Cambridge Medium (2020), **37**, 1–36

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE TYPE

PoliPrompt: A High-Performance Cost-Effective LLM-Based Text Classification Framework for Political Science

Menglin Liu* and Ge Shi*†

University of California, Davis
*Corresponding author. Email: mliliu@ucdavis.edu; geshi@ucdavis.edu

Abstract

Recent advancements in large language models (LLMs) have opened new avenues for enhancing text classification efficiency in political science, surpassing traditional machine learning methods that often require extensive feature engineering, human labeling, and task-specific training. However, their effectiveness in achieving high classification accuracy remains questionable. This paper introduces a three-stage in-context learning approach that leverages LLMs to improve classification accuracy while minimizing experimental costs. Our method incorporates automatic enhanced prompt generation, adaptive exemplar selection, and a consensus mechanism that resolves discrepancies between two weaker LLMs, refined by an advanced LLM. We validate our approach using datasets from the BBC news reports, Kavanaugh Supreme Court confirmation, and 2018 election campaign ads. The results show significant improvements in classification F1 score (+0.36 for zero-shot classification) with manageable economic costs (-78% compared with human labeling), demonstrating that our method effectively addresses the limitations of traditional machine learning while offering a scalable and reliable solution for text analysis in political science. A free software Python package will be publicly available on GitHub very soon.

Keywords: text classification, sentiment analysis, large language models (LLMs)

1. Introduction

Text has always been an important data source in political science. Text analysis in political science can be categorized into several key tasks, including classification (Boussalis and Coan 2016; Farrell 2016), scaling (Barberá and Rivero 2015; Lauderdale and Herzog 2016), text reuse detection (e.g., Hertel-Fernandez and Kashin 2015; Hertel-Fernandez 2018), and natural language processing (e.g., Bird, Klein, and Loper 2009; Leetaru and Schrodt 2013). Among these, classification is the most common task in text analysis in the field. For instance, researchers might investigate whether campaign ads are positive or negative across different media platforms (Fowler et al. 2021), or examine if online posts and newspaper coverage in authoritarian countries highlight local government wrongdoing (Pan and Chen 2018). These examples, along with many others in political science, demonstrate that understanding the nature of politics requires insight into what political actors are saying and writing. With recent advances in theory and practice, political scientists increasingly rely on machine learning (ML) methods to classify large corpora of text by measures such as topic and tone (Grimmer and Stewart 2013).

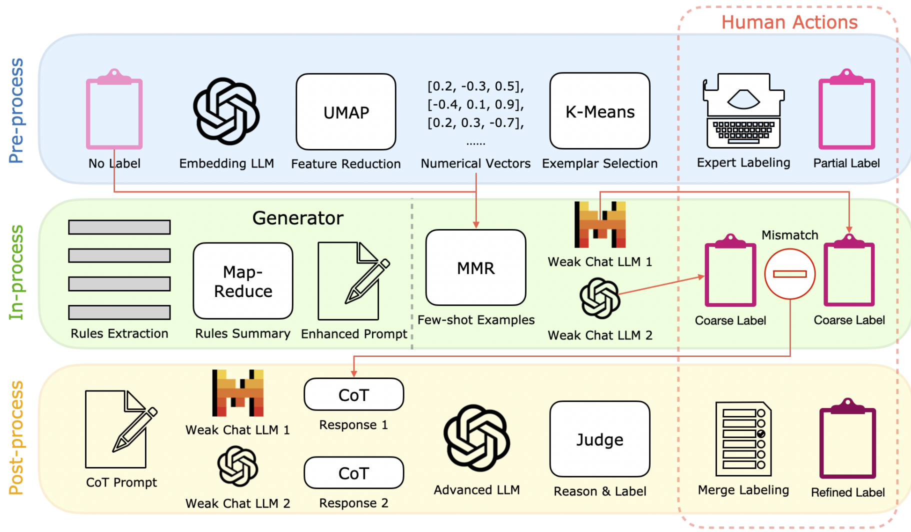

Figure 1. The overview of the three-stage framework with LLM.

Traditional machine learning methods (Qader, Ameen, and Ahmed 2019; Yan et al. 2020; Basu, Walters, and Shepherd 2003; Devlin et al. 2019; Sun et al. 2020), such as Bag of Words (BoW), Support Vector Machines (SVM), and Bidirectional Encoder Representations from Transformers (BERT), have significantly contributed to text classification. Models like BoW and SVM rely on feature extraction from text or selection of features from correlated data, often requiring manual tuning and domain expertise for optimal performance. While BERT represents a more advanced approach, it demands substantial computational resources and extensive data for fine-tuning. These methods necessitate **task-specific training**, meaning that each new task requires a fresh round of model training to achieve optimal results. For example, transitioning from topic modeling to sentiment analysis requires training an entirely new model. As a result, these approaches typically rely heavily on human annotators to label data for training, making them **labor-intensive**, costly, and less flexible. This lack of generalizability across tasks is a major limitation, reducing the reusability of the model for different text classification challenges and piling up the cost.

In recent years, large language models (LLMs) such as Mistral (Jiang et al. 2024) and GPT (Radford and Narasimhan 2018) have emerged as powerful tools for text analysis, offering the ability to generate predictions without the need for task-specific training. These models, pre-trained on a vast corpus of text, possess a deep understanding of language and can perform zero-shot and few-shot predictions—wherein the model makes predictions **without task-specific training**[1]. This capability represents a significant leap forward in terms of efficiency and accessibility, particularly for social scientists who may lack the resources to develop and train their own models.

However, despite their advantages, LLMs are not without their own set of challenges. One of the primary drawbacks is that they can produce inaccurate predictions when provided with simple or **poorly constructed prompts** (Brown et al. 2020). Even when incorporating a few-shot learning approach—wherein the model is given a very small number of annotated examples to guide its predictions—the **fixed nature** of these examples may result in poor performance if the examples do not adequately represent the broader dataset. Besides, current practices (Wang 2023; Egami et al. 2024) that only provide a label **obscure the reasoning** behind the LLM's predictions from humans. As a result, if the LLM misinterprets the human-constructed prompt, this misunderstanding goes unnoticed, limiting the opportunity for human-LLM interaction to refine the prompt, inject additional human knowledge, and perform validation. Moreover, while advanced LLMs such as GPT-4 offer higher levels of accuracy and general intelligence, they come with **significant costs** and **usage restrictions** compared with weak ones such as GPT-3.5 (Table 1).

Table 1. Comparison of GPT-4-turbo and GPT-3.5-turbo at the time of writing (OpenAI).

Feature	GPT-4-turbo	GPT-3.5-turbo
Input Token Pricing	$10 / 1M tokens	$0.5 / 1M tokens
Output Token Pricing	$30 / 1M tokens	$1.5 / 1M tokens
Tokens per Minute	10,000	200,000
Batch Queue Limit	100,000	2,000,000

To address these challenges, we propose a novel LLM-based three-stage (pre-process, in-process, post-process) framework (Figure 1) for text classification which achieves **high accuracy without the need for task-specific training and requires only minimal human annotations.**

First of all, our approach selects a small pool of diverse examples to represent the broader dataset, guiding the LLM to extract classification rules and write enhanced prompts based on accurate human-annotated texts. During inference, it dynamically

1. For instance, unlike traditional machine learning methods that require task-specific training sets—where humans must label data according to the task at hand, such as support/oppose for stance or negative/neutral/positive for sentiment—LLM eliminates the need for such labor-intensive model training each time the task changes.

selects examples that are most similar to the query text[2], thereby enhancing the relevance of the examples used for prediction. This dynamic selection process addresses the limitations of fixed examples in few-shot learning and significantly improves the accuracy of the model's predictions.

Furthermore, our method capitalizes on the consensus among two weaker LLMs, rather than relying on a single advanced model. By aggregating the outputs of these models, we achieve a level of accuracy comparable to that of more sophisticated LLMs at a fraction of the cost. This approach not only reduces computational time but also circumvents the restrictive usage limits imposed by the service provider of the advanced model.

Another key advantage of our framework is its ability to provide explanations for the model's predictions. Unlike traditional machine learning methods that only give a prediction, our method produces interpretable reasoning behind each annotation for hard queries. This transparency is crucial in social sciences, where the stakes of misinterpretation are high, and the ability to understand and justify the reasoning behind predictions is essential for scholarly rigor.

In the following sections, we will delve into our methodology and experiments, elaborate on the implementation details, clarify the rationale behind each module, validate the outlined advantages, and demonstrate the framework's effectiveness through text classification tasks in political science.

2. A query text is the text a user asks the LLM to annotate.

2. Literature Review

Supervised machine learning for text classification. Traditional machine learning methods such as BoW, N-grams, and TF-IDF combined with SVM or Bayes have played a significant role in the analysis of text data within the social sciences (Collingwood and Wilkerson 2012; Drutman and Hopkins 2013; Grimmer and Stewart 2013; Ceron et al. 2014; Wilkerson and Casas 2017). These methods, however, come with substantial limitations. At their core, traditional supervised machine learning approaches require intensive feature engineering—a process in which researchers manually select and construct features or covariates that are thought to be relevant for the prediction task. This feature engineering is often based on human intuition, which is inherently subjectively biased and may lead to flawed conclusions about the observation under study. Language models based on neural networks such as Word2Vec (Rui and Yutai 2020), RNN (Liu, Qiu, and Huang 2016), BERT (Devlin et al. 2019), by contrast, rely solely on the text itself, bypassing the need for human-driven feature selection. However, these models generally require large amounts of data for training, which poses a significant challenge in social science research, where datasets are often scarce or costly to produce.

Both methods rely heavily on extensive human annotations, making them labor-intensive and challenging for researchers. The time required for hiring, managing, and training labor, coupled with the steep learning curve for model development, renders these methods costly and inefficient to scale. Additionally, these approaches often lack the flexibility to adapt to new data or evolving research questions without task-specific training (Terechshenko et al. 2020). For instance, a model trained for sentiment analysis to distinguish between positive and negative sentiment cannot be directly applied to a different task, such as topic modeling.

Large Language Models for Text Classification. LLMs represent an evolutionary approach to text annotation in political science. A token, like a word in natural language, is the minimum input to an LLM for processing. Auto-regressive LLMs (e.g.Touvron et al. 2023; Team et al. 2024; OpenAI et al. 2024), like GPT, operate based on auto-regressive principles, where the model generates text one token at a time, predicting the next token based on the previous ones. Unlike traditional models, LLMs can be treated as black-box tools, accessible via commercial APIs, where users can simply create prompts based on heuristics to direct the model's output. This capability makes LLMs suitable for a wide range of text annotation and classification tasks without the need for task-specific training. While convenient to use, the classification accuracy of LLMs with simple heuristic prompts is not guaranteed, particularly for less advanced models. The development of LLMs from weak to advanced models has vastly improved their performances. However, the use of advanced LLMs, such as GPT-4, comes with significant practical limitations, including higher cost, slower inference speed, and stricter token usage limit compared to models like GPT-3.5 (Table 1). These challenges pose significant barriers to the widespread application of LLMs in political science. Our proposed framework aims to address these challenges by providing a more accurate, efficient, and cost-effective approach to leveraging LLMs in this field.

3. Methodology

In the realm of text classification, classic supervised learning approaches necessitate the availability of a large number of annotated examples, upon which models such as SVM and BERT classifiers are trained. Given a task $\mathcal{T}$, a dataset with input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ are the text inputs and y_i are the corresponding labels, the optimization objective is to find the model parameters θ that maximize the likelihood of the correct labels:

$$\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} P(y_i|\mathbf{x}_i, \theta)$$

These models optimize the likelihood of given labels, adjusting their parameters iteratively to fit the training data. This process, while effective, is often computationally expensive and requires substantial amounts of labeled data to achieve high accuracy.

In contrast, we explore the efficacy of **in-context learning** (Brown et al. 2020), particularly in the context of using auto-regressive large language models (LLM) like Generative Pre-trained Transformer (GPT) for text classification tasks. In-context learning refers to the process where a language model makes predictions based on the context provided by a sequence of input tokens, without updating the model parameters. Instead of updating model parameters θ, the optimization objective of in-context learning for the LLM is to maximize the log-likelihood of the correct class label y given the prompt and query text input $\mathbf{x}$:

$$\max_{\text{prompt}} \frac{1}{N} \sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i, \text{prompt}; \theta)$$

The prompt is natural language text that describes the instructions from humans guiding what the LLM should do. It is usually defined as merely a task description (**zero-shot learning**) or a task description augmented by a few input-output pairs examples (**few-shot learning**).

$$\text{prompt} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_k, y_k), \text{task\_description}\}$$

In zero-shot learning, LLMs make predictions based on common knowledge injected by extensive pre-training conducted by service providers such as OpenAI, Meta, and Google. However, it may not be pertinent enough for a custom user-defined task. Few-shot learning addresses the challenge of injecting special knowledge by leveraging a small number of annotated examples to augment the prompt. The model uses the context provided in the input (e.g., examples of input-output pairs) to make predictions for a new query input. It can be formalized as, given a prompt that includes k ($k < 10$) examples of input-output pairs $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_k, y_k)\}$ and a new input $\mathbf{x}$, the goal is to predict the corresponding output y.

$$P(y|\mathbf{x}, \text{prompt}; \theta)$$

Normally, the tradition of designing prompts heavily relies on human heuristics which is usually referred to as prompt engineering. Users create prompts that guide the

LLM in producing the desired output in a trial-and-error manner. Although there are practical guidances on how to write prompts, they are far from ideal since users don't know if the LLM understands the task description or potential pitfalls. In addition, for a task $\mathcal{T}$, only k input-output pairs are pre-selected to augment the prompt for all N examples in the dataset. That is, for every single query text, an LLM sees the same set of examples to make a prediction. However, considering the same set of examples may not be appropriate for all query texts, this simple design cannot maximize the ability of LLM. For example, the sentiment of an example text in Japanese may not be able to guide the sentiment classification of a text in English. Nevertheless, high-performance labeling relies on advanced LLM which are more and more expensive.

To address these challenges and achieve high-performance labeling, we propose an in-context learning framework with three stages: pre-process, in-process, and post-process, as shown in Figure 1. The following sections detail the methodologies and benefits of every stage.

3.1 Pre-process

LLMs face a significant constraint in their context window size, limiting the number of examples that can be used in few-shot learning scenarios. Despite this limitation, there's often a need to incorporate knowledge from a broader set of examples to enhance the LLM's performance. To address this challenge, we propose a novel approach: first, we select a pool of M representative examples for human labeling, where $k < M \ll N$. Then, at inference time when calling the commercial API to classify unlabeled query text, we dynamically choose the best k examples to augment the prompt. In the "pre-process" stage, we curate such an exemplar pool for human labeling. This curated pool serves two crucial purposes: it allows the LLM to infer the underlying rules used for labeling, and it enables adaptive selection of the most relevant examples for each query text. By creating this initial pool of human-annotated examples, we lay the foundation for more effective and efficient use of the LLM in subsequent stages, balancing the need for comprehensive knowledge injection with the constraints of the model's context window. In this stage, we prepare this exemplar pool, and here are the steps.

Converting texts into embeddings. With natural language inputs, LLMs can convert them into numerical embedding vectors of uniform size. We apply this process to all unlabeled texts in the dataset using embedding models such as "text-embedding-3-small" from OpenAI. Based on relevant literature (Steck, Ekanadham, and Kallus 2024), we use cosine distance to measure the similarity between embedding pairs, providing a quantitative basis for comparing text semantics.

$$\text{cosine\_distance}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$$

Feature reduction with UMap. Uniform Manifold Approximation and Projection (UMap) is a powerful dimensionality reduction technique that preserves data relationships in lower-dimensional space. We employ UMAP (McInnes, Healy, and Melville 2020) with cosine distance for feature reduction of the embeddings, which offers two key advantages: it allows us to use simpler Euclidean distance calculations on the reduced embeddings in subsequent stages, and it significantly decreases the feature set size, thereby accelerating computational processes.

Exemplar selection. To create a diverse and representative pool of examples, we employ an exemplar selection method (Bien and Tibshirani 2011) based on the manifold structure of the reduced embeddings. While various approaches exist for exemplar or prototype selection, such as set cover algorithms and density-based sampling, we opt for the k-means selector due to its simplicity and effectiveness. This method involves performing k-means clustering on the embeddings and then designating the text whose embedding is nearest to each cluster center as an exemplar. This approach ensures that every distinct group within the data is represented by one example, allowing for easy control over the number of exemplars selected while maintaining a comprehensive coverage of the embedding space. We keep M such texts as an exemplar pool and engage a human expert to label them. Typically, M is less than 100, requiring minimal human effort.

3.2 In-process

In this stage, we use the exemplar pool to improve both the task description and few-shot prompt. Then, ask LLM to perform text classification with this prompt.

Enhanced task description generation. To enhance the initial task description, we leverage the LLM's analytical capabilities on the labeled exemplar pool. For each input-output pair from the pool, the LLM examines the rationale behind the human-assigned label. We then employ a Map-Reduce approach, where the LLM first "maps" by analyzing individual examples, and then "reduces" by summarizing the labeling rules for each class. This process ensures the generated rules are LLM-interpretable. Human experts can verify these rules for accuracy and intent. Typically, humans copy and append these generated rules to the initial prompt, creating an enhanced task description. In cases of inaccuracies, humans shall explicitly instruct the LLM to oppose specific incorrect rules in the prompt.

Few-shot example retrieval. When calling LLM API for text classification, in a few-shot prompt setup, we retrieve an unlabeled query text and utilize its pre-computed embeddings to search for the top-k texts from a pool, selecting those with the highest scores using the Maximal Marginal Relevance (MMR) algorithm. The MMR (Parmar, Wu, and Blackhurst 2007) algorithm balances relevance and diversity by considering both the similarity between embeddings and the uniqueness of the selected examples, ensuring that the retrieved examples are not only relevant to the query but also varied enough to provide a comprehensive context.

Given a query embedding $\mathbf{x}_q$, a pool R of exemplar texts, the MMR score for a candidate item $\mathbf{x}_j$ from the pool R is defined as:

$$\mathbf{x}_j = \arg\max_{\mathbf{x}_j \in R \setminus S} \mathrm{MMR}(R) := \arg\max_{\mathbf{x}_j \in R \setminus S} [\lambda \cdot \mathrm{Sim}(\mathbf{x}_q, \mathbf{x}_j) - (1 - \lambda) \cdot \max_{\mathbf{x}_i \in S} \mathrm{Sim}(\mathbf{x}_j, \mathbf{x}_i)],$$

where S is the set of already selected items that is initially empty, and λ is a trade-off parameter between relevance and diversity ($0 \leq \lambda \leq 1$). In this way, we retrieve exemplars that are either semantically close to the query text with the same correct label or hard negatives that share some similarity with $\mathbf{x}_q$ but from a different class. These input-out pair examples are automatically appended to the task description for further enhancement.

Coarse annotation with weak LLMs. Using a prompt enhanced with clearer task descriptions and carefully selected examples, we employ chat LLMs to assign labels from predefined options. To efficiently handle the labeling process, we utilize two instances of a weaker LLM (e.g., GPT-3.5) to label all the unlabeled texts twice. This approach mitigates high costs, usage limits, and long processing times. The weaker LLMs are expected to agree on easy queries with high accuracy, while any discrepancies are tracked in a mismatch collection for further review.

3.3 Post-process

In this stage, we leverage more advanced LLMs (e.g., GPT-4) and in-context learning techniques to address mismatches identified in the in-process stage. The prompts used here are refined versions of those from the previous stage, focusing on enhancements for fine-grained annotation and versatility. Although these prompts are less cost-effective, they are applied only to a limited number of queries in the mismatch collection, thereby mitigating the cost concern.

Chain-of-Thought Prompting. A chain-of-thought (CoT) prompt guides a large language model (LLM) through a step-by-step reasoning process to enhance its ability to tackle complex tasks (Wei et al. 2023). To implement it, the LLM is instructed to first analyze the content according to the task description, providing reasoning at each step before delivering the final answer. This approach works by mimicking human problem-solving, breaking down tasks into smaller components, which helps the model grasp the underlying logic and produce more accurate responses by not only assigning a label but also offering the reasoning behind it. The output of CoT is a sequence $y = (y_1, y_2, \ldots, y_t)$ tokens, where y_t is the desired prediction and $y_{<t}$ are the reasons:

$$P(y_t | y_{<t}, \mathbf{x}, \text{prompt}; \theta)$$

Advanced LLM as a judge. Leveraging an LLM as a judge (Zheng et al. 2023) involves utilizing its advanced reasoning capabilities to evaluate and validate the outputs of other models. In our approach, we first ask two weaker LLMs to provide annotations and reasoning to the small mismatch set through a CoT process. Then, an advanced LLM assesses the quality, accuracy, and consistency of these responses, adding an extra layer of validation. This method refines coarse predictions on challenging examples, enhancing the reliability of outputs in complex tasks.

3.4 Summary of Framework

In this section, we summarize the entire framework, outlining each module's input-output and the human involvement required throughout the process.

The "pre-process" stage prepares a pool of representative and diverse examples for human annotation in the subsequent stages. The input is human-collected, unlabeled texts, which we process using LLM embedding models, saving selected indices as an exemplar pool. Following this stage, human experts accurately annotate the texts in the pool and draft an initial prompt that includes only a task description. The "in-process" stage focuses on enhancing this initial prompt with a refined task description and more appropriate examples. Here, two weaker LLMs generate coarse annotations for the unlabeled examples. Users are responsible for running the task description generator, verifying its validity, and appending it to the initial prompt. After labeling, humans clean the predictions, identify mismatches, and record their indices in the dataset. The "post-process" stage refines the predictions for these mismatches and provides reasoning for the adjustments. Humans shall clean the responses from an advanced LLM, replace coarse predictions with fine-grained ones, or conduct human evaluations with the aid of LLM-generated reasoning.

By following this framework, we minimize human effort, enabling rapid experimentation and delivering high-accuracy predictions infused with human knowledge, all while reducing costs in terms of both time and resources.

4. Experiments

Setup. We evaluated our method using three distinct text datasets with human-labeled classifications. In each experiment, we initially treated all texts as unlabeled and tasked LLMs with predicting the labels. We use the model "text-embedding-3-small" from OpenAI to convert all texts into embedding vectors and reduce the size of a vector to 24 using UMap. Subsequently, we selected 80 exemplars through KMeans from each dataset to form the example pool used to develop the enhanced prompts[3] Throughout our experiments, "GPT-3.5-turbo" and "Mistral-medium" were consistently employed as the weak models. A λ parameter of value 0.8 is chosen for the MMR dynamic retriever. When referencing the advanced or judge model, we refer to either "GPT-4-turbo" or "Mistral-large-latest".

In the first experiment, using a clean, correctly labeled multi-class news dataset(Greene and Cunningham 2006), we demonstrated our method's exceptional performance in accurately labeling text topics, even for lengthy inputs. Our three-stage approach systematically enhanced precision, with each module contributing cumulative improvements, as confirmed by an ablation study. Notably, our method achieved a significant improvement in overall F1 score, increasing from 0.94 with the naïve LLM labeling to an impressive 0.96.

In the second experiment, we specifically highlight the effectiveness of our second-stage generator. We tasked LLMs with labeling tweets as "support" or "oppose" in relation to Brett Kavanaugh's Supreme Court confirmation process. Initially, we observed that weaker LLMs often struggled with discerning sentiment and stance, leading to confusion. However, when utilizing our second-stage in-process generator, the LLM generated an enhanced prompt by summarizing the labeling rules for "support" and "oppose" based on 80 human-labeled tweets. This resulted in a dramatic improvement in labeling accuracy, with F1 scores soaring from 0.57 to 0.95.

In the third experiment, we attempted to replicate Fowler et al.'s (2021) study, which examines how the medium of campaign ads influences their tone. However, we encountered a very noisy, human-labeled dataset. Our method effectively uncovered potential issues within this supposedly gold-standard dataset, highlighting how noisy human labels can significantly impact downstream political science analysis. We also addressed some pitfalls of using our method, emphasizing the importance of careful data handling.

4.1 Classifying BBC News Reports Topics

In our first experiment, we applied our method to a multi-category classification task involving extensive and lengthy text. We chose to label the topics of BBC news reports due to their diversity and relevance in benchmarking machine learning models. The dataset comprises 2,225 news articles sourced from the BBC News website, covering stories across five topical areas—business, entertainment, politics, sport, and tech—from

3. We conducted hyperparameter tuning experiments to assess the impact of varying the number of exemplars, testing sample sizes of 20, 40, 60, 80, and 100. Although F1 scores consistently improved as the number of exemplars increased, the rate of improvement slowed beyond a certain point. For our analysis, we chose to use only 80 exemplars in all three experiments, deliberately demonstrating that our method can outperform traditional ML approaches even with a relatively modest number of examples. As a result, the performance metrics we report are conservative estimates, not reflecting the highest potential performance. Full results from the hyperparameter experiments can be found in Appendix 1.

the years 2004-2005. This dataset, originally compiled by Greene and Cunningham (2006), has been widely used in machine learning research as a benchmark for evaluating the performance of various classification algorithms. The diversity and structure of this dataset make it an ideal candidate for testing the robustness and accuracy of our proposed method in a real-world, multi-class classification scenario.

Starting with a simple heuristic prompt, as depicted in Figure 2, we guided two weak LLMs to identify the primary topic of a news report from five categories: politics, business, sport, entertainment, and technology. Next, we employed a prompt generator, feeding the LLM with 80 exemplars labeled with accurate human classifications and asking the LLM to summarize the rules for categorizing news reports. This newly generated prompt was then fed back to the weak LLMs. As a result, the models not only corrected some of their initial misclassifications but also provided justifications for their decisions. In both zero-shot and five-shot settings[4], these weaker models showed significant improvement in performance when we dynamically selected 80 exemplars compared to their initial raw predictions[5]. For example, as illustrated in Figure 3 when using zero-shot prompting, both GPT-3.5 and Mistral-medium initially achieved F1 scores of approximately 0.89 in the "politics" category. However, these scores increased to around 0.94 after an enhanced prompt was generated. In categories like "sport," where the LLMs already performed exceptionally well with the naïve prompt—achieving F1 scores above 0.97—our method further boosted their performance. we were able to raise the F1 scores to around 0.99 putting into the enhanced prompt.

In Figure 2, we present an example where the two weak LLMs continued to differ in their classifications even after prompt enhancement[6]. In this case, one model classified the news report as related to technology, while the other identified it as business. To resolve this discrepancy, we applied our third stage, which involved chain-of-thought prompting combined with a judge model. "GPT-4-turbo", as the judge model, ultimately classified the report as business, aligning with the human label. This decision was well-justified, as the judge model highlighted that the report, while contextualized within technology, primarily focused on business aspects.

The implementation of the third stage led to a notable improvement in performance metrics across all categories. For instance, when categorizing "politics," both chain-of-thought and judge prompting significantly boosted F1 scores in both zero-shot and five-shot scenarios, with an increase of approximately 0.1. Overall, the macro-level F1 score for all categories saw a substantial increase, reaching nearly 0.97.

In a dataset with accurate human labels, our method shows a progressive improvement in labeling accuracy, requiring only 80 labeled news reports to attain such high performance. This experiment underscores the efficiency and effectiveness of every stage in our approach.

4. We employed five-shot prompting, providing the LLMs with one example per category since there are five topics to classify.
5. Detailed results are provided in Tables B4 and B5.
6. The full text of this news report example is available in the Appendix B2

Prompts for Classifying Topics of BBC News Report

Simple Heuristic Prompt:

In your judgment, what is the major topic of the given news report Text? Please choose only one answer from the following options: "politics", "business", "sport", "technology", or "entertainment". Complete the task very succinctly using only one word written between '<' and '>'.

[Enhanced Prompt][BBC News Text (lengthy text in the Appendix B2)]

Weak Chat Model1: The news report discusses the history and future of Yahoo, including its competition with Google and MSN, its acquisitions, and its plans for the future. It also mentions the importance of search technology and the potential expansion into rich media. Based on this, the major topic of the news report is "technology." Therefore, the correct answer is <technology>.

Weak Chat Model2: The text discusses the future prospects of Yahoo, a major internet company, focusing on its competition with Google and MSN, its business strategies, and its potential dominance in the next decade. Therefore, the correct answer is <business>.

Judge Prompt:

You are given 2 responses ("Response 1" and "Response 2") to the following task about the "Text", which can be correct or wrong.
Please judge which response (or neither) to the following content is correct step by step and provide your reasoning succinctly (Do not exceed 100 words). Finally, based on the reasoning, choose the correct answer and repeat the response's (or your) choice once in '<' and '>'.

[Response 1][Response 2]

Judge Chat Model: Let's **think step-by-step**. Response 1 focuses on the technological aspects of Yahoo's history and future, such as search technology and expansion into rich media. However, the text primarily discusses Yahoo's business strategies, competition, and market position, rather than technological innovations. Response 2 correctly identifies the text's focus on Yahoo's business aspects, such as competition with Google and MSN, acquisitions, and future prospects. Therefore, the correct answer is <business>.

Figure 2. An Example of 0-Shot Prompts for Analyzing BBC News Reports

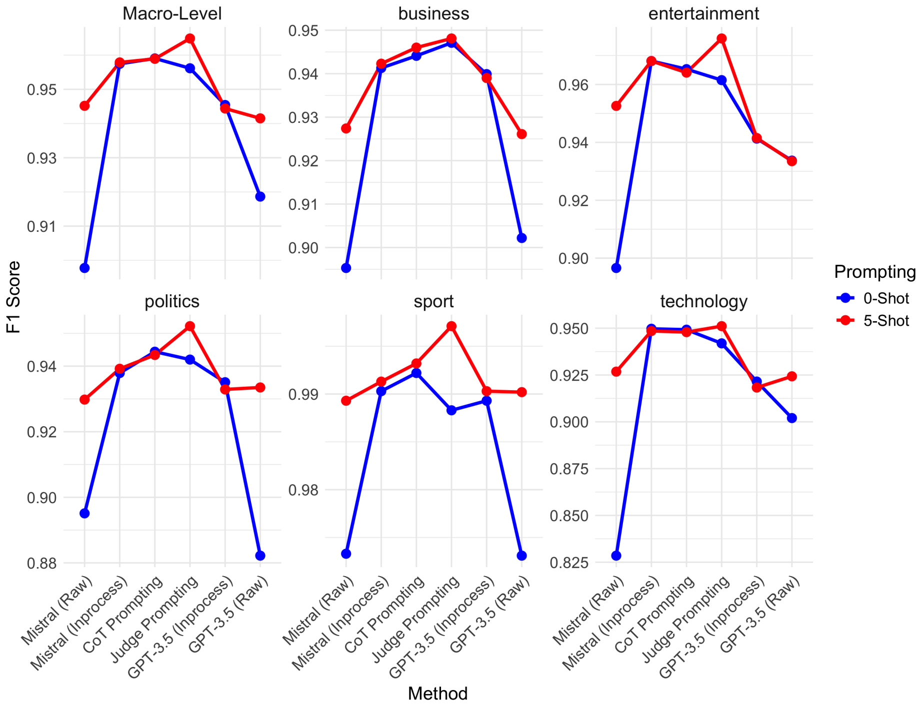

Figure 3. Analyzing BBC Topics: Comparison of F1 Scores Across Different Methods

4.2 Measuring Public Opinion Toward Brett Kavanaugh's SCOTUS Nomination

In the second experiment, we focus on demonstrating the effectiveness of our second stage—the prompt generator—in accurately labeling nuanced political science concepts.

In their 2023 paper, Bestvater and Monroe argue that sentiment and stance are fundamentally distinct concepts, a difference that traditional sentiment analysis tools like the VADER dictionary and the Lexicoder Sentiment Dictionary (LSD) often fail to capture. While sentiment analysis gauges the emotional tone of a document—whether positive, negative, or neutral—stance identification determines the author's position on a specific issue, which may not align directly with the sentiment.

Bestvater and Monroe explore this distinction by analyzing a corpus of tweets about Kavanaugh's nomination(2023). Their study compares the effectiveness of various text classifiers in identifying stance and concludes that, for many stance detection tasks, training a new supervised classifier on a hand-labeled dataset yields more accurate results than relying on existing models or dictionaries designed for sentiment analysis. The key findings from their study are summarized in the Appendix Table B1.

To assess the ability of our method to distinguish nuanced political science concepts and generate accurate predictions, we applied it to the human-labeled dataset from Bestvater and Monroe (2023), which serves as the ground truth benchmark. We adopted the same instructions given to the human coders in their study and used them as the foundational heuristic prompt for two weaker language models (Mistral-medium and GPT-3.5). The simple heuristic prompt is detailed in Figure 4:

Using this 0-shot prompt, both Mistral-medium and GPT-3.5-turbo achieved F1 scores below 60%, as the weak LLMs struggled to differentiate between sentiment and stance. For instance, these LLMs mistakenly labeled the following Tweet as "oppose":

RT @atensnut Democrats can't just "move on" and jump on the bandwagon of Sketchy allegations against Kavanaugh, without accepting the egregiousness of turning their backs on the victims of Bill Clinton.

The LLMs categorized this as "oppose" because the tweet conveys a tone of negativity and distrust, implicitly criticizing the Democratic Party for supporting allegations against Kavanaugh while highlighting their perceived hypocrisy. This example illustrates how weak LLMs, when using a simple heuristic prompt, can easily confuse sentiment with stance.

We then provided weak models with enhanced prompts produced by LLM in the second stage of our method, shown in Figure 4, offering a clear and concise guideline that significantly refined the LLMs' judgment. Notably, the enhanced prompt goes beyond mere factual classification; for instance, it correctly identifies tweets as "approve" when they present evidence that undermines Kavanaugh's accusers. Additionally, the enhanced prompt effectively captures and summarizes emotional tones that could be interpreted as supportive of Kavanaugh, such as mocking his opponents and expressing frustration toward actions obstructing his confirmation. These nuances are easily confused with opposition when using a simple heuristic prompt, as the negative tone and sentiment of such tweets can misleadingly suggest an opposing stance. This simple guideline ensures that both factual content and emotional cues are considered, leading to more accurate and nuanced classifications. The enhanced prompt resulted in substantial improvements, with Mistral-medium and GPT-3.5 achieving impressive F1 scores of 91.69% and 92.59%, respectively. These results mark a significant improvement over the initial performance achieved with the simple heuristic prompt, showing approximately a 36% increase in F1 score for zero-shot prompting, as illustrated in Figure A1. Similarly, in the four-shot[7] prompting scenario, our enhanced prompt enabled the weak LLMs to achieve over a 20% improvement, demonstrating the importance of having an enhanced prompt in generating the right classifications and label[8].

Our method demonstrates superior performance compared to raw LLM predictions, largely due to the enhanced prompt generated in the second stage of our approach. This underscores an important consideration for political scientists: when using LLMs for text classification, relying on raw labeling alone can lead to significant misclassifications, as LLMs may struggle with nuanced concepts like sentiment and stance. However, by providing a few exemplars and instructing the LLM to generate a more refined prompt with simple, targeted guidelines, we achieve outstanding results. Furthermore, our method outperforms both dictionary-based and traditional supervised learning

7. Detailed few-shot prompts are provided in Appendix B1.

8. The detailed resulting performance metrics for 4-shot promptings are presented in Appendix B3. Essentially, by incorporating the final stage of chain-of-thought reasoning and the "Mistral-large-latest" judge model, our approach further enhanced the F1 scores, surpassing traditional supervised classifiers and reaching impressive levels above 95%.

approaches, all without the need to train a new model—contrary to the recommendation by Bestvater and Monroe (2023) for improved classification. We successfully minimized the cost associated with model training while maximizing classification accuracy.

Prompts for Analyzing Stance towards Kavanaugh

Simple Heuristic Prompt:

In your judgment, whether the specific stance the author expresses toward the confirmation of Brett Kavanaugh is approving or opposing? Please choose your answer only from the 2 options – "approve" and "oppose". Complete the task very succinctly using only one word written between '<' and '>'.

Enhanced Prompt:

You are a stance analyzer. In your judgment, whether the specific stance the tweet text expresses toward the confirmation of Brett Kavanaugh is approving or opposing? Note: Focus on the stance expressed regarding Kavanaugh's confirmation. Emotional tone (e.g., anger, happiness) should be considered only if it directly influences the stance. Please choose your answer only from the 2 options – "approve" and "oppose". Complete the task very succinctly using only one word written between '<' and '>'.

Oppose Stance:

- *Lending credibility to allegations or accusations against Kavanaugh*
- *Highlighting potential disqualifying factors or controversies about Kavanaugh*
- *Expressing criticism, concerns, or questions about Kavanaugh's suitability*
- *Suggesting credible misconduct allegations should disqualify Kavanaugh as a nominee*

Approve Stance:

- *Discrediting or undermining accusations against Kavanaugh*
- *Presenting evidence that weakens the case against Kavanaugh*
- *Defending or rationalizing Kavanaugh's nomination despite allegations*
- *Expressing frustration towards actions obstructing/delaying Kavanaugh's confirmation*
- *Mocking, dismissing, or discrediting Kavanaugh's accusers/opponents*

Figure 4. 0-Shot Prompts for Analyzing Tweets about Brett Kavanaugh

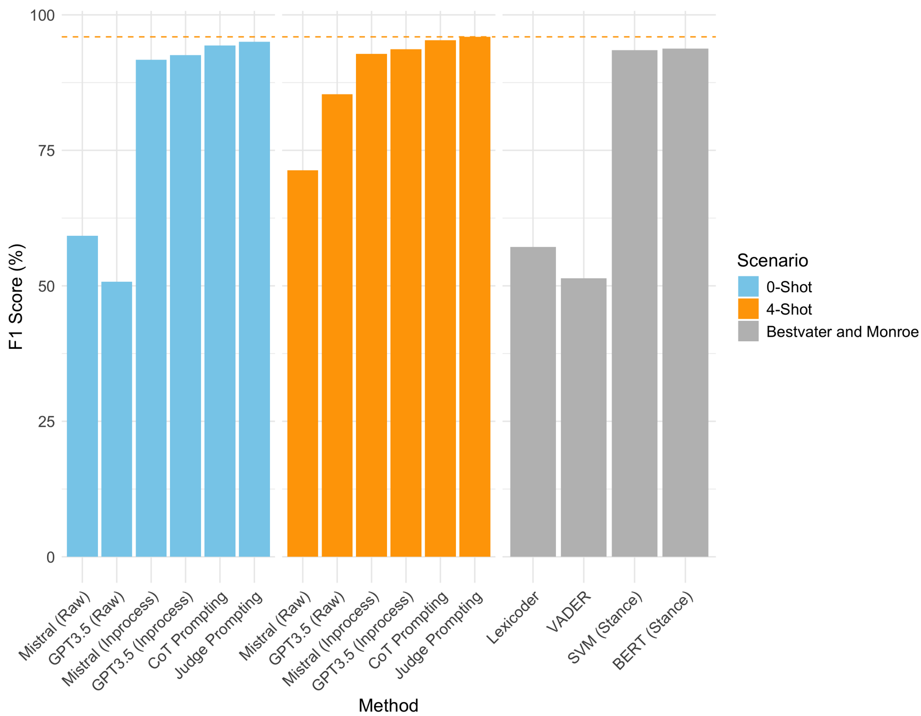

Figure 5. Measuring Opinion toward Kavanaugh: Comparison of F1 Scores Across Different Methods

4.3 Classifying Campaign Ads Tones in the 2018 Election

The medium through which political communication is delivered plays a critical role in shaping the message's tone and its audience reach. In a recent study, Fowler and her colleagues examined the impact of Facebook as a medium on the tone of political advertisements (2021). They proposed that ads on Facebook are more likely to adopt a negative tone compared to other platforms.

To explore this hypothesis, the researchers collected data from political advertisements by all federal, statewide, and state legislative candidates during the 2018 elections. A team of research assistants then classified a sample of these ads based on their tone—whether they were *promoting*, *contrasting*, or *attacking*. The dataset comprises a total of 14,642 advertisements, with 9,073 originating from Facebook and 5,569 from television ads, offering a comprehensive basis for comparing online and offline political messaging. We randomly selected a sample of 3,000 observations from the coded training set. Within this sample, 2,374 ads were classified as promoting a candidate, 448 as contrasting between candidates, and 178 as attacking a candidate[9].

We applied the same three-stage approach as in our previous two experiments, this

9. The original dataset exhibits a similar imbalance, with a significantly higher proportion of ads expressing a promotional tone. Importantly, our analysis shows that sampling 3,000 observations does not compromise the validity of downstream political analysis, as regression estimates derived from this subset are consistent with those reported in the original study.

time focusing solely on reporting the final post-process results of the Chain-of-Thought (CoT) and Judge prompting methods. We evaluated their performance under both zero-shot and few-shot prompting conditions[10]. The final F1 scores, as shown in Table 2, indicate notably low performance in the "attack" and "contrast" categories. Specifically, the average F1 score for "attack" is approximately 0.55, while for "contrast," it is even lower, averaging around 0.5. It is important to note that these F1 scores were calculated against gold-standard human labels, which are particularly noisy in this dataset, potentially contributing to the reduced accuracy.

Table 2. Analyzing Ads Tones: Summary of F1 Scores across Different Methods

Class	CoT (k=0)	Judge (k=0)	CoT (k=6)	Judge (k=6)
promote	0.9285	0.9323	0.9260	0.9314
contrast	0.5385	0.4567	0.5180	0.4978
attack	0.5719	0.5368	0.5648	0.5856

Notes: The reported figures represent the F1 scores across different categories, calculated after applying the third-stage chain-of-thought method and the judge model for labeling campaign advertisements. We noted that the F1 scores for both "contrast" and "attack" are particularly low.

Despite these lower F1 scores, we proceeded to use the predicted labels generated by zero-shot chain-of-thought prompting to re-estimate the same fixed-effect regression model described in Fowler's study. Specifically, we utilized a candidate-level fixed effects model, where the dependent variable is the average tone of the candidate's ads across various media platforms[11]. Figure 6 illustrates the impact of different labeling strategies on downstream regression estimates. The left panel shows results obtained using zero-shot chain-of-thought prompting and the judge model, compared to fully human-labeled data. We observe that, except for the prompt tone, the estimates derived from LLM-labeled data diverge significantly from those obtained with human-labeled data. This divergence is expected, given that the F1 scores for these two categories are particularly low.

These findings facilitated a re-examination of the data, particularly in cases where LLM-generated labels differ, as these discrepancies may indicate conceptual confusion and could lead to controversial or inconsistent decisions. Examples of such label inconsistencies, as shown in Table 3, underscore the need to question the consistency and reliability of the so-called gold standard human labels. In the first text, which was labeled as "contrast" by human coders, the primary focus is on criticizing career politicians for failing to prevent the destruction of industries, without directly contrasting specific policies with those proposed by others. On the contrary, the second piece, labeled as "promote" by human coders, explicitly criticizes the Republican Party in the state legislature while calling for support for other candidates. This message could reasonably be interpreted as both a promotion of alternatives and a contrast with the criticized party. Despite the similarities between these two ads—both of which criticize opponents and call for the support of others—the human labels differed. On the other hand,

10. Detailed raw and in-process results are available in the Appendix.

11. Following Fowler et al. (2021), we computed expenditure-weighted averages of the message content for each candidate.

GPT and the final chain-of-thought method remained consistent in their classifications, demonstrating a more stable and coherent approach to labeling.

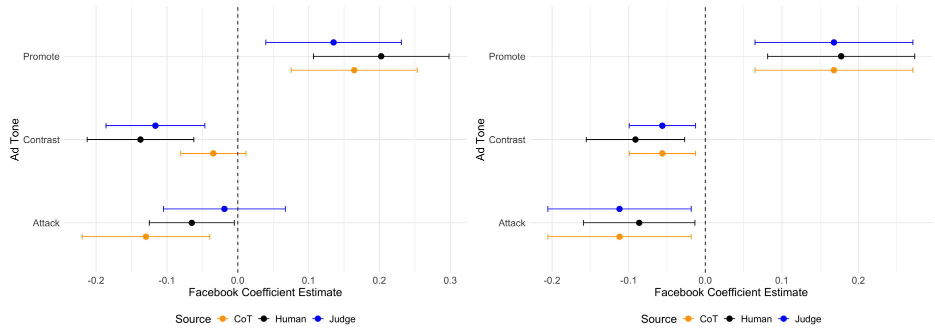

Figure 6. Analyzing Ad Tones: Comparison of Estimates Using Different Labeling Approaches.
Notes: The left panel of this figure shows the effect of medium on campaign ad tones across different labeling approaches, including initial human-labeled data before removing controversial labels. The right panel presents regression estimates after excluding the controversial observations.

Given the noise within the human-labeled data, we proceeded to remove all controversial observations – the observations are labeled differently from two weak LLMs, and Table 4 highlights the substantial improvements in F1 scores across all categories following this adjustment. Notably, the "contrast" and "attack" categories saw over a 30% increase in F1 scores. After this refinement, we reran the downstream fixed effects model. The right panel of Figure 6 shows that the estimates obtained through automated classification now closely align with those derived from human-labeled data[12].

Additionally, Table 2 revealed that zero-shot metrics surpassed few-shot performance. This anomaly can be attributed to the sampling of 80 examples for few-shot prompting during the preprocessing stage. Given the noise within the human labels, these samples might have included incorrectly labeled examples, impacting the accuracy of few-shot promptings. This reliance on potentially flawed data could degrade the performance of few-shot prompts to levels below that of zero-shot prompts, which remain uninfluenced by human input.

This experiment underscores an additional advantage of our method: its ability to identify and flag problematic human labels. While human coding has traditionally been regarded as the gold standard, this assumption falters when dealing with controversial classifications[13]. Our findings urge caution against placing absolute trust in human coding, particularly when relying on a single coder, as disagreements are inevitable.

Furthermore, this experiment also reveals a potential pitfall of our approach. Specifically, when using our method—especially in few-shot prompting scenarios—researchers must ensure the accuracy of the exemplar text labeling. If the exemplars are noisy, it is highly likely to degrade the overall accuracy of the method.

12. The right panel displays identical estimates for both the chain-of-thought and judge models because mismatched observations were removed. As a result, the remaining labels for both models are consistent, leading to the same regression estimates.
13. The inconsistency among human coders is well-documented in the literature.

Table 3. Analyzing Ads Tones: Examples of Inconsistencies and Controversies among Human and LLM Labels

Text	Human	GPT	Mistral	GPT-4 CoT
They wanted jobs - - - NOT HANDOUTS. It was the Career Politicians job to prevent the destruction of an industry in his District. Vote ArtforCongress.com	Contrast	Promote	Attack	Attack: The ad criticizes "Career Politicians" for failing to protect an industry, implying negligence or incompetence. It promotes "ArtforCongress.com" as an alternative, suggesting a vote for change. The focus is more on attacking the incumbents than contrasting policies or promoting Art extensively.
These are many of the common sense pieces of legislation that the GOP controlled Colorado Senate has killed, instead of working for Coloradans. This Tuesday, let's elect folks who represent our Colorado values.	Promote	Promote	Contrast	Attack: The ad criticizes the GOP controlled Colorado Senate for killing legislation, implying they do not represent Colorado values, and calls for electing representatives who do. It does not promote a specific candidate or contrast multiple candidates directly.

Table 4. Analyzing Ads Tones: F1 Scores after Addressing Discrepancies: CoT Prompting vs. Judge Prompting (0-Shot)

Class	CoT Prompting			Judge Prompting		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Attack	0.7051	0.9322	0.8029	0.7094	0.9379	0.8078
Contrast	0.7163	0.7984	0.7551	0.7398	0.7280	0.7339
Promote	0.9895	0.9520	0.9704	0.9838	0.9608	0.9721

Notes: The reported figures represent the F1 scores across different categories, recalculated after removing the controversial human labels.

5. Discussion and Conclusion

Besides the benefits shown in experiments, our method possesses additional advantages worth discussing.

Cost-effective. Our method is highly cost-effective. In our experiments, we needed to annotate an average of 3,000 texts, each approximately 60 words in length for each experiment. Using the standard practice of cross-validation, hiring two workers on Amazon Mechanical Turk (MTurk) would result in a conservative cost estimate of $350 per worker, plus a $100 MTurk fee, totaling $850. In contrast, our approach incurs a total cost of approximately $183, which includes $2 for the embedding model, $92 for the weak models, and $79 for the advanced model. This means our method costs at most 21.5% of the total price of hiring human workers, not to mention the additional computational cost of training.

Extensible. Our method is extensible, allowing researchers to adapt or replace the algorithms within our framework to fit specific research needs. For instance, while we used UMAP, K-means, and MapReduce in our experiments for exemplar selection and prompt generation, these algorithms are not fixed and can be easily substituted with alternatives. This flexibility enables the method to be tailored to different types of data and research objectives, making it a versatile tool for a broad range of political science applications. Additionally, our approach is designed to integrate the latest LLM advancements seamlessly, ensuring that it remains at the forefront of technological progress without requiring significant adjustments or incurring additional costs. For example, if OpenAI releases GPT-5 in the future, our framework supports users to add this new model to their research just by modifying the configure file. This adaptability future-proofs our method, allowing it to consistently deliver cutting-edge performance as LLM technology evolves.

Limitation. Our method has several limitations. First, it is currently limited to categorical classification and is not suitable for tasks that require measuring intensity, degrees, or rankings. For example, our approach lacks the capability to accurately assess political ideology on a traditional 5-point scale, such as distinguishing between strong Democrats, moderate Democrats, Independents, Republicans, and strong Republicans. In future research, we plan to extend our method to handle pairwise comparisons, enabling us to better capture the intensity of party identification and other continuous variables.

Second, our method is currently limited to text data and cannot be applied to more complex data types such as images, videos, and audio, all of which are increasingly gaining attention in political science (e.g., Torres and Cantú 2022; Girbau et al. 2024; Torres 2024). Expanding our method to handle these diverse forms of data is a crucial direction for future research, allowing us to better address the growing variety of data sources in the field.

Conclusion. In summary, our proposed method offers a significant improvement over both traditional machine learning approaches and existing LLM-based practices. By eliminating the need for extensive human laboring, dynamically selecting relevant examples, and leveraging the consensus of weaker LLMs, our framework provides a high-performance and cost-effective solution for text classification in social science research. This approach not only enhances the accuracy and reliability of predictions but also aligns closely with the methodological standards of the field, making it a valuable tool for researchers aiming to conduct robust and reproducible studies.

Data Availability Statement Replication code and data for this article will be made available upon acceptance. The materials will be uploaded to the Harvard Dataverse, and the corresponding citation and DOI will be provided in an updated version of this statement.

Competing Interests The authors declare none

References

Barberá, Pablo, and Gonzalo Rivero. 2015. Understanding the political representativeness of twitter users. *Social Science Computer Review* 33 (6): 712–729.

Basu, A., C. Walters, and M. Shepherd. 2003. Support vector machines for text categorization. In *36th annual hawaii international conference on system sciences, 2003. proceedings of the,* 7 pp.-. https://doi.org/10.1109/HICSS.2003.1174243.

Bien, Jacob, and Robert Tibshirani. 2011. Prototype selection for interpretable classification. *The Annals of Applied Statistics* 5, no. 4 (December). ISSN: 1932-6157. https://doi.org/10.1214/11-aoas495. http://dx.doi.org/10.1214/11-AOAS495.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural language processing with python: analyzing text with the natural language toolkit.* O'Reilly Media.

Boussalis, Constantine, and Travis G. Coan. 2016. Text-mining the signals of climate change doubt. *Global Environmental Change* 36:89–100.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. *Language models are few-shot learners.* arXiv: 2005.14165 [cs.CL]. https://arxiv.org/abs/2005.14165.

Ceron, Luigi Curini, et al. 2014. Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society* 16 (2): 340–358.

Collingwood, Loren, and John Wilkerson. 2012. Tradeoffs in accuracy and efficiency in supervised learning methods. *Journal of Information Technology & Politics* 9 (3): 298–318.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *Bert: pre-training of deep bidirectional transformers for language understanding.* arXiv: 1810.04805 [cs.CL]. https://arxiv.org/abs/1810.04805.

Drutman, Lee, and Daniel J. Hopkins. 2013. The inside view: using the enron e-mail archive to understand corporate political attention. *Legislative Studies Quarterly* 38 (1): 5–30.

Egami, Naoki, Musashi Hinck, Brandon Stewart, and Hanying Wei. 2024. Using imperfect surrogates for downstream inference: design-based supervised learning for social science applications of large language models. *Advances in Neural Information Processing Systems* 36.

Farrell, Justin. 2016. Corporate funding and ideological polarization about climate change. *Proceedings of the National Academy of Sciences* 113 (1): 92–97.

Fowler, Erika Franklin, et al. 2021. Political advertising online and offline. *American Political Science Review* 115 (1): 130–149.

Girbau, Andreu, Tetsuro Kobayashi, Benjamin Renoust, Yusuke Matsui, and Shin'ichi Satoh. 2024. Face detection, tracking, and classification from large-scale news archives for analysis of key political figures. *Political Analysis* 32 (2): 221–239. https://doi.org/10.1017/pan.2023.33.

Greene, Derek, and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd international conference on machine learning (icml'06),* 377–384. ACM Press. https://doi.org/10.1145/1143844.1143892.

Grimmer, Justin, and Brandon M. Stewart. 2013. Text as data: the promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21 (3): 267–297.

Hertel-Fernandez, Alexander. 2018. *Politics at work: how companies turn their workers into lobbyists.* Oxford University Press.

Hertel-Fernandez, Alexander, and Konstantin Kashin. 2015. Capturing business power across the states with text reuse. In *Annual conference of the midwest political science association,* 16–19. Chicago.

Jiang, Albert Q., Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, et al. 2024. *Mixtral of experts.* arXiv: 2401.04088 [cs.LG]. https://arxiv.org/abs/2401.04088.

Lauderdale, Benjamin E., and Alexander Herzog. 2016. Measuring political positions from legislative speech. *Political Analysis* 24 (3): 374–394.

Leetaru, Kalev, and Philip A. Schrodt. 2013. Gdelt: global data on events, location, and tone, 1979–2012. In *Isa annual convention,* 2:1–49. 4.

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. 2016. *Recurrent neural network for text classification with multi-task learning.* arXiv: 1605.05101 [cs.CL]. https://arxiv.org/abs/1605.05101.

McInnes, Leland, John Healy, and James Melville. 2020. *Umap: uniform manifold approximation and projection for dimension reduction.* arXiv: 1802.03426 [stat.ML]. https://arxiv.org/abs/1802.03426.

OpenAI. *Api pricing.* https://openai.com/api/pricing/. Accessed: 2024-08-22.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2024. *Gpt-4 technical report.* arXiv: 2303.08774 [cs.CL]. https://arxiv.org/abs/2303.08774.

Parmar, Darshit, Teresa Wu, and Jennifer Blackhurst. 2007. Mmr: an algorithm for clustering categorical data using rough set theory. 25th International Conference on Conceptual Modeling (ER 2006), *Data And Knowledge Engineering* 63 (3): 879–893. ISSN: 0169-023X. https://doi.org/https://doi.org/10.1016/j.datak.2007.05.005. https://www.sciencedirect.com/science/article/pii/S0169023X07001012.

Qader, Wisam A., Musa M. Ameen, and Bilal I. Ahmed. 2019. An overview of bag of words;importance, implementation, applications, and challenges. In *2019 international engineering conference (iec),* 200–204. https://doi.org/10.1109/IEC47844.2019.8950616.

Radford, Alec, and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. https://api.semanticscholar.org/CorpusID:49313245.

Rui, Zhang, and Han Yutai. 2020. Research on short text classification based on word2vec microblog. In *2020 international conference on computer science and management technology (iccsmt),* 178–182. https://doi.org/10.1109/ICCSMT51754.2020.00042.

Steck, Harald, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is cosine-similarity of embeddings really about similarity? In *Companion proceedings of the acm on web conference 2024,* 201:887–890. WWW '24. ACM, May. https://doi.org/10.1145/3589335.3651526. http://dx.doi.org/10.1145/3589335.3651526.

Sun, Chi, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. *How to fine-tune bert for text classification?* arXiv: 1905.05583 [cs.CL]. https://arxiv.org/abs/1905.05583.

Team, Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, et al. 2024. *Gemini: a family of highly capable multimodal models.* arXiv: 2312.11805 [cs.CL]. https://arxiv.org/abs/2312.11805.

Terechshenko, Fridolin Linder, et al. 2020. *A comparison of methods in political science text classification: transfer learning language models for politics.* Available at SSRN 3724644.

Torres, Michelle. 2024. A framework for the unsupervised and semi-supervised analysis of visual frames. *Political Analysis* 32 (2): 199–220. https://doi.org/10.1017/pan.2023.32.

Torres, Michelle, and Francisco Cantú. 2022. Learning to see: convolutional neural networks for the analysis of social science data. *Political Analysis* 30 (1): 113–131. https://doi.org/10.1017/pan.2021.9.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. 2023. *Llama: open and efficient foundation language models.* arXiv: 2302.13971 [cs.CL]. https://arxiv.org/abs/2302.13971.

Wang, Yu. 2023. On finetuning large language models. *Political Analysis,* 1–5. https://doi.org/10.1017/pan.2023.36.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. *Chain-of-thought prompting elicits reasoning in large language models.* arXiv: 2201.11903 [cs.CL]. https://arxiv.org/abs/2201.11903.

Wilkerson, John, and Andreu Casas. 2017. Large-scale computerized text analysis in political science: opportunities and challenges. *Annual Review of Political Science* 20 (1): 529–544.

Yan, Dongyang, Keping Li, Shuang Gu, and Liu Yang. 2020. Network-based bag-of-words model for text classification. *IEEE Access* 8:82641–82652. https://doi.org/10.1109/ACCESS.2020.2991074.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, et al. 2023. *Judging llm-as-a-judge with mt-bench and chatbot arena.* arXiv: 2306.05685 [cs.CL]. https://arxiv.org/abs/2306.05685.