Evaluating LLM-Prompting for Sequence Labeling Tasks in Computational Literary Studies

Axel Pichler
Department of German Studies
University of Vienna
axel.pichler@univie.ac.at

Janis Pagel and **Nils Reiter**
Department of Digital Humanities
University of Cologne
{firstname.lastname}@uni-koeln.de

Abstract

Prompt engineering holds the promise for the computational literary studies (CLS) to obtain high quality markup for literary research questions by simply prompting large language models with natural language strings. We test prompt engineering's validity for two CLS sequence labeling tasks under the following aspects: (i) how generalizable are the results of identical prompts on different dataset splits?, (ii) how robust are performance results when re-formulating the prompts?, and (iii) how generalizable are certain fixed phrases added to the prompts that are generally considered to increase performance. We find that results are sensitive to data splits and prompt formulation, while the addition of fixed phrases does not change performance in most cases, depending on the chosen model.

1 Introduction

Large language models (LLMs) have taken over the field of natural language processing (NLP) in the past years. LLMs implement the transformer architecture and are fine-tuned to follow instructions (Mishra et al., 2022; Zhang et al., 2024), which also led to the introduction of a new paradigm: 'prompting'.[1] In contrast to pre-training, fine-tuning or classical machine learning, prompting does not actually update the weights of the model itself. Instead, prompt strategies aim at producing the best possible prompt for a given task (Liu et al., 2023), thus providing a textual context for the model to generate reasonable replies.

LLM-prompting is a promising development for digital humanities in general, because task descriptions can be expressed in natural language, presumably making it easier to connect to classical, non-digital research in the humanities. This may also apply to the model's output, if it is in natural language or can be verbalized (correctly) as such. A distinction can be made between two prompting scenarios: i) Interactive prompting, as with a chatbot, is the scenario in which most people currently experience LLMs, as it is easily available even without technical background. It is characterized by a direct application and associated implicit validation, often used in an exploratory manner. Note that results obtained must not be perfect or even correct to be useful, and in following Gricean conversation maxims (Grice, 1975), human users put in interpretation effort to make sense of the results. ii) Batch-use comes into play if prompts are applied to a large(r) quantity of data, and the LLM is used for automatic detection of some textual concept. This paradigm is closely related to established machine learning scenarios, and thus needs to follow established machine learning best practices. The remainder of this article is about this batch-use of LLM prompting.

Evaluation of LLMs can also be separated into two areas: i) With the goal of evaluating LLMs as such (and unrelated to a specific task), they are usually confronted with test items from multiple benchmark data sets that cover a certain range of tasks. ARC (Clark et al., 2018), for instance, defines 7787 natural science questions with four possible answers, out of which one is correct. The model is tasked to provide the identifier of the correct answer. Models can then be ranked according to their (average) performance on such benchmarks, resulting in rankings such as the HuggingFace Open LLM Leaderboard[2]. ii) For a task-specific evaluation, reference data for the specific task is needed, and allows comparing system and reference output as is established in machine learning. In both evaluation setups, it is important to realize that what is evaluated is not (only) the model itself, but a tuple of model, task formalization, parameters and prompt, and that an exhaustive evaluation of all possible settings is usually not possible. This paper, as

[1] Also called 'in-context-learning' (Brown et al., 2020).

[2] https://tinyurl.com/3ms6bmhm

Proceedings of LaTeCH-CLfL 2025, pages 32–46

do many others, selects a number of parameters for the experiments and this selection has theoretical and pragmatic reasons.

This paper explores the use of LLM-prompting in computational literary studies (CLS). CLS analyzes literary texts and text corpora using methods of statistics, machine learning and NLP. In doing so, CLS draws partly on traditional literary studies, but does so with the help of data-driven approaches and methods. Past studies in CLS focused on authorship attribution, drama and genre analysis, literary-historical questions, narratological and gender analysis and questions of canonicity (cf. Schöch et al., 2023; Pielström et al., 2023; Andresen and Reiter, 2024). Non-computational literary research questions are typically highly complex, context-dependent and embedded in a deep theoretical framework, that is often expressed somewhat vaguely. Addressing such questions thus requires a multitude of tools and methods that form components in an argumentation that uses manual and automatic work steps. The tasks we discuss in this paper are representative for such components.

Concretely, this paper's contribution is the systematic evaluation of a number of LLMs and prompts on two different CLS-relevant sequence classification tasks for which manually annotated reference data sets exist. Sequence classification in NLP is the task of assigning a categorical label to each element in a sequence of data, such as words in a sentence or characters in a word. Such tasks are complex as they combine two potentially separate work steps in one: the selection of a token span to be classified and the classification of this span. Such tasks are common in CLS as manual annotation tasks.[3]

An important methodological aspect of such an evaluation is that as soon as prompting strategies make use of manually or automatically optimizing prompts on a data set ("prompt engineering"), this needs to be treated as a training process, even if no weight updates are performed: Selecting the best prompt on a data set and evaluating its performance on the very same data set is a case of overfitting and the measured performance is not indicative of its performance on new data. This does not mean that performance on unseen data must be lower in every case – if the model-prompt-combination has generalized properly, it may even achieve similar performance on unseen data. We suspect that in practice this optimization process is usually based on a small, hand-picked selection of examples, and often not evaluated on an independent test set. Accordingly, to avoid overfitting, we propose to follow established best practices and make a (documented) split into train and test data, with similar roles as in classical machine learning: Train data is used to optimize a prompt and test data to evaluate it.

Research questions. Against this background, we will focus on the following three research questions: i) **How generalizable are performance measurements?** This question rests on the assumption that a good model shows similar performance on different data sets. If its performance varies strongly, the model has failed to capture the essence of the task. ii) **How robust is the model against meaning-preserving prompt variations?** This question is related to the issue that Mizrahi et al. (2024) have uncovered (and named "prompt brittleness"): That the performance of prompted LLMs reacts very strongly to minor changes in the prompts, be it minimal changes such as adding or changing punctuation marks, or lexical changes such as paraphrasing the task. iii) **How generalizable are recommendations on prompt components?** Because an exhaustive search over all possible prompts (or other parameters) is impossible, prompting usually relies on best practices developed in interactive prompting scenarios (Saravia, 2022; Bsharat et al., 2024), such as promising the model a reward. Our question is to find out whether following these best practices for non-interactive prompting leads to consistently best (or even good) results. I.e., we investigate if general recommendations on how to construct a prompt actually lead to performance gains and/or consistently best results on CLS tasks and data set.

Documentation of all our experiments (including prompt templates) is done in a GitHub repository, to facilitate the reproduction of our experiments.[4]

2 Related Work

Several studies in NLP use LLMs for classic classification tasks. Balkus and Yan (2023) use GPT-

[3] Following the categorization of classification tasks in cultural analytics according to Bamman et al. (2024), this primarily involves the category of "replacing human labeling at scale," which is also a prerequisite for "top-down theory testing". Note also the survey paper by Hatzel et al. (2023) on machine learning in computational literary studies.

[4] https://github.com/pagelj/prompt-cls

3's API to classify the topics of short texts and use both the generative completion capabilities as well as a dedicated classification end point of the API. Zhao et al. (2023) use ChatGPT to classify agriculture-related texts with regards to sentiment, prediction of natural disasters and text topic. Wang et al. (2023) test GPT-3.5, GPT-4 and Llama 2 on, among others, sentiment analysis of tweets. In addition to this, Clavié et al. (2023) show that in the binary classification of qualification requirements for job advertisements, LLMs such as OpenAI's text-davinci-003 model clearly outperform classical ML approaches such as SVM but also smaller 'foundational models' such as DeBERTaV3.

Many studies investigate the influence of prompts for prediction performance (Schick and Schütze, 2021; Zhao et al., 2021; Perez et al., 2021; Lu et al., 2022; Ceron et al., 2024). All come to the conclusion that the form and quality of manually crafted prompts is highly influential on performance and often suggest methods for automatically generating prompts or using methods such as prompt tuning to circumvent the shortcomings of hard prompts. Many studies distinguish systematically between different prompt components, such as "Definition", "Things to Avoid", etc. (Mishra et al., 2022). Sadr et al. (2025) investigate which words are most important in a prompt by systematically replacing words in prompt components like "Let's think step-by-step" and measure the change in performance via a newly introduced metric. They find that nouns are consistently among the most important words regarding prediction and that the most important word varies according to the task performed. Mizrahi et al. (2024) demonstrate how single prompts lead to chance-based outcomes and suggest using a suite of prompts and averaging over their performance (this strategy is called 'prompt ensemble' in Liu et al. (2023)). Lastly, Schaeffer et al. (2023) suggest that the proclaimed emergent abilities of LLMs disappear once appropriate evaluation metrics are used.

The largest study on the usages of LLMs for classification tasks in a computational humanities context to date comes from Ziems et al. (2024). They work in the context of computational social science and perform zero-shot learning on a wide variety of tasks on different textual levels such as sarcasm and ideology detection, misinformation detection, empathy classification, politeness, event detection and roles and tropes. The study uses one prompt template per task and does not address the potential impact of prompt brittleness on the evaluation. They find that, except for certain minor tasks, LLMs in a zero-shot setting are not able to outperform fine-tuned classifiers or replace the work of human annotators (Ziems et al., 2024, p. 240).

Pichler and Reiter (2024) come to a similar conclusion in the context of an ICL-experiment in the CLS, in which they investigate the extent to which OpenAI's text-davinci-003-LLM can reproduce the performance of smaller older models used by Piper (2020) in the course of a classification task based on complex knowledge from literary theory, namely the determination of domain specific generalizing statements in literary studies.

Pagel et al. (2024) tested several open and close-sourced LLMs in zero and few-shot setups on the task of identifying knowledge transfers about family relations in German dramas. They also conclude that, in the current state, LLMs are not suitable to sufficiently perform high-level CLS classification tasks out-of-the-box.

Bamman et al. (2024), recently published as a pre-print, arrives at differentiated results. The study identifies ten tasks from computer-assisted text analysis, characterized as cultural analytics, for which annotated reference data is available, and investigates how well these tasks can be solved by LLMs compared to pretrained language models (PLMs). The choosen LLMs are GPT-4o, LLAMA 3 70B and Mixtral 8x22B, which are prompted with a single prompt template containing 10 examples but no Chain-of-thought-prompts. They find that "LLMs offer competitive performance through prompting alone for established tasks, while traditional supervised methods excel for newly constructed phenomena (even in scenarios with limited training data)". In a further comparison, for which the models were fine-tuned on the task-specific reference data, the performance differences between masked PLMs and LLMs are even smaller. Issues of prompt brittleness and prompt generalizability are not addressed.

Hicke et al. (2024) perform zero-shot classification for focalization on 16 Stephen King novels with LLAMA 3 and GPT-4o and compare to a NaiveBayes and DistilBERT baseline. They find that GPT-4o performed best with an F1 score of 86.90, but also that initial inter-annotator agreement between the three annotators was relatively low with Krippendorff's α of 0.55. However, an ad-

judicated version could be created after discussion between the annotators. They also find a correlation between a model's confidence scores and its performance, as well as a robustness of GPT-4o's performance with regard to multiple runs and small changes in the prompt.

We are not aware of any studies dedicated to sequence classification tasks in CLS.

3 Sequence Classification Tasks and Data

This section describes the two sequence classification tasks (emotion and event) and data sets used in our experiments. Note that the event dataset is in German, while the emotion dataset is in English language. Regarding the issue of data leakage (Balloccu et al., 2024), please also note that both the emotion and event dataset are publicly available. It can therefore not be excluded that (parts of) the public data sets and their annotated labels have been included in the pre-training of our models.

Emotion The dataset for the emotion task is coming from work by Kim and Klinger (2018) and is called REMAN (Relational Emotion Annotation for Fiction). They provide annotations of 200 English texts from Project Gutenberg[5] and annotate the emotions *anger*, *anticipation*, *disgust*, *fear*, *joy*, *sadness*, *surprise* and *trust* plus a category *other emotion* for cases that do not fall into one of the above. Annotated is either a single word or phrase with a preference for shorter spans. For instance, the annotated span for the sentence "His smile was distinctly attractive." is "smile" and was given the *joy*-label. In a multi-step process, all spans that do not match exactly between annotators, but overlap, were adjudicated by an expert.

Kim and Klinger provide baseline experimental results on predicting emotions on their dataset, using dictionary and bag-of-words-based baselines, a conditional random field (CRF) model as well as a long short-term memory model (LSTM) architecture with a CRF classification on top. The LSTM-CRF performs best with an F1 score of $43\,\%$ in a strict setting where all spans have to match exactly, but the authors note that recall is low for both models. They report inter-annotator agreement scores for their annotations, ranging from an average Cohen's κ of 0.11 for *anticipation* to a κ value of 0.35 for *joy*. See Table 3 for an example of each emotion.

Event Vauth and Gius (2022) take six German-language texts from the TextGrid[6] and d-prose (Gius et al., 2021) repositories. They annotate three different event types, *process*, *stative* and *change of state*, as well as *non-event* (see Vauth and Gius, 2021). Each span receives exactly one of these labels.

The original annotation task consisted of three parts: In a first step, the annotation span had to be identified, in a second step it had to be marked with the corresponding labels, and then in a third step subordinate property tags had to be assigned. Following this procedure, they achieved an agreement for these event types of Krippendorff's α between 0.57 and 0.75, depending on the text.

To our knowledge, there are currently no published studies on automatic annotation of the dataset. Examples for annotation spans for each of the four categories look like the ones in Table 4.

4 Formalization

In this section, we describe which measurement techniques we use to answer the three research questions introduced above. In general, our prompts consist of a frame structure describing the role of the LLM, the task, the expected output format, and the labels to be used, with slots for variable components and the text to analyze: A prompt is thus defined as a complete input sequence that realizes one of 8 possible combinations of so-called prompt components, where *prompt components* are elements that can be switched on and off. The implementation of one of these 8 possible combinations as a prompt, we call *prompt configuration*. Additionally, there are 3 *paraphrases* (semantically equivalent reformulations) of each prompt. These were generated automatically by using GPT to generate 10 alternative reformulations based on an initial manually created prompt that follows current prompt engineering recommendations, from which we then manually selected three. All in all, this leads to $4 * 8 = 32$ different prompt configurations — for each model and each task — which results in a grand total of 64 different prompts and 256 model runs.

4.1 RQ1: Generalizability of Performance Measurements

To check whether and to what extent a particular prompt configuration performs equally well on dif-

[5] https://www.gutenberg.org/

[6] https://textgridrep.org/

ferent test samples, we proceed as follows: For each model, we test each prompt configuration on two test data sets and calculate the difference and p-values between the F1 scores obtained using a paired sample t-test. This way, we test the null hypothesis that different data samples have no effect on the performance.

4.2 RQ2: Robustness against Meaning-Preserving Prompt Variations

In order to investigate how robust each model is against semantic rephrasings in prompt formulations, we first define (with the help of a language model) four different but semantically equivalent paraphrases of each (fully instantiated) prompt. These changes cover the entire prompt: Next to the prompt components, elements of the frame structure of the prompt are also reformulated (see listings 1-4). We then look at the standard deviation of F1 scores over each of those prompt variants by comparing the paraphrases that realize the same components. We hypothesize that a more robust model is less sensitive against these paraphrases, and thus shows lower standard deviation.

4.3 RQ3: Generalizability of Prompt Component Optimization

For the final research question, we investigate how well different components added to a prompt generalize across tasks and models.

Under the term *component*, we understand phrases or instructions added to the prompt that are meant to improve model performance, but are not specific to solving a concrete task. One of the most popular examples of such a component is to assign a **role** or occupation to the model and ask it to provide an answer under the assumption that it behaves like a person with the specified role (for example "You are an expert mathematician").

Bsharat et al. (2024) provide an extensive list of principles to construct good prompts, including prompt components, from which we pick three that we perceive as currently popular options: (i) the model gets **bribed** to give a good answer, (ii) the **stakes** are high, and (iii) the model should think **step** by step.[7]

Concretely, we checked which of the prompt components were present in the best performing prompts per model and how often. This investigation sheds light on which components actually make a measurable positive impact on performance. We hypothesize that, provided the components are actually useful in boosting model performance, they should appear in all or close to all of the best-performing prompt variations.

5 Experiments

We carry out experiments on all tasks described above, using the following LLMs: GPT[8] (GPT-4o[9]), LLAMA (Llama3.1-8B-Instruct (AI@Meta, 2024)[10], MIXTRAL (Mixtral-8x7B-Instruct (Jiang et al., 2024)[11]), and SAUERKRAUT (SauerkrautLM[12]). The models provide a balance of close and (semi-)open source systems and with SAUERKRAUT there is a model that was especially re-pretrained on German language texts. Furthermore, all models displayed high scores on popular NLP benchmarks and should therefore generally be able to tackle the two CLS tasks. Due to the computer resources available, we quantified LLAMA and SAUERKRAUT into a 4-bit version using HuggingFace's `bitsandbytes` library.

5.1 Experimental Setup

For the **Event** dataset, we remove annotated categories which occur less than 600 times. This leads to the *change of state* class being removed, leaving us with the *process*, *stative* and *no event* labels. We use a single text out of four, Effi Briest by Theodor Fontane, as it is by far the longest text and the only one for which the requirement of 600 instances per class can be kept. As the **Emotion** data set is smaller, we have set a threshold of at least 150 occurrences per label. This leaves us with the classes *anger*, *disgust*, *joy*, *sadness* and *surprise*.

From these samples, we create two random subsets for each task, each with 15 % of the instances. The distribution of labels in each subset corresponds to the distribution of label occurrences in the whole dataset. These sets are subsequently called *test 1* and *test 2*.

For all tasks, each prompt contained only a single target sentence together with a fixed frame and

[7] For the specific formulations of the components, see section B

[8] In the following, we will use short names in small caps to refer to the concrete models used in the experiments.

[9] https://www.wikidata.org/wiki/Q125919502

[10] https://huggingface.co/meta-llama/Meta-Llama-3.1-8b-Instruct

[11] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[12] https://huggingface.co/VAGOsolutions/Llama-3.1-SauerkrautLM-8b-Instruct

some of the components (see Listing 1 for an example). The models were asked to i) select word sequences that match the definition and ii) assign a class label in a second step. This procedure differs from the standard procedure for text and sequence classification in that the probabilities of the labels for a selection of tokens are not determined by the LLM, but rather the LLM is prompted to generate both the text sequence to be classified and the corresponding label. To evaluate the output of the LLMs generated in this way, we mapped the classified text sequences to the input sentence, then tokenized it and assigned the label "None" to all those tokens that were not labeled. The evaluation was then based on these token-label pairs.

For all models, we set the temperature to 0.1 and left `top_k` at the default of 5, in order to get results relatively close to deterministic for reproducability. For all other hyperparameters, we used the model-specific default values.

5.2 Results

Before discussing results related to our research questions, the general, best possible performance measured in F1 on the entire test set for each model can be seen in Table 1. Note that different models achieve best performance with different prompt configurations. As can be seen, performance scores for the emotion task are generally lower than for the event task. Best models are GPT (for emotion) and MIXTRAL (for event). We also compare with current average results from the HuggingFace Open LLM Leaderboard that — albeit on very different tasks than ours — are in a similar range. The HuggingFace average is composed of scores for six different benchmarks, including math problems, formatting challenges and language understanding. The leaderboard does not include results for GPT-4o. The similar range of results shows that the scores in our experiments are not only due to our CLS tasks, but also occur for more general tasks. It should however be noted that the standard deviation for the benchmark results from HuggingFace are relatively high, with some benchmarks showing scores of around 70% accuracy, while for other benchmarks, the accuracy is under 10%.

5.2.1 Generalizability of Performance Measurements

The results relevant to RQ1 can be found in Table 2. Generally, the models achieve a mean of differences for the different data sets between 4.2 % and 10.9 %. While these numbers seem small, they represent a deviation of up to almost 11 percentage points in F1 score, which would be a substantial difference for most applications. The differences between the F1 scores on the two data sets are statistically significant on both tasks for all models (p-values < 0.05). The null hypothesis that different data samples have no effect on the measurement of performance can therefore be rejected in all cases. This indicates that the measurement of the performance on one test set does not generalize well to another test set. It must therefore be expected that performance on new/unseen data sets is significantly different. Possible reasons for this are a.) that the models did not properly generalize (i.e., learn the true nature of the task) or b.) that the two test data sets are distributed differently.

Model	Emotion	Event	HF
GPT	27.04	29.03	-
LLAMA	19.21	28.93	28.20
MIXTRAL	22.72	32.6	23.84
SAUERKRAUT	21.79	28.04	28.68

Table 1: Overall best possible performance, measured in F1 score. Results have been achieved with different prompt configurations. We also compare to the average scores of the HuggingFace (HF) benchmark on https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, last access on 15^{th} November 2024.

Task	Model	Diff. (pp)
Emotion	GPT	7.7
	MIXTRAL	6.7
	LLAMA	4.2
	SAUERKRAUT	5.2
Event	GPT	6.2
	MIXTRAL	10.9
	LLAMA	6.6
	SAUERKRAUT	6.3

Table 2: Mean of differences of the F1-scores obtained on the two test stets and p-values between the two test sets per model for the Emotion and Event task. All differences are statistically significant ($p < 0.05$).

5.2.2 Model Robustness against Prompt Variations

The results for RQ2 can be found in Tables 5 and 6 (see Appendix) for the emotion and event task

respectively. Please note that the table shows mean and standard deviation of the F1 scores on the entire test data set (i.e., the union of *test 1* and *test 2*), using four different variants of the prompts.

Generally, the models achieve a mean standard deviation for the different component configuration between 2.4 and 5.92 %. While these numbers seem small, they represent a deviation of up to 6 percentage points in F1 score, which would be a substantial difference for most applications.

For the emotion task, LLAMA achieves the smallest deviation over the formulations, and can thus be considered the most robust model. For the event task, SAUERKRAUT achieves the smallest average deviation, although LLAMA's deviation is only slightly higher. GPT and MIXTRAL do not show an interpretable pattern in this evaluation.

Compared to the results reported by Mizrahi et al. (2024), we can confirm the observation that, depending on the prompt formulation, any ranking of the models can be achieved. We also note, however, that the deviations are much smaller, albeit on a generally low performance level.

5.2.3 Generalizability of Prompt Component Importance

The analysis of prompt components, shown in Figure 1 reveals that there are only few components that occur in all best performing prompts (**steps** three times, **bribe** one time out of a possible eight).[13] Only for LLAMA, **steps** occurs in all best performing prompts, making it the only occurrence were this happens. On average, components occur only half of the time in all best performing prompts across all models and tasks. Since this is around chance level and we expected to see a relatively high frequency for each component, we conclude that the components are generally not a useful addition to the prompts. Overall, no general recommendation can be derived from these figures for the inclusion of certain components in a uniformly designed prompt, at least for the two CLS tasks and four models examined.

6 Discussion

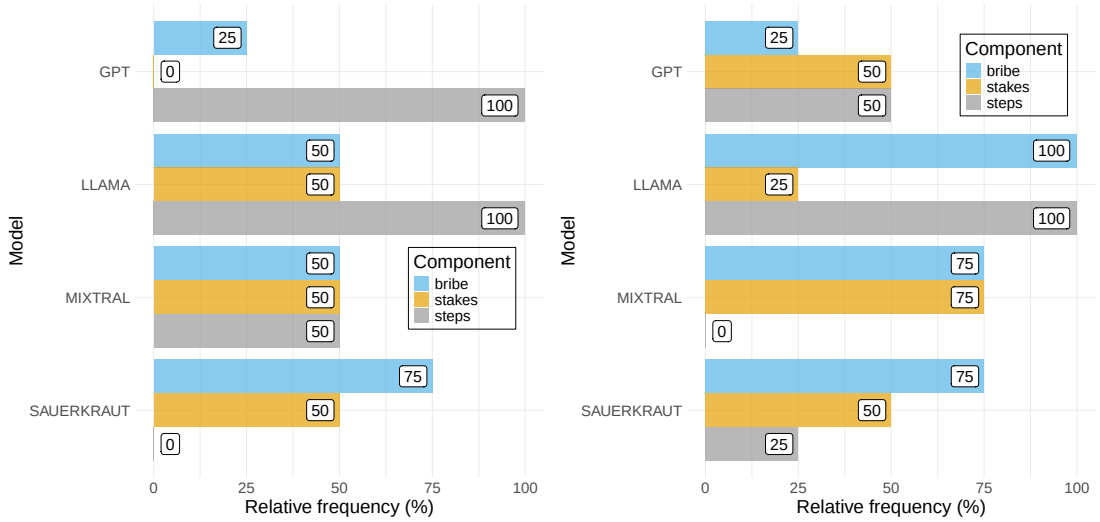
Dividing the test data into two sub-data sets (RQ1) shows a clear tendency: All four models perform in a statistically significant way differently on the two data sets. This is arguably not specific to prompting or large language models, but a general property of machine learning approaches, although we are not aware of work that systematically investigates this. We believe this to be a consequence of how test data is sampled, how much variety of the phenomenon it covers, and, ultimately, how representative the selected test sample is for other test samples or the 'population' in general. In particular the latter question is not easy to answer, given that we are dealing with historical and cultural data, which is subject to a number of highly intransparent selection processes (cf. Levi, 2013). Still, as it has been hinted that large language models "understand" a prompt (Bubeck et al., 2023)[14] (which nobody has claimed for classical machine learning algorithms), it can be argued that if the models would have understood those prompts, they would not show a statistically significant difference on different test data sets.

The fact that different prompts lead to different responses (RQ2) is not surprising per se. What Mizrahi et al. (2024) have uncovered is that meaning-preserving prompt variants (e.g., spelling variation or paraphrases) also lead to different responses, and that – when ranking models for their performance – the exact prompt formulation has tremendous influence on the ranking of such models. They therefore recommend to use the mean performance over multiple prompts. Generally, we also observe a difference in F1 score depending on the exact prompt. While model ranking is not our prime goal here, different model rankings can be established from our experiments as well – which makes the search for the 'best model' for a given task more complex. However, the differences we observe are rather modest, with standard deviations over various prompt variants between 2 and 6 points in F1 score. Still, if the overall absolute performance results were better, a difference in this range could very well have impact on the applicability of such a model in practice. To address specific tasks, there is no alternative to having annotated reference data and experimenting with different formulations and parameters. At the same time, exhaustively searching the best setting is impossible.

Finally, we have investigated recommendations that are often given for manually constructed prompts (RQ3), on what to include in the prompt.

[13] RQ3 has only been evaluated on *test 1*, since it yielded the best average performance scores.

[14] The paper contains sentences such as: "One of the key aspects of GPT-4's intelligence is its generality, the ability to seemingly understand and connect any topic, and to perform tasks that go beyond the typical scope of narrow AI systems." (Bubeck et al., 2023, p. 7)



(a) Emotion (b) Event

Figure 1: RQ 3: Relative frequency of enabled prompt components in the best performing prompts for *test 1*, measured per model and task and across paraphrases.

Our results support these recommendations only partially. First of all, we see different results for different tasks. Across the two tasks discussed in this paper, we can only extract three clear trends: a) LLAMA seems to benefit from using the steps-component (asking the model to think step by step). b) The same component seems to be detrimental for the SAUERKRAUT model. c) SAUERKRAUT, on the other hand, benefits from the bribe-component for both tasks. For all other components and models, no tendency can be discerned.

In general, across the three research questions in this paper, there seems to be a generalizability issue (which is also discussed in recent papers in philosophy of science, cf. Buijsman and Durán, 2024). Generalizing from any scientific experiment to the 'real world' (or, more technically, from lab data to application data) rests on certain assumptions about model behavior and data sets. This applies first to the performance measures that have been achieved on a test set – assuming representativity of the test set, performance will be roughly similar during application. This is, in practice, impossible to control and verify. Secondly, as the actual performance of a model-prompt-pair varies substantially depending on prompt variations, it is impossible to recommend a model or prompt formulation that is *in general* beneficial to the performance results. This holds not only to the formulation variants of a prompt, but, thirdly, also to the selection of prompt components. While there is no reason to believe that the same prompt component will always be beneficial (or detrimental) to the results – properly establishing prompt components that *often* lead to better results would require either a huge project or a number of meta studies that investigate many different existing publications.

Conversely, the scientific use of LLMs and prompting as a ML technique is usually not about general chat functionality (as is a smart personal assistant or "general artificial intelligence"), but about very specific questions and tasks. The general performance of a LLM (measured on some benchmark) may not be indicative for the specific tasks that a researcher from CLS has as their goal. For solving specific tasks, using reference data as train/test data still is the only way to systematically search for the best performing combination of model, prompt and parameters.

7 Conclusions

We were able to show that (i) LLM models are sensitive to data splits (ii) the choice of prompt-model combination determines the success in performance to a high degree and (iii) the helpfulness of fixed components in the prompts to increase performance can not be corroborated for all models for the given tasks. Overall, it could also be shown that all tested models have problems to reach satisfying results on both tasks (emotion and event sequences classification), casting doubt on the immediate usefulness of in-context, zero-shot LLM-sequence-classification for the given CLS tasks.

Limitations

Due to the complexity of the model architectures, which is known to be not publicly available for many models (Liesenfeld et al., 2023), as well as the effort involved in the manual creation of reference data curated by specialists, the present study could not take into account all factors that we believe are relevant for assessing its results. This is not least due to the fact that there is still no generally valid and generic formula for what is ultimately relevant for the results that a specific LLM achieves on specific data. Of the factors that we consider relevant, we were unable to take into account the following in particular: 1) The theory dependency of the evaluation data: In the Digital Humanities in general and CLS in particular, the theoretical orientation determines which concepts are operationalized and how they are subsequently measured. It can be assumed that alternative annotation guidelines that are also plausible from a literary studies perspective can be created for the two tasks we examined. In this respect, the classification tasks evaluated here should be tested on several curated reference data sets in order to check the extent to which different operationalization approaches affect the performance of the models via the detour of the reference data. 2) The statistical representativeness of the data split: this is unclear since we only worked with two test splits, although it is unlikely that different splits on the current data would result in significant difference in performance. 3) The data on which the models were trained: for each task, we only evaluated one dataset with certain choices made that other datasets on the same task might not contain. 4.) the answer-space-mapping: i.e. it is completely unclear if the internal representations of the model that produce natural-language-like output correspond directly to the assumptions that domain specialists have when applying predefined class-labels.

Another limitation that needs to be mentioned is related to the tasks we discuss here: Both of them have clear roots in CLS, although they may not be what is ultimately interesting to a literary scholar. Literary research questions, if they are not on specific interpretations of specific texts, which rules out quantitative approaches a priori, are complex, multi-modal and highly context- and theory-dependent. Addressing such tasks requires the integration of many different analysis components, and we consider the two tasks under investigation to be able to fill the role of two such components. Thus: Both event and emotion detection do not address literary research questions per se, the detection of events and emotions is a relevant ingredient for many, more abstract, literary research questions.

References

AI@Meta. 2024. Llama 3 model card.

Melanie Andresen and Nils Reiter, editors. 2024. *Computational Drama Analysis. Reflecting on Methods and Interpretations*. De Gruyter.

Salvador V. Balkus and Donghui Yan. 2023. Improving short text classification with augmented data using gpt-3. *Natural Language Engineering*, page 1–30.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

David Bamman, Kent K. Chang, Li Lucy, and Naitian Zhou. 2024. On classification with large language models in cultural analytics. Preprint.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. ArXiv:2303.12712 [cs].

Stefan Buijsman and Juan M. Durán. 2024. Epistemic implications of machine learning models in science. In *The Routledge Handbook of Philosophy of Scientific Modeling*, 1 edition, pages 456–468. Routledge, London.

Tanise Ceron, Neele Falk, Ana Barić, Dmitry Nikolaev, and Sebastian Padó. 2024. Beyond prompt brittleness: Evaluating the reliability and consistency of political worldviews in LLMs. ArXiv preprint 2402.17649.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. _eprint: 1803.05457.

Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. 2023. Large language models in the workplace: A case study on prompt engineering for job type classification. In *Natural Language Processing and Information Systems*, volume 13913 of *Lecture Notes in Computer Science*, pages 3–17, Cham. Springer Nature Switzerland.

Evelyn Gius, Svenja Guhr, and Benedikt Adelmann. 2021. d-prose 1870-1920.

Herbert Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3(S 41):58.

Hans Ole Hatzel, Haimo Stiemer, Chris Biemann, and Evelyn Gius. 2023. Machine Learning in Computational Literary Studies. *it - Information Technology*. Read_Status: New Read_Status_Date: 2024-10-18T18:55:15.614Z.

Rebecca M. M. Hicke, Yuri Bizzoni, Pascale Feldkamp, and Ross Deans Kristensen-McLachlan. 2024. Says who? effective zero-shot annotation of focalization.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Evgeny Kim and Roman Klinger. 2018. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Amalia S. Levi. 2013. Humanities 'Big Data'. myths, challenges, and lessons. In *IEEE International Conference on Big Data*.

Andreas Liesenfeld, Alianda Lopez, and Mark Dingemanse. 2023. Opening up chatgpt: Tracking openness, transparency, and accountability in instruction-tuned text generators. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, CUI '23, New York, NY, USA. Association for Computing Machinery.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation.

Janis Pagel, Axel Pichler, and Nils Reiter. 2024. Evaluating in-context learning for computational literary studies: A case study based on the automatic recognition of knowledge transfer in German drama. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 1–10, St. Julians, Malta. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems*.

Axel Pichler and Nils Reiter. 2024. »LLMs for everything?« Potentiale und Probleme der Anwendung von In-Context-Learning für die Computational Literary Studies. In *Book of Abstracts of DHd*.

Steffen Pielström, Fotis Jannidis, Evelyn Gius, Jonas Kuhn, Nils Reiter, Christof Schöch, and Simone Winko. 2023. SPP 2207 Computational Literary Studies (CLS) Projects. Accessed: 2024-10-25.

Andrew Piper. 2020. *Can We Be Wrong? The Problem of Textual Evidence in a Time of Data*, 1 edition. Cambridge University Press.

Nikta Gohari Sadr, Sangmitra Madhusudan, and Ali Emami. 2025. Think or step-by-step? UnZIPping the black box in zero-shot prompts.

Elvis Saravia. 2022. Prompt Engineering Guide. Accessed: 2024-07-09.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage?

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christof Schöch, Julia Dudar, and Evegniia Fileva, editors. 2023. *Survey of Methods in Computational Literary Studies (=CLS INFRA D3.2: Series of Five Short Survey Papers on Methodological Issues)*. CLS INFRA, Trier. With contributions by Joanna Byszuk, Julia Dudar, Evegniia Fileva, Andressa Gomide, Lisanne van Rossum, Christof Schöch, Artjoms Šeļa and Karina van Dalen-Oskam.

Michael Vauth and Evelyn Gius. 2021. Richtlinien für die Annotation narratologischer Ereigniskonzepte.

Michael Vauth and Evelyn Gius. 2022. Event annotations of prose. *Journal of Open Humanities Data*, 8(19):1–6.

Zhiqiang Wang, Yiran Pang, and Yanbin Lin. 2023. Large language models are zero-shot text classifiers.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey.

Biao Zhao, Weiqiang Jin, Javier Del Ser, and Guang Yang. 2023. Chatagri: Exploring potentials of chatgpt on cross-linguistic agricultural text classification. *Neurocomputing*, 557:126708.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of International Conference on Machine Learning 2021 (ICML)*.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

A Dataset Examples

Sentence	Class
For I fear the failing will go with me to the grave that I am very ready to be annoyed, even to the **loss of my temper**, at the urgings of ignoble prudence.	anger
She would brighten up greatly at this, taking it for a compliment **of the best sort**.	anticipation
For I fear the failing will go with me to the grave that I am very ready to be annoyed, even to the loss of my temper, at the urgings of **ignoble prudence**.	disgust
Through all its **tremor**, there was a look of constancy that greatly pleased me.	fear
His **smile** was distinctly attractive.	joy
'Eh,' said the old man, **staring at the floor** and lifting his hands up and down, while his arms rested on the elbows of his chair, 'it's a poor tale if I mun leave th' ould spot an be buried in a strange parish.	sadness
Then she went on with a **sudden outbreak of passion**, a burst of summer thunder in a clear sky:	surprise
"**Not a doubt** of it, my dear.	trust

Table 3: Examples for annotations (bold) in dataset "Emotion".

Sentence	Class
Ich glaube, Mama würde sich freuen, wenn sie wüßte, daß ich so was gesagt habe. "**I think** mom would be happy if she knew I said something like that."	stative
Sidonie nickte. "**Sidonie noded**."	process
Effi, als sie seiner ansichtig wurde, **kam in ein nervöses Zittern**; "**Effi**, when she saw him, **began to tremble nervously**;"	change of state
In drei Tagen feiern wir Sylvester. "**In three days we will celebrate New Year's Eve**."	non event

Table 4: Examples for annotations (bold) in dataset "Event" from the text *Effi Briest*.

B Prompt Templates

```
### Role
You are a literary scholar.

### Instruction
Your task is to classify parts of sentences on the basis of labels given to you.
This should be done in two steps: First, extract the part of the sentence to which one of the three labels applies. Then output this label.

Let's think step by step. <step>
I'm going to tip $1000 for a better solution! <bribe>

### Labels
Select one of the following labels to classify a text excerpt:
        Label: process
        Label: stative_event
        Label: non_event

### Application
When annotating text snippets, the following steps should be taken to determine the appropriate label:
1. **Identify the Main Verb**: Determine the main verb in the sentence or clause to understand the nature of the action or state being described.
2. **Analyze the Context**: Consider the surrounding context to ensure the correct interpretation of the verb and the overall meaning of the snippet.
3. **Assign the Label**:
        - If the text is purely descriptive or provides background information without any action, label it as non_event.
        - If the text describes a state or condition without any dynamic action, label it as stative_event.
        - If the text describes an action or process that involves change or progression, label it as process.

### Output format
Use the following output format:
Part of Sentence to be labeled: str
Label: str

Do NOT generate any more text or repeat the input!
Doing this task well is very important for my career. <stakes>

### What types of event can be found in the following sentence: {snippet}
Part of Sentence to be labeled:
Label:
```

Listing 1: Example prompt (Template 1; Event). The occurrance of the component phrases is annotated in angle brackets.

```
### Role
You are a literary scholar.

### Instruction
Your assignment is to identify and categorize specific segments of sentences according to predefined labels provided to you.
This process involves two steps: First, isolate the relevant portion of the sentence that corresponds to one of the three labels. Then, assign the appropriate label to that portion.

Let's approach this systematically, one step at a time.
I will reward $1000 for anyone who can deliver a more optimal solution.

### Labels
Select one of the following labels to classify a text excerpt:
        Label: process
        Label: stative_event
        Label: non_event

### Application
When annotating text snippets, the following steps should be taken to determine the appropriate label:
1. **Identify the Main Verb**: Determine the main verb in the sentence or clause to understand the nature of the action or state being described.
2. **Analyze the Context**: Consider the surrounding context to ensure the correct interpretation of the verb and the overall meaning of the snippet.
3. **Assign the Label**:
        - If the text is purely descriptive or provides background information without any action, label it as non_event.
        - If the text describes a state or condition without any dynamic action, label it as stative_event.
        - If the text describes an action or process that involves change or progression, label it as process.

### Output format
Use the following output format:
Part of Sentence to be labeled: str
Label: str

Do NOT generate any more text or repeat the input!

### What types of event can be found in the following sentence: {snippet}
Part of Sentence to be labeled:
Label:
```

Listing 2: "Prompt (Template 2; Event; all components)."

```
### Role
You are a literary scholar.

### Instruction
Your objective is to analyze sentences and label specific parts based on the given set of labels.
This task should be completed in two phases: Initially, identify the segment of the sentence that matches one of the three labels. Subsequently, assign the corresponding label to that segment.

Let's break this down into manageable steps.
I'm prepared to give a $1000 tip for a superior solution.

### Labels
Select one of the following labels to classify a text excerpt:

        Label: anger
        Label: joy
        Label: surprise
        Label: sadness
        Label: disgust

### Application
When annotating text snippets, span annotations of key words (e. g., "afraid") should be prefered, except cases when
emotions are only expressed with a phrase (e. g., "tense and frightened") or indirectly (e. g., "the corners of her mouth went down").
Each span is associated with one or more emotion.

### Output format
Use the following output format:
Part of Sentence to be labeled: str
Label: str

Do NOT generate any more text or repeat the input!

### What types of emotion can be found in the follwing text snippet: {snippet}
Part of Sentence to be labeled:
Label:
```

Listing 3: "Prompt (Template 3; Emotion; all components)."

```
### Role
You are a literary scholar.

### Instruction
Your mission is to examine sentences and categorize certain elements using the provided labels.
This should be accomplished in two stages: First, pinpoint the portion of the sentence that aligns with one of the three labels. Then, designate the appropriate label for that portion.

Let's tackle this challenge methodically, step by step.
To encourage a superior answer, I will provide a tip of $1000.

### Labels
Select one of the following labels to classify a text excerpt:

        Label: anger
        Label: joy
        Label: surprise
        Label: sadness
        Label: disgust

### Application
When annotating text snippets, span annotations of key words (e. g., "afraid") should be prefered, except cases when
emotions are only expressed with a phrase (e. g., "tense and frightened") or indirectly (e. g., "the corners of her mouth went down").
Each span is associated with one or more emotion.

### Output format
Use the following output format:
Part of Sentence to be labeled: str
Label: str

Do NOT generate any more text or repeat the input!

### What types of emotion can be found in the follwing text snippet: {snippet}
Part of Sentence to be labeled:
Label:
```

Listing 4: "Prompt (Template 4; Emotion; all components)."

C Full Results

Model	Components			Over 4 variants	
	Bribe	Stakes	Steps	Mean	Std. dev.
GPT	−	−	−	18.27	5.24
	−	−	+	18.58	5.31
	−	+	−	18.01	4.76
	−	+	+	17.15	3.95
	+	−	−	17.74	4.65
	+	−	+	17.5	4.83
	+	+	−	17.67	4.3
	+	+	+	17.42	4.58
	Mean			17.79	4.7
LLAMA	−	−	−	14.47	2
	−	−	+	15.41	3.04
	−	+	−	14.63	2.21
	−	+	+	15.16	2.53
	+	−	−	14.15	1.96
	+	−	+	15.27	2.46
	+	+	−	14.41	2.54
	+	+	+	15.19	2.47
	Mean			14.84	2.4
MIXTRAL	−	−	−	16.34	3.44
	−	−	+	16.85	3.93
	−	+	−	16.74	3.54
	−	+	+	16.52	3.35
	+	−	−	16.92	4.14
	+	−	+	16.33	3.54
	+	+	−	16.89	3.86
	+	+	+	16.51	3.46
	Mean			16.64	3.66
SAUERKRAUT	−	−	−	16.0	2.69
	−	−	+	15.53	2.69
	−	+	−	15.74	2.48
	−	+	+	15.87	2.88
	+	−	−	15.97	3.23
	+	−	+	15.3	2.43
	+	+	−	16.16	3.29
	+	+	+	16.12	3.47
	Mean			15.84	2.9

Table 5: RQ 2: Robustness against prompt variations (emotion task)

Model	Components			Over 4 variants	
	Bribe	Stakes	Steps	Mean	Std. dev.
GPT	−	−	−	21.86	3.9
	−	−	+	22.03	3.69
	−	+	−	21.45	4.08
	−	+	+	21.85	3.91
	+	−	−	21.29	3.88
	+	−	+	21.34	3.8
	+	+	−	21.48	3.9
	+	+	+	21.01	3.3
	Mean			21.54	3.81
LLAMA	−	−	−	20.22	3.05
	−	−	+	20.17	4.46
	−	+	−	19.47	3.14
	−	+	+	20.59	4.56
	+	−	−	19.71	3.28
	+	−	+	22.0	4.79
	+	+	−	19.71	3.67
	+	+	+	21.43	4.39
	Mean			20.41	3.92
MIXTRAL	−	−	−	24.3	6.08
	−	−	+	23.98	5.72
	−	+	−	23.68	5.54
	−	+	+	24.18	5.84
	+	−	−	24.07	5.9
	+	−	+	23.8	5.86
	+	+	−	24.51	6.53
	+	+	+	23.84	5.88
	Mean			24.05	5.92
SAUERKRAUT	−	−	−	22.19	3.68
	−	−	+	21.9	3
	−	+	−	22.36	3.61
	−	+	+	22.46	3.79
	+	−	−	22.63	4.04
	+	−	+	22.04	3.01
	+	+	−	22.94	3.59
	+	+	+	22.6	3.25
	Mean			22.39	3.5

Table 6: RQ 2: Robustness against prompt variations (event task)