint for handling edge cases*
	Output Format
binary_numeric	Answer “1” if there is a violation of the cease and desist letter in the product description. Otherwise, answer with “0”. Do not write any other text, answer only with “1” or “0”.
binary_yes_no	Answer “Yes” if there is a violation of the cease and desist letter in the product description. Otherwise, answer “No.” Do not write any other text, answer only with “Yes” or “No”.
binary_semantic	Answer “Violation” if there is a violation of the cease and desist letter in the product description. Otherwise, respond with “No violation.” Do not write any other text, answer only with “Infringement” or “No Infringement”.

Table 2. Prompt Components

We send every generated output to a parser that trims any leading or trailing whitespace and attempts to match it to either "1" or "0", "yes" or "no", and "violation" or "no violation". If a match is identified, the model's prediction is recorded as a binary value. If no match is found, the example is labeled incorrect, affecting only the total accuracy of the classification metrics.

The experiments were run over the course of five days on an `NVIDIA DGX A100` instance utilizing six `NVIDIA A100` GPUs with 80 GB of VRAM each. The results were saved to an MLflow tracking server and evaluated using the `pandas` and `matplotlib` Python libraries.

6 Results & Discussion

We present a statistical evaluation of model performance as measured by total accuracy, accuracy and micro F1 across all prompt variants in Table 3. The data shows that `StableBeluga2` was the best performing model with a total accuracy of 84.5% on the best prompt variant, meaning it generated a valid and correct prediction for 84.5% of test cases. It is closely followed by `Platypus2` (81.9%) and `gpt-3.5-turbo-0301` (80.17%). The newer `gpt-3.5-turbo-0613` performed significantly worse and `LLaMA2` achieved the lowest performance. Looking at the mean total accuracy over all variants, `StableBeluga2` again seemed to provide the most reliable performance while the other models showed a fairly low average performance. This can also be seen by looking at the *Min* and *Std* (standard deviation) columns. Except for `StableBeluga2`, all models had some variants with a total accuracy of or close to 0% and also fairly high standard deviation. This can be attributed to a large portion of prompt variants for which the models did not generate any or few valid parseable outputs. Interestingly, while `LlaMA2` generally performed worst, its performance appears to be more stable across prompt variants than the other models (excluding `StableBeluga2` which performed exceptionally well on all prompt variants).

Even when only considering the valid outputs of each model using the accuracy metric, we can see that `StableBeluga2` was the most reliable with an average 80% of valid outputs being correct though `gpt-3.5-turbo-0301` was not far behind. Also, `gpt-3.5-turbo-0301` seemed to perform much more reliably with a significantly reduced standard deviation. `LLaMA2`, on the other hand, appeared to have a much higher variance across variants. This does not change significantly when looking at the micro F1 scores which means that all models performed similarly on both classes. For clarity, we omit the *Max* and *Min* values for accuracy and micro F1. Displaying them could be misleading, as certain prompt variants may result in a limited number of valid outputs. If these outputs happen to be correct, it could artificially inflate performance metrics.

To conclude, our evaluation shows that `StableBeluga2` was by far the best performing and most reliable model across all prompt variants, classes and metrics. In second place, we see `gpt-3.5-turbo-0301`, closely followed by `Platypus2`. The newer `gpt-3.5-turbo-0613` exhibited significantly lower performance and higher variance across prompt variants compared to its predecessor. This ob-

servation is not surprising as changes in model performance over time have also been reported by other researchers for this model though performance of `gpt-3.5-turbo` generally improved over time in their experiments [5]. `LLaMA2` performed worst overall.

	Total Accuracy				Accuracy		Micro F1	
Model	Mean	Max	Min	Std	Mean	Std	Mean	Std
gpt-3.5-turbo-0301	50.34%	80.17%	3.45%	18.77%	72.50%	7.76%	67.22%	10.13%
gpt-3.5-turbo-0613	33.37%	75.86%	0.00%	31.29%	59.76%	33.96%	56.52%	32.56%
llama2	9.90%	61.21%	0.00%	14.13%	42.66%	44.97%	40.11%	43.17%
platypus2	49.55%	81.90%	0.00%	35.33%	49.55%	35.33%	50.42%	35.95%
stablebeluga2	**79.55%**	**84.48%**	**70.69%**	**3.33%**	**80.00%**	**3.52%**	**80.26%**	**2.85%**

Table 3. Performance of tested LLMs across all variants

Regarding the performance of individual prompt components, we present a statistical evaluation of total accuracy, accuracy and micro F1 for each component value across all models in Table 4. For the *role* component, using the values with id `expert` or `none` did not lead to a significant difference in performance. This is in-line with recent, anecdotal observations that role prompting is less effective or not effective at all for newer models. However, it might also be attributed to the fact that role prompting often impacts the style of the generated text which is not of importance in cases where binary outputs are generated.

	Total Accuracy		Accuracy		Micro F1	
ID	Mean	Std	Mean	Std	Mean	Std
Role						
expert	44.27%	32.47%	60.43%	32.86%	58.22%	32.24%
none	44.81%	33.19%	61.36%	33.03%	59.59%	32.56%
Instruction						
default	42.86%	33.30%	61.11%	32.95%	58.97%	32.54%
def_verb	44.62%	33.30%	61.82%	33.19%	59.31%	32.76%
short	47.01%	32.09%	61.47%	32.79%	59.74%	32.38%
verbose	43.67%	32.86%	59.17%	33.15%	57.60%	32.28%
Step-by-Step						
default	36.14%	33.51%	56.49%	36.64%	55.31%	36.26%
elmh	41.40%	32.81%	58.19%	34.88%	56.42%	34.15%
none	56.08%	28.78%	67.99%	25.06%	64.99%	24.95%
Output Format						
binary_numeric	35.27%	35.85%	46.53%	38.51%	45.98%	38.06%
binary_semantic	43.89%	34.25%	58.24%	34.70%	56.10%	34.25%
binary_yes_no	54.46%	24.47%	77.92%	8.39%	74.64%	10.63%

Table 4. Performance for each prompt component across all models

For the *instruction* component, the value with id `short` appeared to perform slightly better than other values. This is a surprising observation as it is typically recommended to provide the model with detailed and comprehensive instructions to achieve better performance. That being said, when only considering the successfully parsed examples, all values of the *instruction* component appeared to perform similarly. In other words, the `short` instruction seems to lead to a slightly higher rate of correctly formatted outputs but does not improve performance itself.

For the *step-by-step* component, we see that best performance (both total accuracy as well as raw performance) is achieved using no value at all. This stands in contrast to reports that framing instructions in an itemized manner or allowing for a chain-of-thought improves performance [16, 25]. However, the results presented here might be explained by the fact that these strategies are only effective when the model is allowed to output its thoughts which was not the case here. In all other cases, the model might be misled or distracted when providing these instructions.

Lastly, for the *output format* component, best performance was achieved using a "yes"/"no" format. Interestingly, this not only improved the models' ability to generate valid outputs but also their ability to correctly identify violations. Even more, variance in accuracy and micro F1 also appears to be significantly reduced across models when using the "yes"/"no" format.

To summarize, across all models, only the selection of the *step-by-step* and *output format* components appeared to make a significant difference for model performance. In particular, leaving out the step-by-step instruction and requesting a simple "yes"/"no" format had the highest positive impact on performance. This finding is consistent with the fact that irrelevant text input can potentially distract LLMs from the core information and consequently decrease performance [20].

Regarding individual combinations of components (i.e. individual prompts) across models, we display the top-3 prompts (ranked by total accuracy, accuracy and micro F1) in Table 5. We can see that best performance (both total accuracy, accuracy and micro F1) was achieved using a combination of `none` for the *role* component and `binary_yes_no` for *output format*. While the result for the output format is not surprising, the clear result for the role component is slightly unexpected as there was no clear difference looking at the components individually. However, even a small performance boost for one value (which can indeed be seen in Table 4 when comparing `expert` vs `none`) can lead to it dominating the best instances due to tail dependence.

Interestingly, leaving out the *step-by-step* component was only optimal in terms of total accuracy. This suggests that not providing a step-by-step instruction generally helped the model generate more valid outputs, but adding it *can* significantly improve raw model performance. That being said, standard deviation was exceptionally high for values other than `none` (see Table 4).

With respect to the *instruction* component, no clear effect can be seen as all component values are present in the top-3 runs (see Table 5).

Component IDs				Total Accuracy		Accuracy		Micro F1	
Role	Instr.	St.-St.	Output Form.	Mean	Std	Mean	Std	Mean	Std
Top-3 by Total Accuracy									
none	default	none	binary_y_n	69.66%	7.01%	73.52%	2.44%	69.35%	6.99%
none	verbose	none	binary_y_n	68.97%	8.18%	74.48%	1.54%	70.81%	6.71%
none	short	none	binary_y_n	68.28%	22.50%	77.94%	3.31%	73.91%	8.64%
Top-3 by Accuracy									
none	short	default	binary_y_n	50.52%	29.53%	83.10%	11.40%	82.17%	10.53%
none	def_verb	default	binary_y_n	44.31%	31.30%	82.89%	8.61%	80.30%	10.67%
none	short	elmh	binary_y_n	52.41%	25.64%	80.78%	12.04%	78.28%	13.89%
Top-3 by Micro F1									
none	short	default	binary_y_n	50.52%	29.53%	83.10%	11.40%	82.17%	10.53%
none	def_verb	default	binary_y_n	44.31%	31.30%	82.89%	8.61%	80.30%	10.67%
none	short	elmh	binary_y_n	52.41%	25.64%	80.78%	12.04%	78.28%	13.89%

Table 5. Performance for individual Prompts across all models

For performance of individual prompts per model, we display the top-3 prompts by total accuracy, accuracy and micro F1 per model (excluding the worst performing models `LLaMA2` and `gpt-3.5-turbo-0613`) in Table 6. These results are generally consistent with our previous findings. One thing to note is that only the `Platypus2` model seemed to perform best using a "yes"/"no" output format. All other models performed better using a numeric or semantic format. Manual inspection showed that this is due to the fact that `Platypus2` mostly outputs decimal numbers instead of integers resulting in invalid outputs for a numeric output format.

Also, only `Platypus2` consistently performed best without a step-by-step instruction. The other models benefited from a more detailed guidance when looking at accuracy and micro F1 (both `StableBeluga2` and `gpt-3.5-turbo-0301`) or even total accuracy (only `StableBeluga2`).

An important observation from Table 6 is that while `gpt-3.5-turbo-0301` achieved exceptionally high accuracy and micro F1 scores (100%) with some prompt variants, its low total accuracy ($< 10\%$) suggests that the model produced a small number of valid outputs in these cases. Of the valid outputs, all were correct. It's crucial to understand that these metrics, in isolation, do not capture the model's overall performance.

Finally, for the best model (`StableBeluga2`), we present the performance (total accuracy) of each prompt variant in a grid of heatmaps in Figure 2.[4] The figure clearly shows that performance was lowest for the "yes"/"no" output format (middle column) and best for the numeric format (left column). Also, there does not seem to be a significant difference between the `none` (top row) and `expert` (bottom row) values for the *role* component which is consistent with our previous findings. While generally, there was not a big performance difference between the *step-by-step* component values, removing that component led to a

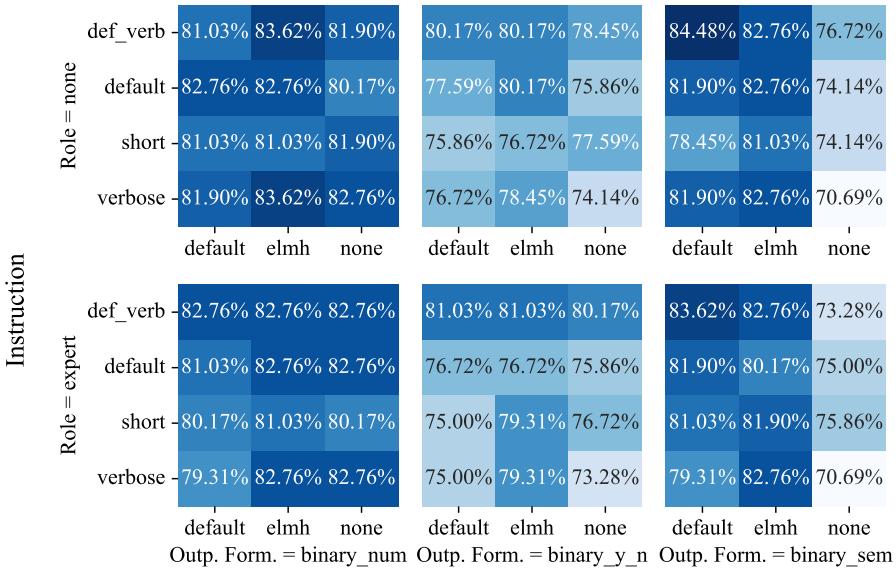
[4] Heatmap visualizations are available for all models in our project GitHub repository: `https://github.com/ecapx/kivedu-public`

Model	Role	Instr.	St.-St.	Output Form.	Total Acc.	Accuracy	Micro F1
Top-3 by Total Accuracy per Model							
gpt-3.5-turbo-0301	none	short	none	binary_num	80.17%	80.17%	79.86%
	none	short	none	binary_y_n	80.17%	80.17%	78.72%
	expert	def_verb	elmh	binary_num	74.14%	74.78%	68.87%
platypus2	none	short	none	binary_y_n	81.90%	81.90%	82.67%
	none	def_verb	none	binary_y_n	78.45%	78.45%	78.90%
	none	default	none	binary_sem	77.59%	77.59%	78.74%
stablebeluga2	none	def_verb	default	binary_sem	84.48%	85.22%	84.57%
	none	verbose	elmh	binary_num	83.62%	84.35%	83.28%
	none	def_verb	elmh	binary_num	83.62%	84.35%	82.67%
Top-3 by Accuracy per Model							
gpt-3.5-turbo-0301	none	default	default	binary_num	6.03%	100.00%	100.00%
	none	default	default	binary_sem	3.45%	100.00%	100.00%
	none	def_verb	default	binary_num	15.52%	94.74%	93.93%
platypus2	none	short	none	binary_y_n	81.90%	81.90%	82.67%
	none	def_verb	none	binary_y_n	78.45%	78.45%	78.90%
	none	default	none	binary_sem	77.59%	77.59%	78.74%
stablebeluga2	none	def_verb	default	binary_sem	84.48%	85.22%	84.57%
	none	verbose	elmh	binary_num	83.62%	84.35%	83.28%
	none	def_verb	elmh	binary_num	83.62%	84.35%	82.67%
Top-3 by Micro F1 per Model							
gpt-3.5-turbo-0301	none	default	default	binary_num	6.03%	100.00%	100.00%
	none	default	default	binary_sem	3.45%	100.00%	100.00%
	none	def_verb	default	binary_num	15.52%	94.74%	93.93%
platypus2	none	short	none	binary_y_n	81.90%	81.90%	82.67%
	none	def_verb	none	binary_y_n	78.45%	78.45%	78.90%
	none	default	none	binary_sem	77.59%	77.59%	78.74%
stablebeluga2	none	def_verb	default	binary_sem	84.48%	85.22%	84.57%
	expert	def_verb	default	binary_sem	83.62%	84.35%	84.35%
	none	default	elmh	binary_sem	82.76%	83.48%	83.71%

Table 6. Performance for individual Prompts per Model

significant performance drop when using the `binary_semantic` output format (heatmaps in right column). We are not sure why this is the case, it might be that a numeric or "yes"/"no" output is less ambiguous and less context-dependent than a semantic output format, enabling the model to perform well even when no explicit guidance is provided.

Fig. 2. Total Accuracy of StableBeluga2 for each prompt variant



7 Conclusion

In this paper, we evaluated multiple state-of-the-art LLMs using different prompt variants on the task of identifying cease and desist violations in German product descriptions. We ran evaluations for 5 LLMs (2 proprietary, 3 open source) with 72 prompt variants (consisting of 4 different components) each on a manually curated dataset of 116 examples. Our objective was to determine the superior model for this particular task (**RQ1**) and to understand the influence of prompt variations on model performance (**RQ2**).

For **RQ1**, our evaluation reveals that `StableBeluga2` outperformed the other models, achieving the highest metrics in both accuracy and micro F1 score. Additionally, `StableBeluga2` proved to be the most reliable, consistently delivering high performance with minimal variance across prompts. Following closely be-

hind were `Platypus2` and `gpt-3.5-turbo-0301`, which also exhibited high performance for the most effective prompt variants but showed significant variations depending on the specific prompt. Although `gpt-3.5-turbo-0613` demonstrated good performance in optimal conditions, it was highly sensitive to the choice of prompt and ultimately yielded a low average performance. `LLaMA2` was the worst-performing model, scoring lowest in both average and peak performance.

Regarding **RQ2**, our experiments demonstrate that the choice of prompt can have a significant impact on model performance. This variation can be attributed to the difficulty some prompts introduce in task comprehension for the model. Additionally, some prompts lead to an increased number of invalid outputs, consequently lowering the overall performance. We found that there was no significant difference between including a role in the prompt versus omitting it. Similarly, providing the model with a detailed versus a short instruction only had a low impact on performance. However, generally speaking, providing a step-by-step instruction decreased model performance significantly compared to not providing one - both in terms of total accuracy, taking invalid outputs into account, as well as accuracy, leaving out invalid outputs. Moreover, the performance was highest when using a "yes"/"no" output format, compared to semantic or numeric output formats.

We also observed that several of these results were highly dependent on the particular model. For example, the highest performance for `gpt-3.5-turbo-0301` and `StableBeluga2` were achieved with a numeric or semantic output format, not a "yes"/"no" format. Similarly, when a step-by-step instruction was present, these models appeared to perform better on raw identification but worse on total accuracy. Thus, the optimal prompt is highly dependent on the employed LLM and only few general recommendations can be made independent of models.

As part of the KIVEDU project, this evaluation provides a promising assessment of the application of LLMs for the detection of cease and desist violations in German product descriptions. It demonstrates that AI can support and automate the current process to ensure consumer rights and foster fair competition. In future work, we aim to conduct a more comprehensive evaluation that encompasses additional LLMs, a broader range of prompt variants, including one-shot and few-shot prompting, as well as multiple languages and an expanded dataset. In addition, we plan to fine-tune the tested LLMs with domain-specific data to further improve performance.

References

1. Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., Wolf, T.: Open LLM Leaderboard (2023), `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`
2. Braun, D., Matthes, F.: NLP for Consumer Protection: Battling Illegal Clauses in German Terms and Conditions in Online Shopping. In: Proceedings of the 1st Workshop on NLP for Positive Impact. pp. 93–99. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.nlp4posimpact-1.10

3. Braun, D., Scepankova, E., Holl, P., Matthes, F.: Consumer Protection in the Digital Era: The Potential of Customer-Centered LegalTech. In: David, K., Geihs, K., Lange, M., Stumme, G. (eds.) INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft. pp. 407–420. Gesellschaft für Informatik e.V., Bonn (2019). https://doi.org/10.18420/inf2019_58
4. Chakrabarti, D., Patodia, N., Bhattacharya, U., Mitra, I., Roy, S., Mandi, J., Roy, N., Nandy, P.: Use of Artificial Intelligence to Analyse Risk in Legal Documents for a Better Decision Support. In: TENCON 2018 - 2018 IEEE Region 10 Conference. pp. 0683–0688. IEEE, Jeju, Korea (South) (Oct 2018). https://doi.org/10.1109/TENCON.2018.8650382
5. Chen, L., Zaharia, M., Zou, J.: How is ChatGPT's behavior changing over time? (Aug 2023). https://doi.org/10.48550/arXiv.2307.09009, http://arxiv.org/abs/2307.09009, arXiv:2307.09009 [cs]
6. Contissa, G., Docter, K., Lagioia, F., Lippi, M., Micklitz, H.W., Pałka, P., Sartor, G., Torroni, P.: Claudette Meets GDPR: Automating the Evaluation of Privacy Policies Using Artificial Intelligence (Jul 2018). https://doi.org/10.2139/ssrn.3208596
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (May 2019). https://doi.org/10.48550/arXiv.1810.04805
8. European Commission: 2016/0148 (COD) Cooperation between national authorities responsible for the enforcement of consumer protection laws (May 2016), https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52016PC0283
9. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models (Oct 2021). https://doi.org/10.48550/arXiv.2106.09685, http://arxiv.org/abs/2106.09685, arXiv:2106.09685 [cs]
10. Juranek, S., Otneim, H.: Using machine learning to predict patent lawsuits (Jun 2021). https://doi.org/10.2139/ssrn.3871701
11. Lee, A.N., Hunter, C.J., Ruiz, N.: Platypus: Quick, Cheap, and Powerful Refinement of LLMs. arXiv preprint arxiv:2308.07317 (2023)
12. Lippi, M., Palka, P., Contissa, G., Lagioia, F., Micklitz, H.W., Panagis, Y., Sartor, G., Torroni, P.: Automated Detection of Unfair Clauses in Online Consumer Contracts. In: Legal Knowledge and Information Systems, pp. 145–154. IOS Press (2017). https://doi.org/10.3233/978-1-61499-838-9-145
13. Lippi, M., Pałka, P., Contissa, G., Lagioia, F., Micklitz, H.W., Sartor, G., Torroni, P.: CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. Artificial Intelligence and Law **27**(2), 117–139 (Jun 2019). https://doi.org/10.1007/s10506-019-09243-2
14. Liu, Q., Wu, H., Ye, Y., Zhao, H., Liu, C., Du, D.: Patent Litigation Prediction: A Convolutional Tensor Factorization Approach. In: International Joint Conference on Artificial Intelligence (2018). https://doi.org/10.24963/ijcai.2018/701
15. Mahan, D., Carlow, R., Castricato, L., Cooper, N., Laforte, C.: Stable Beluga models, https://huggingface.co/stabilityai/StableBeluga2
16. Mishra, S., Khashabi, D., Baral, C., Choi, Y., Hajishirzi, H.: Reframing Instructional Prompts to GPTk's Language. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 589–612. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.findings-acl.50, https://aclanthology.org/2022.findings-acl.50

17. Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., Awadallah, A.: Orca: Progressive Learning from Complex Explanation Traces of GPT-4 (Jun 2023). https://doi.org/10.48550/arXiv.2306.02707, http://arxiv.org/abs/2306.02707, arXiv:2306.02707 [cs]
18. OpenAI: Introducing ChatGPT (Nov 2022), https://openai.com/blog/chatgpt
19. Shanahan, M., McDonell, K., Reynolds, L.: Role-Play with Large Language Models (May 2023). https://doi.org/10.48550/arXiv.2305.16367, http://arxiv.org/abs/2305.16367, arXiv:2305.16367 [cs]
20. Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., Schärli, N., Zhou, D.: Large Language Models Can Be Easily Distracted by Irrelevant Context (Jun 2023). https://doi.org/10.48550/arXiv.2302.00093, http://arxiv.org/abs/2302.00093, arXiv:2302.00093 [cs]
21. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open Foundation and Fine-Tuned Chat Models (Jul 2023). https://doi.org/10.48550/arXiv.2307.09288, http://arxiv.org/abs/2307.09288, arXiv:2307.09288 [cs]
22. Trappey, C.V., Trappey, A.J.C., Liu, B.H.: Identify trademark legal case precedents - Using machine learning to enable semantic analysis of judgments. World Patent Information **62**, 101980 (Sep 2020). https://doi.org/10.1016/j.wpi.2020.101980
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
24. Waidelich, L., Lambert, M., Al-Washash, Z., Kroschwald, S., Schuster, T., Döring, N.: Using Large Language Models for the Enforcement of Consumer Rights in Germany. In: Maślankowski, J., Marcinkowski, B., Rupino da Cunha, P. (eds.) Digital Transformation. pp. 1–15. Lecture Notes in Business Information Processing, Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-43590-4_1
25. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Jan 2023). https://doi.org/10.48550/arXiv.2201.11903, http://arxiv.org/abs/2201.11903, arXiv:2201.11903 [cs]
26. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.Y., Wen, J.R.: A Survey of Large Language Models (Jun 2023), http://arxiv.org/abs/2303.18223, arXiv:2303.18223 [cs]
27. Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P., Irving, G.: Fine-Tuning Language Models from Human Preferences (Jan 2020). https://doi.org/10.48550/arXiv.1909.08593, http://arxiv.org/abs/1909.08593, arXiv:1909.08593 [cs, stat]