Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Navigating the Prompt Space: Improving LLM Classification of Social Science Texts Through Prompt Engineering

Erkan Gunes

eguenes@constructor.university

Constructor University

Christoffer Florczak

Aalborg University

Tevfik Murat Yildirim

University of Stavanger

Research Article

Keywords: large language models, text classification, prompt engineering, text as data

Posted Date: July 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10237932/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Navigating the Prompt Space: Improving LLM Classification of Social Science Texts Through Prompt Engineering

Erkan Gunes[1], Christoffer Florczak[2], and Tevfik Murat Yildirim[3]

[1]Constructor University, eguenes@constructor.university

[2]Aalborg University, ckfl@society.aau.dk

[3]University of Stavanger, murat.yildirim@uis.no

ABSTRACT

Recent developments in text classification using Large Language Models (LLMs) in the social sciences suggest that costs can be cut significantly, while performance can sometimes rival existing computational methods. However, with a wide variance in performance in current tests, we move to the question of how to maximize performance. In this paper, we focus on prompt context as a possible avenue for increasing accuracy by systematically varying three aspects of prompt engineering: label descriptions, instructional nudges, and few shot examples. Across two different examples, our tests illustrate that a minimal increase in prompt context yields the highest increase in performance, while further increases in context only tend to yield marginal performance increases thereafter. Alarmingly, increasing prompt context sometimes decreases accuracy. Furthermore, our tests suggest substantial heterogeneity across models, tasks, and batch size, underlining the need for individual validation of each LLM coding task rather than reliance on general rules.

Keywords

large language models, text classification, prompt engineering, text as data

1. Introduction

Text annotation using instruction-tuned large language models (LLMs) has emerged as an alternative to training custom supervised models for text annotation in social science research. While findings on accuracy were initially mixed [15, 26], there is now

growing evidence that for some tasks, they could offer satisfactory accuracy when used out-of-the-box with minimal context provided by the user, and therefore have significant potential to replace existing computational tools when fine-tuned for a specific task or with more sophisticated prompting or context-building strategies [7, 12, 16]. As base accuracy has been established against benchmarks, the cost cutting potential is clear, and given that models are likely to improve in the future, the questions around LLMs for text annotation should change nature. Rather than exploring if they are valuable text annotators, the social science community could benefit by exploring how to optimize LLMs for text annotation. Focusing on one avenue, namely prompt engineering, we seek to answer this question by providing an initial test of how researcher input might affect LLM text classification performance.

Prompt engineering is an obvious first avenue to test optimization, as there is large control for the individual researcher, ample opportunity to experiment as the costs are usually low, and one cannot avoid prompting the model, thus making it a necessary step for all researchers. This is, however, complicated by the fact that there are almost an infinite number of possibilities for prompt configurations that can be submitted to a model. To narrow this down, we focus on three areas of prompting. First, we analyze the depth of candidate label description, where we vary if the labels are only given to the model, or if the labels are also explained to the model. Second, we experiment with instructional nudges, where we vary if small notes clarifying some nuances about the classification task and setting expectations about LLM output are included in the prompt. Third, we vary whether the model is presented with few-shot examples, or whether the model is given no examples. Additionally, we also vary the size of the input text batch and the model used to further evaluate how prompt engineering interacts with other specifications to affect LLMs' label prediction. We thus provide a fairly comprehensive set of tests on common prompt engineering dilemmas faced by social scientists.

We focus on two classification tasks relevant to social science research and present them as two separate studies in this work. The first study draws on a political science specific task and involves classification of congressional bill titles into 21 policy issue topic categories. The second study has relevance to many fields such as psychology

and sociology, as well as political science, and involves the classification of open-ended survey responses into emotional vs. neutral categories. Results from our experiments spanning all combinations of the above-mentioned context components suggest that researchers can achieve substantial accuracy gains when they go beyond the most minimalist configuration, but there is not a uniform pattern of improvement, e.g. increasing relevant contextual information within the prompt does not linearly increase accuracy. We also find that presenting a single input text within a prompt is usually the suboptimal strategy in terms of performance as well as the more costly option, thus suggesting that batch classification will be the superior choice in many cases. Additionally, our results from the emotionality classification provide evidence that deterministic LLM outputs are not guaranteed even when the temperature parameter is set to zero, and inter-request variation with the exact same prompt content depends on model and prompt attributes. Taken together, our results highlight how performance varies across prompt configuration, task, and model, and there is therefore a need for individual case specific validation tests when performing text annotation using LLMs if researchers want to harness the efficiency of LLM based text classification.

This article proceeds as follows. In the second section, we review recent research on text annotation with LLMs and discuss the shifting focus from model benchmarking to navigating the prompt space for understanding and improving model behavior. In the third section, we introduce the experimental setup of our two studies, and present the results from our prompt engineering experiments. In the fourth section, we discuss what we can learn from our results about understanding and improving LLMs text annotation performance. In the fifth section, we conclude with some general lessons we can draw from our two studies and discuss future research.

2. Related Works

Scholarly interest in the use of instruction-tuned LLMs for various scientific research tasks has increased sharply since the release of ChatGPT by OpenAI. In social science research, this has been especially evident in computational text annotation literature. Until recently, this branch of computational social science was dominated by efforts to

develop custom supervised models for specific tasks. The flexibility, the ease-of-use, and the promising accuracy performance of the latest instruction-tuned LLMs has shifted the field's attention away from training and fine-tuning models toward text annotation using in-context learning methods enabled by instruction-tuned LLMs.

This shift, in turn, motivated an initial wave of benchmarking studies that compared zero-shot LLM-based annotation to established baselines, including supervised models and human annotators. The resulting evidence from this initial wave of studies helped clarify if LLMs could replace human coders and the more conventional computational tools and methods, or if they could be a collaborator to humans. Gilardi et al. [10] showed GPT 3.5 outperforms crowd workers on several text classification tasks such as stance detection and frame detection, and does it much more cheaply. Similarly, Törnberg [22] experimented with GPT 4 on a political affiliation detection task and found that it overperforms both supervised models and human coders. That work also highlights groundbreaking implications of LLMs for the future of computational text annotation in social science research. Experimenting with a larger number of models on a more complicated multiclass text classification task, which involves classification of congressional bill titles and descriptions of congressional hearings into policy issue categories, Gunes and Florczak [11] reported LLMs can sometimes rival human coders and supervised models. They also found that best performance occurs when humans and LLMs collaborate in a way where LLMs label cases only when multiple LLMs agree, and humans label cases when the LLMs disagree.

While this first wave of studies hinted that annotation performance could be sensitive to prompt design choices, most research did not systematically explore the prompt space to identify optimal prompt configurations or to develop procedures for analyzing and reporting prompt-related sensitivities. With growing awareness that annotation quality depends on prompt design choices [2], especially the size and structure of contextual information included in the prompt and the reasoning strategies the model is instructed to use, recent research on text annotation with LLMs has started focusing more on navigating the prompt space than on comparing the capabilities of a set of models to humans and conventional computational tools.

LLMs learn from contextual information presented in the prompt [8]. A central

problem in research on prompt engineering for text annotation is how to improve the quality of LLM output by providing the right contextual information. One contextual component researchers often manipulate in prompt engineering experiments is few-shot examples. Providing a few examples for each candidate label in a classification task can often help boost accuracy. Bosley [7] demonstrates that, as the number of in-context examples increases, LLMs can approach expert-level fidelity on an intricate multidimensional coding scheme, the Discourse Quality Index for deliberative debates. Performance improved markedly when moving from zero-shot to a handful of examples, and then plateaued around 25-50 examples, suggesting diminishing returns beyond a certain context length. Liu and Shi [16] also emphasize more systematic prompt optimization pipelines that go beyond manually selecting exemplars. They introduce PoliPrompt, a three-stage framework that iteratively refines prompts, dynamically selects relevant in-context examples, and uses a consensus mechanism to reconcile outputs across prompts or models. They report sizable gains in classification performance alongside large reductions in annotation costs.

Getting the number and the kind of few-shot examples right is a central objective in more generic prompt optimization frameworks like DSPy [14], as well. In DSPy, users tell the model what the inputs and outputs are for their task, choose additional parameters such as an optimizer, e.g., MIPRO [18] or GEPA [1], and an evaluation metric, and then run DSPy to search for a high performing prompt configuration under that metric. In a text classification task, the input is a short natural language text or document, and the output is a label describing the category the text belongs to. DSPy requires a dataset with ground truth labels, which it uses to compare different possible prompt configurations and find the best performing one on the user-defined input-output formulation. While DSPy offers a wide range of controllable modules, including mechanisms for selecting or synthesizing few-shot examples, its relatively steep learning curve and higher technical barriers to entry may make it impractical for many social scientists. In such cases, researchers may seek greater manual control over prompt content and prioritize domain expertise rather than relying on more automated prompt optimization to determine what information is presented to the model.

Beyond few-shot prompting, researchers are developing systematic methods to

discover optimal prompting strategies. Reich et al. [19] propose the HALC pipeline as a general and codebook-driven approach for automated coding in social science research. HALC allows integrating various prompting strategies (e.g. role prompting, chain-of-thought prompting, context information) and evaluates them empirically at scale. They provide results from experiments where they tried over 1500 candidate prompts in more than two million requests to a local LLM, where the results highlight the importance of high-quality ground-truth data, the choice of prompting strategy, and using a majority decision procedure with multiple LLMs.

While the focus still is largely on highly automated frameworks and pipelines for text annotation, Than et al. [21] explore using generative LLMs in a more interactive manner. They argue that LLMs, prompted in natural language, can mimic the inductive and iterative process of human qualitative analysis to some extent. Their experiments with multiple models and prompt styles showed that LLMs can replicate many of the outcomes of traditional hand-coding, coming close to the accuracy of human-coded themes and even matching prior machine learning baselines. More intriguingly, they highlight how a researcher can engage in a back-and-forth dialogue with an LLM, such as asking the model to explain why a text was labeled a certain way, or to suggest new labels. This points to an emerging paradigm of AI-in-the-loop qualitative research, where the LLM is not just an automated coder but also a conversational partner that can help analysts explore data.

The emerging prompt engineering in computational text annotation literature has already identified various effective prompt components and prompt building strategies. In this study, we contribute to the exploration of the vast prompt space by shifting attention to some structural prompt design elements that can help improve accuracy performance of LLMs and to some sensitivity issues that have not been paid enough systematical attention. In the next section, we introduce these elements and the experimental setup in greater detail.

3. Experiments on Context-Building for Text Classification Tasks

In this work, we deal with two different classification problems, which we present as two separate studies. The first study is a multiclass classification problem where congressional bill titles are classified into policy issue topics. We use the Congressional Bills dataset [23] from the Comparative Agendas Project (CAP). The dataset contains human coded congressional bill titles. The CAP topic scheme offers a comprehensive set of policy issue topic classes, where there are 21 unique classes. Classification of policy documents into the CAP topics is a challenging multiclass classification problem given the large number of classes and vague semantic boundaries between some classes. The second study focuses on a binary classification problem where we work on emotionality detection in open-ended survey responses from American National Election Studies surveys [24]. A statement belongs to the "emotional" class if it contains any emotional expression, or it belongs to the "neutral" class if it features no emotional expression.

Our primary goal in this work is not comparing classification capabilities of specific LLMs, but to explore a wide range of prompt design configurations and consequences on classification accuracy and reproducibility. However, to avoid overgeneralization about LLM behavior under the configurations we test and to find out if cross-configuration patterns we observe are model dependent, we implemented our experiments with more than one LLM. In both studies, we use two different instruction-tuned LLMs. We use GPT 4o from OpenAI and Gemini 2.0 Flash from Google, which are relatively fast and cheap models with high scores across many performance benchmarks.

In the following subsections, we present details about datasets, prompt configuration components and the results from experiments for both studies.

3.1. Study 1: Classifying Congressional Bill Titles into Policy Issue Topics

A large body of work within the literature on policy agendas focuses on attention allocation dynamics in policy making processes [4, 13]. Identifying attention allocation distribution across issue categories at a given time and policy venue is essential for describing the dynamics of attention allocation in policy making processes and investigating the implications for policy outputs. Researchers have long been processing

various policy documents and manually classifying them into a policy issue category. CAP offers the most widely used topic scheme for policy issue classification with more than 21 major issue topics and numerous issue subtopics under each major topic [5]. There has been a major effort to automate issue topic classification as large advances in natural language processing have revolutionized how researchers can process unstructured text data [3, 20]. More recently, there have been efforts to automate issue topic classification using instruction-tuned LLMs [11], where the results showed promising accuracy performance rivaling custom-trained supervised learning models. In this first study, we explore further how we can get better classification performance from instruction-tuned LLMs when they are used for a highly complex classification problem as the policy issue classification.

In our policy issue classification experiments, we use the prompt structure depicted in Figure 1. The three small boxes in the center represent the contextual information components we systematically vary in the experiments. The left box is the "label description" component. When this component is present in a prompt configuration, we present candidate labels along with a description of the kinds of policy issues that belong to that CAP major topic category. When this component is absent in a prompt configuration, we only include the list of candidate labels without any description. Label descriptions come from the master codebook of the CAP. In the congressional bills dataset, there is an additional class, named "private bills" which does not exist in the default CAP scheme. Private bills refer to proposals that are proposed to solve a problem concerning a private individual, typically addressing a unique situation not covered by public laws. The label description for that class comes from the official website of the U.S. Senate.[1] The box in the middle of the center part is the "instructional nudges" component. These are notes that serve as instructional nudges, and they are informed by domain expertise and small-scale trials with models. For example, abortion related bill titles belong to the civil rights category in the CAP codebook, but the models we used were often assigning them the "health" label. We included a note to explain how abortion related titles must be handled by the model. Such notes focus on instructions that are very specific in nature, and they provide extra context which

[1] https://www.senate.gov/legislative/common/briefing/leg_laws_acts.htm

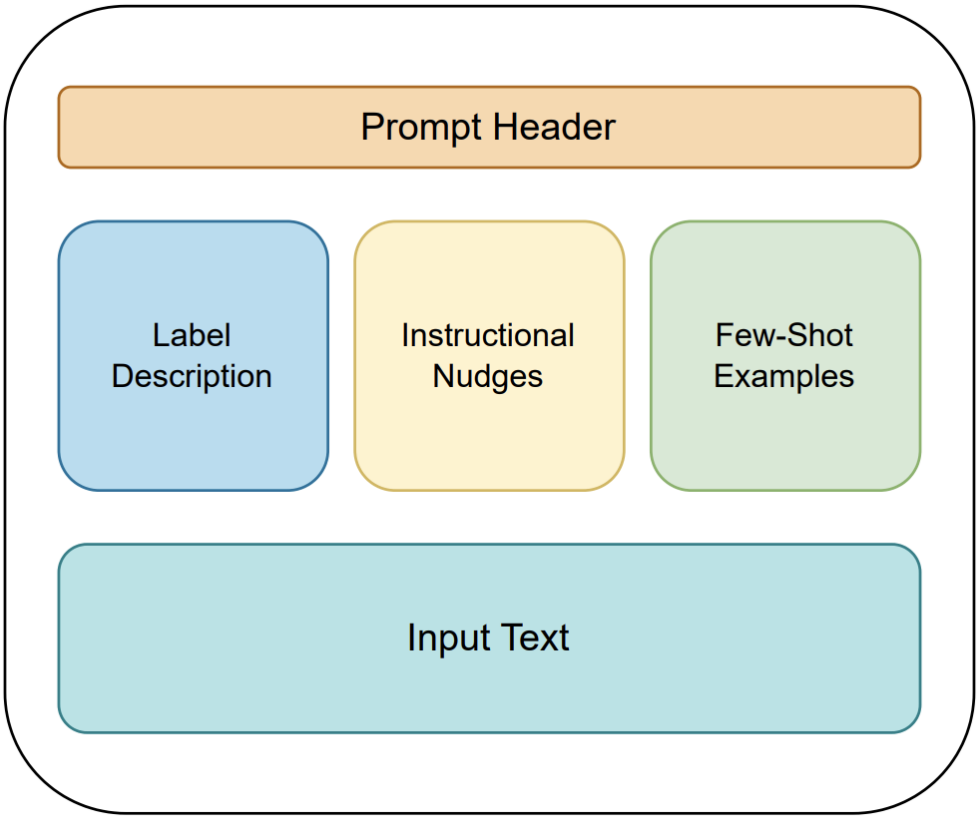


Figure 1. Prompt components that vary across experimental configurations

cannot be presented in the "label descriptions" or "few-shot examples" parts. When this component is present in a prompt configuration, we show such instructions in the middle of the prompt, and we show no such instructions when it is absent. The right box in the center is the "few-shot examples" component, which shows the LLM model a few congressional bill title examples for each unique class.

The examples come from the part of the full Congressional Bills dataset which did not end up in the random sample we used in the experiments. When this component is absent in a prompt configuration, we do not show any example titles for any major topic category. The bottom component in Figure 1 is the input text component, which is congressional bill titles in our case. Batch size is the dimension that varies in this component of the prompt. We use batch sizes of 1, 10, 100, 500, and 1000. When the batch size is 1, we show a single bill title to be classified. When it is greater than 1, we show multiple bill titles in a single prompt and ask the LLM model to return a

list of labels generated in a format specified within the prompt. We show full prompt examples, where all the context components are present and have a batch size of 1, in Appendix 1.

We present the results from our prompt engineering experiments with the congressional bills data in Figure 2 and Figure 3. They show weighted F1 scores for all possible configurations of the contextual information components described above, five different input text batch sizes, and two different LLM models. Each vertical section separated by dashed lines correspond to a different combination of the three components, which are respectively "label descriptions", "instructional nudges", and "few-shot examples". The plus and minus signs in the x-axis labels show whether a component is present or absent in a configuration. For example, in the configuration denoted by (+,-,+), "instructional nudges" component does not exist, but the other two components are present.

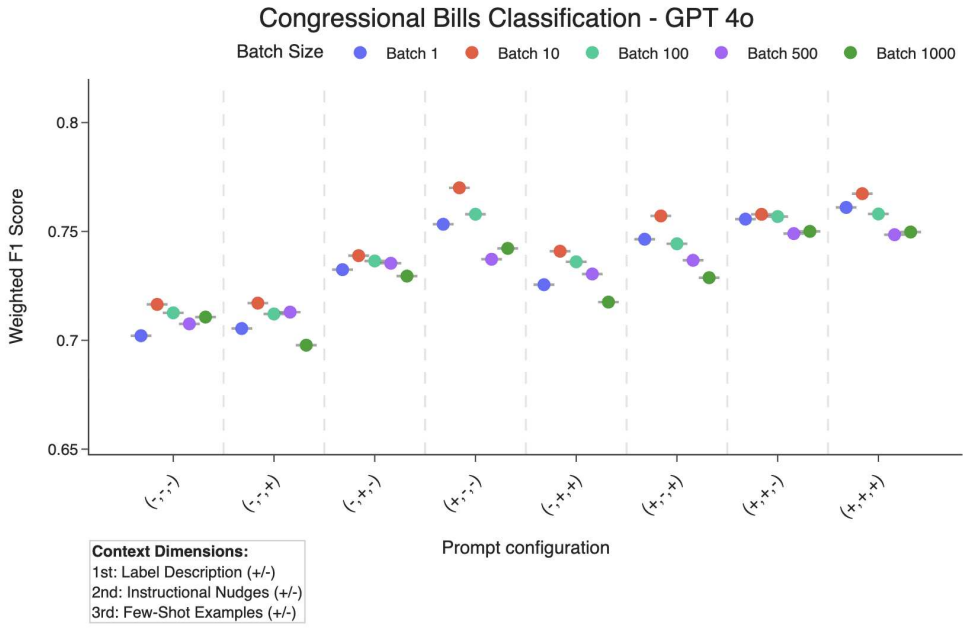


Figure 2. Issue topic classification performance of GPT 4o across different contextual information configurations and input text batch sizes

Accuracy performance of the two models across the numerous configurations presented in Figure 2 and Figure 3 exhibit a positive correlation with the complexity of the contextual information configuration. The leftmost configuration is the most minimal configuration where none of the three contextual information components are

present. The next three configurations to the right of the most minimal configuration feature only one of the three components, and the next three configurations to the right of them feature two of the three components. The rightmost configuration has all three components.

For each contextual information configuration, we present results with five different input text bath sizes. We tried a maximum batch size of 1000, where we started to observe significant decline in performance, especially with the GPT 4o model. Batch size 1 is what most researchers use when they use an instructed-tuned LLM for a task annotation task, but inserting multiple input texts within a single prompt might improve cost efficiency, but also enable some performance gains. The results from experiments with GPT 4o and Gemini 2.0 Flash indicate that presenting a single input text to annotate in a single LLM call will not give the most accurate output in more than half of the configurations.

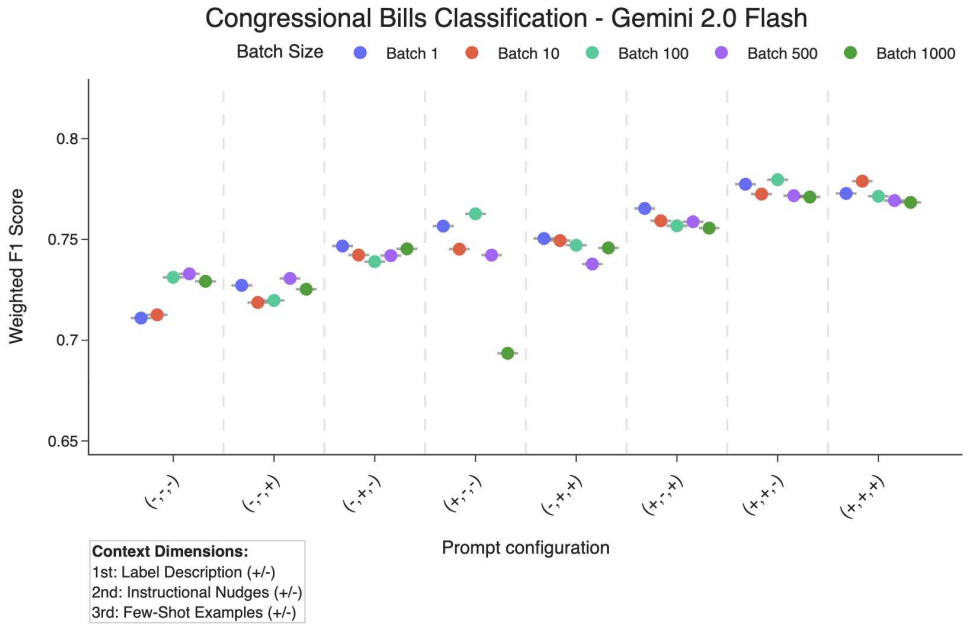


Figure 3. Issue topic classification performance of Gemini 2.0 Flash across different contextual information configurations and input text batch sizes

3.2. Study 2 : Emotionality Detection in Open-Ended Survey Responses

We now turn to our analysis of open-ended survey responses from the American National Election Studies (ANES) data. Before delving into emotionality in open-ended responses, it is important to note that the responses analyzed here were labelled after the cutoff dates of the language models used in this study, which ensures that these data could not have been part of the models' training data and therefore potential concerns about data leakage or memorization. A large strand of research in political science has examined answers to the most important problem question in surveys, and their relevance for various micro- and macro-political phenomena. However, this literature almost exclusively focused on the frequency of MIP mentions. While the content of responses to "most important problem" (MIP) questions reveals public policy priorities, the emotional tone and affective language used by respondents (such as anger toward politicians, fear of economic collapse or war, or frustration with institutions) provide crucial insights into the intensity and valence of salience, shaping political mobilization and responsiveness beyond mere topic mention. Analyzing affective language in open-ended MIP answers uncovers how emotions amplify volatility (e.g., scandal-driven spikes) and signal deeper distrust, offering a richer measure of agenda dynamics than frequency coding alone. This dual focus on "what" and "how" enhances validity for studying anti-politics and issue thermostats.

In an attempt to demonstrate the utility of this approach, we present an illustrative case on the emotion-coding of open-ended answers to the MIP question in ANES surveys. For this illustrative case, we downloaded the open-ended answers to the 'most important problem' question that has been continuously asked in ANES surveys over the past decades.[2] Respondents' verbatim answers were recorded by interviewees, and, after any identifying information that might reveal the respondent's identity was removed, verbatim answers were posted on the ANES website. Following Ekman's [9] framework of six basic emotions, namely, anger, disgust, fear, joy, sadness, and surprise, we coded these answers as emotional if they contained a word or phrase which was emotionally loaded (e.g., "love", "feel threatened", "hate"). Responses were coded

[2]The question takes the following form: "What is the most important problem facing the country?" Although slightly different versions of the question have been asked by various survey organizations since 1939, the question asked by ANES has followed the same structure across decades.

as neutral if they described a problem without any expression of emotion, typically involving factual statements.

Below we provide two example answers for the emotional and neutral answer categories, the first being coded as emotional whereas the second as neutral:

> "They should give the government back to the people. This country was supposed to be by the people, for the people. They should follow the Constitution. They're all liers, cheats, crooks don't care what the people want or say. They take away from us what we have worked hard for and earned get rid of the IRS and that bank that runs everything the federal bank that has 13 branches and runs everything. Stop messing with people let them farm or do whatever they want like we used to do without interference. No."

> "The national debt. The money we owe out to other countries is tremendous. One time we have multi-trillion deficit in this country. We've loaned out money to other countries and they never paid us back."

In this illustrative case, we use the same prompt structure depicted in Figure 1, although the implementation of its individual components differs slightly from that

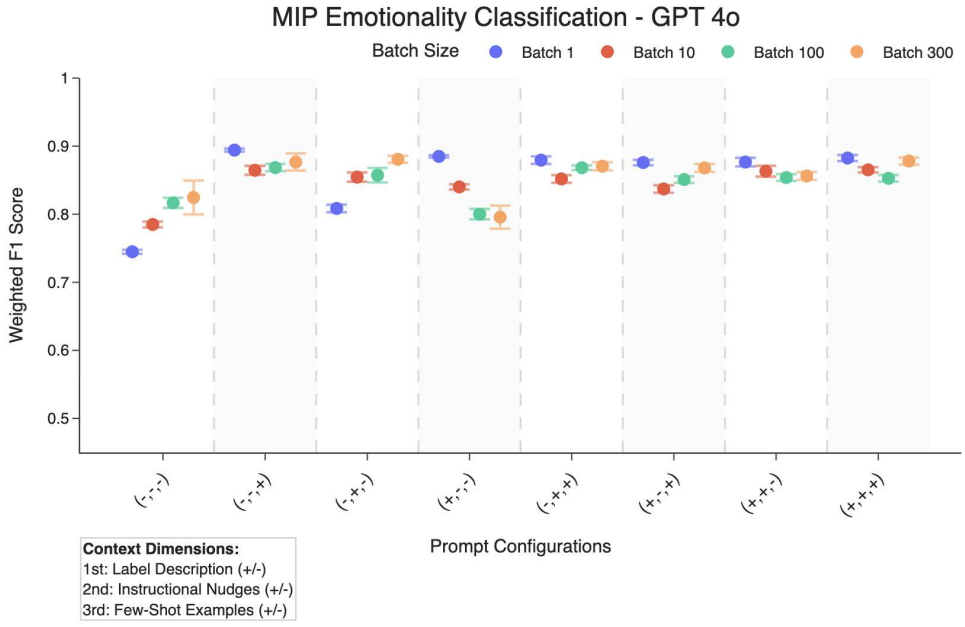


Figure 4. Emotionality classification performance of GPT 4o across different contextual information configurations and input text batch sizes

in Section 3.1, given the nature of the task. In this application, we wrote the label descriptions rather than drawing from an existing codebook, as no established coding scheme exists for emotional and affective content in MIP responses. The instructional nudges consist of a single statement encouraging the model to approximate an expert-informed label distribution (see Appendix A), which reflects our substantive judgments about the relative prevalence of categories. The few-shot examples included in the prompt were selected separately from the randomly drawn analysis sample to avoid overlap. Finally, due to the smaller size of the MIP subsample, the maximum batch size used in this application was lower than in the bills study.

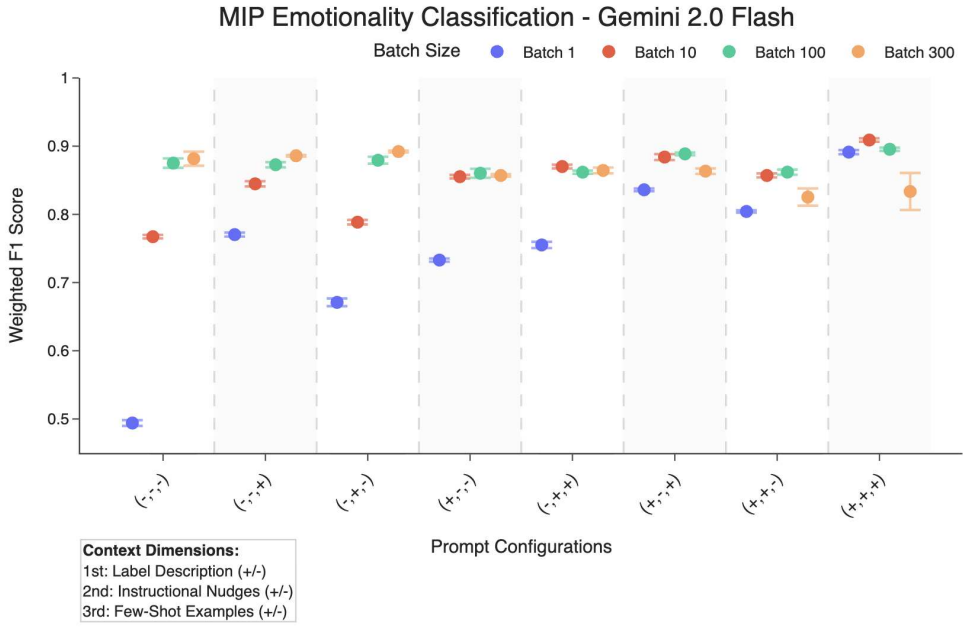


Figure 5. Emotionality classification performance of Gemini 2.0 Flash across different contextual information configurations and input text batch sizes

In Figure 4 and Figure 5, we present the weighted F1 scores for MIP emotionality classification across different prompt configurations and batch sizes for GPT-4o and Gemini 2.0 Flash. For each prompt configuration, results are reported for four batch sizes (1, 10, 100, and 300), as was done in the previous section. Across both models, performance varies across prompt configurations defined by the inclusion or exclusion of label descriptions, instructional nudges, and few-shot examples. For GPT-4o, weighted F1 scores generally range between approximately 0.75 and 0.90 across configurations

and batch sizes. For Gemini 2.0 Flash, performance spans a wider range, particularly at batch size 1, with higher batch sizes displaying relatively little variance across configurations. In both models, configurations that include additional contextual components tend to be associated with higher and more stable F1 scores across batch sizes.

4. Discussion

In this paper we presented two tests of LLM performance on classification tasks across varying configurations of context prompting and batch sizes. We aimed at opening the black box of LLM performance by systematically varying the input to assess the relative importance of context prompting elements and batch sizes, as well as testing how robust the patterns of performance were across multi and dual classification tasks and the possible inclusion of the data in the model's training.

Several patterns of interest emerge. First, across all tests we see that prompting context matters in a pattern where prompt engineering tends to increase the F1-score. In much early work on text annotation using LLMs, researchers have been making use of minimalists prompts, which are likely a suboptimal strategy, and should be avoided in future research relying on LLMs. However, the difference between light prompt engineering and more heavy-handed prompt engineering (e.g making use of more strategies simultaneously) is oftentimes rather small, suggesting that the majority of the effect is achieved rather rapidly and then tapers off. It should be noted that while the performance gain in the later stages of prompt engineering is relatively modest, it is also practically easy to implement and monetarily cheap and thus most researchers should not feel limited in employing these techniques and reaping the advantages. It must be noted, that while we test a few strategies here, more are likely to emerge and we are unable to assess the effectiveness of adding these additional strategies along with ours based on current results.

There is also future research to be done in exploring the finer details of the individual prompt engineering strategies we employ here. For instance, future research could examine how many examples are needed to stabilize the results of few-shot prompting,

and how this varies by models. Hopefully, as the field keeps advancing and models improve [11] many heterogeneities may slowly disappear and thus make this type of prompting strategy less important. However, the current heterogeneity may also be sparking the different recommendations currently found in the literature, where some authors actively discourage the use of LLMs for text annotation [15]. We show fairly good results on our tests, and therefore believe that LLMs can be used, but that it needs to be evaluated on a case-by-case basis.

We also observe heterogeneity between our two tasks and datasets. There can be several reasons for that. It could be the case that the labeling of emotion from text is a type of problem wherein the notion of a gold standard is itself challenged [6] as is also the case with ground truth measurements [10, 22]. In any case, while many studies solely look at one type of texts to classify, the value added from comparing our rather diverse sets of texts and tasks is clear: it cannot be assumed that different tasks and texts follow a similar pattern in performance measures or engineering, and it cannot be assumed that performance measures necessarily decrease with the complexity of the task. While it is generally the case, performance is also dependent on the configuration of prompt engineering, batch size and model, as well as task and text type, and we therefore strongly encourage all users of LLM's for text annotation tasks to conduct, report and evaluate validation efforts for their specific tasks rather than solely relying on existing research as validation. Therefore, documentation of the prompting strategy and the trial and error approach in each individual task is paramount, and we suggest that this should be included in appendix material. This is important for two reasons. First, our large-scale tests revealed that the temperature setting on LLMs does not necessarily guarantee deterministic (e.g. fully replicable) results even at a setting of 0 (see also Yuan et al. [25]). Second, we have instances of sudden, rather substantial, changes in classification performance following minor changes in prompting structure. This brittleness in model performance is an essential factor to take into account, and it can only really be discovered through multiple trials. Thus, while LLMs as text classification tools are clearly useful if the configuration is well chosen, researchers should be aware of the rather large range of possible outcomes if the configuration is not – and that eliminating all randomness in output is likely not feasible given the

current models. This may be mitigated through local models, where deterministic parameters can be controlled.

As algorithms continue to improve it is also likely that the effects of prompt engineering may change. However, emerging research on the usefulness of LLM approaches in the face of rapid technological development paints varying pictures of how that may affect the trustworthiness of current results. For instance, Morucci and Spirling [17] argue that the data used in political science is typically of a low dimensionality and that adding more complexity in terms of additional variables rarely provides us with much more accurate analyses than the usual low dimensionality structure. As such, the more advanced models will often be unable to significantly outperform the current ones. Should the same logic apply to LLM based classification, it may be the case that newer models may not exhibit significantly superior performance compared to those that are already available. However, the high degree of dimensionality in text data specifically may mean that improvements can still be made through more advanced methods in this particular use-case.

Future research should investigate the possibility of diminishing returns to prompt optimization in newer models compared to older. It may be the case that as models become better, their ability to accurately interpret varying instructions may also improve, and that this decreases the usefulness of prompt engineering. Procedures to automatically generate prompts is another research avenue that underscores that manual prompt-engineering is itself contested as an approach to optimizing LLM output. Iteratively generating prompt structures to maximize coding output by a fixed pipeline structure such as through DSPy [14] would likely improve the output while also keeping the costs of applying LLMs low. While the advantages are clear, this approach still implicitly accepts a black box approach to generating prompts, which should be kept in mind.

5. Conclusion

This work offers a structured approach to building context within a text classification prompt and to systematically testing how different contextual information components

affect LLMs' accuracy and reproducibility in text classification tasks. Our experiments with GPT 4o and Gemini 2.0 Flash, spanning all possible combinations of the three contextual information components, showed that researchers should avoid prompt designs which contain only the minimum information needed for task execution, and should instead include richer contextual information along task-relevant dimensions. Our findings show large accuracy gains even when a single contextual information component is included in a text classification prompt, but accuracy does not always increase monotonically as the prompt becomes more complex with the addition of more than one contextual component. We also present evidence for the non-determinism of LLM responses even when model parameters are set to produce deterministic outputs, and our results show the extent of variability across different LLM requests with the exact same configuration also depends on prompt complexity.

In this work, we focused on a small number of models and contextual information components, and explored a limited range of variation along each of the tested components. Future research should explore our prompt engineering and experimentation strategy with a larger set of models, and especially with open source models which offer researchers greater control over data and conditions that affect reproducibility. We also need greater attention from researchers to experiment with different sizes of few-shot examples and different example selection strategies. By pursuing these lines of inquiry, researchers can better understand when context helps, when it saturates, and how to make LLM-based classification more dependable.

Supplementary information

Supplementary materials file can be accessed at doi.org/10.7910/DVN/UENFMA

Declarations

Availability of Data and Material

Data and code files are available at doi.org/10.7910/DVN/UENFMA

Competing Interests

The authors declare no competing interests.

Funding

This work was supported by the European City Squared project, funded by the European Union's Horizon Europe research and innovation programme under grant agreement DOI: 10.3030/101178170.

Authors' Contributions

E.G. developed experimental design, executed experiments, annotated data for study 2, processed responses from LLMs, prepared figures and tables, prepared supplementary material file, and contributed to the writing of several sections in the main text. C.F. helped refine experimental design and the presentation of the results. He also wrote some sections of the main text and helped refine other sections. T.M.Y annotated data for study 2, wrote some sections of the main text, and helped refine other sections.

subsection*Acknowledgements Not applicable

References

[1] Agrawal LA, Tan S, Soylu D, Ziems N, Khare R, Opsahl-Ong K, Singhvi A, Shandilya H, Ryan MJ, Jiang M, Potts C, Sen K, Dimakis AG, Stoica I, Klein D, Zaharia M, Khattab O (2025) GEPA: reflective prompt evolution can outperform reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2507.19457

[2] Atreja S, Ashkinaze J, Li L, Mendelsohn J, Hemphill L (2025) What's in a prompt? A large-scale experiment to assess the impact of prompt design on the compliance and accuracy of LLM-generated text annotations. Proc Int AAAI Conf Web Soc Media

[3] Baumgartner F, Bevan S, Sebők M (2026) Agenda-setting studies in public policy: origins, development, and new possibilities for coding in the age of AI. Commun Change 2. https://doi.org/10.1007/s44382-026-00021-8

[4] Baumgartner FR, Jones BD (2015) The politics of information: problem definition and the course of public policy in America. University of Chicago Press, Chicago

[5] Baumgartner FR, Jones BD, Mortensen PB (2018) Punctuated equilibrium theory: explaining stability and change in public policymaking. In: Weible CM, Sabatier PA (eds) Theories of the policy process, 4th edn. Routledge, New York, pp 55–101

[6] Bisbee J, Spirling A (2025) What to do when humans are no longer the gold standard: large language models, state of the art and robustness

[7] Bosley M (2025) Towards qualitative measurement at scale: a prompt-engineering framework for large-scale analysis of deliberative quality in parliamentary debates. J Polit Inst Polit Econ 6(3–4):355–383. https://doi.org/10.1561/113.00000128

[8] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al (2020) Language models are few-shot learners. NeurIPS. https://doi.org/10.48550/arXiv.2005.14165

[9] Ekman P (1992) An argument for basic emotions. Cognition and Emotion 6(3–4):169–200

[10] Gilardi F, Alizadeh M, Kubli M (2023) ChatGPT outperforms crowd workers for text-annotation tasks. Proc Natl Acad Sci 120(30):e2305016120

[11] Gunes E, Florczak CK (2025) Replacing or enhancing the human coder? Multiclass classification of policy documents with large language models. J Comput Soc Sci 8(2):31

[12] Halterman A, Keith KA (2025) Codebook LLMs: evaluating LLMs as measurement tools for political science concepts. Polit Anal. https://doi.org/10.1017/pan.2025.10017

[13] Jones BD, Baumgartner FR (2005) The politics of attention: how government prioritizes problems. University of Chicago Press, Chicago

[14] Khattab O, Singhvi A, Maheshwari P, Zhang Z, Santhanam K, Vardhamanan S, et al (2023) DSPy: compiling declarative language model calls into self-improving pipelines. arXiv:2310.03714

[15] Kristensen-McLachlan R, Canavan M, Kárdos M, Jacobsen M, Aarøe L (2025) Are chatbots reliable text annotators? Sometimes. PNAS Nexus 4(4). https://doi.org/10.1093/pnasnexus/pgaf069

[16] Liu M, Shi G (2024) Enhancing LLM-based text classification in political science: automatic prompt optimization and dynamic exemplar selection for few-shot learning. arXiv:2409.01466. https://doi.org/10.48550/arXiv.2409.01466

[17] Morucci M, Spirling A (2025) Model complexity for supervised learning: why simple models almost always work best, and why it matters for applied research

[18] Opsahl-Ong K, Ryan MJ, Purtell J, Broman D, Potts C, Zaharia M, Khattab O (2024) Optimizing instructions and demonstrations for multi-stage language model programs. In:

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp 9340–9366. Association for Computational Linguistics

[19] Reich A, Thoms C, Schrimpf T (2025) Introducing HALC: a general pipeline for finding optimal prompting strategies for automated coding with LLMs in the computational social sciences. arXiv. https://doi.org/10.48550/arXiv.2507.21831

[20] Sebők M, Kacsuk Z (2021) The multiclass classification of newspaper articles with machine learning: the hybrid binary snowball approach. Polit Ana 29(2):236–249

[21] Than N, Fan L, Law T, Nelson LK, McCall L (2025) Updating "The future of coding": qualitative coding with generative large language models. Sociol Met Res 54(3):849–888

[22] Törnberg P (2024) Large language models outperform expert coders and supervised classifiers at annotating political social media messages. Soc Sci Comput Rev

[23] Wilkerson J, Adler ES, Jones BD, Baumgartner FR, Freedman G, Theriault SM, Craig A, Epp DA, Sullivan ME, Khandekar S, Little D (2025) Policy Agendas Project: Congressional Bills

[24] Yildirim TM, Chang MK, Williams L (2026) Origins of public agenda capacity: the role of attention, information, and civic resources. Public Opin Q

[25] Yuan J, Li H, Ding X, Xie W, Li YJ, Zhao W, et al (2025) Give me FP32 or give me death? Challenges and solutions for reproducible reasoning. arXiv:2506.09501

[26] Ziems C, Held W, Shaikh O, Chen J, Zhang Z, Yang D (2024) Can large language models transform computational social science? Comput Linguist 50(1):237–291

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- NPSSuppMaterials.pdf