Using LLMs for Classification of Text Messages from Experiments

Can Çelebi*[1] and Stefan P. Penczynski[2]

[1] *Vienna Center for Experimental Economics (VCEE), University of Vienna,*
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria. Email: `can.celebi@univie.ac.at`
[2] *School of Economics and Centre for Behavioural and Experimental Social Science (CBESS),*
University of East Anglia, Norwich Research Park, Norwich NR4 7TJ, United Kingdom.
Email: `s.penczynski@uea.ac.uk`*, Tel. +44 1603 59 1796*

November 18, 2025

This study examines the use of large language models (LLMs) for text classification in economics. We investigate whether original instructions can be effectively repurposed as prompts with moderate changes to achieve classification results comparable to human-coded benchmarks. We classify text from four economic experiments on promises and strategic thinking. Covering tasks with varying complexity and prevalence in pre-training data provides insights into how task characteristics influence classification performance. We find that LLMs can accurately and cost-effectively classify text across tasks and replicate human annotations well. Our findings offer guidance for integrating LLM-based text classification into economic research.

Teaser: LLMs achieve performance comparable to humans in text classification using prompts based on the codebooks written for humans.

*Corresponding author.

1. Introduction

Text classification organizes unstructured text into predefined categories and thus increases the scope and depth of scientific insights. Across the social sciences, researchers have long leveraged "text as data," (Gentzkow et al., 2019) to gain insight into decision-making, political discourse, media bias and more (Glavaš et al., 2019; Capra, 2019; Nelson et al., 2021; Fyffe et al., 2024). However, its broader adoption is held back due to the limitations of human annotation, which is costly, time-intensive, and prone to subjective biases (Çelebi and Penczynski, 2025). Machine learning methods offer an alternative but come with their own limitations such as the requirement of classified training datasets for supervised learning (Georgalos and Hey, 2020; Penczynski, 2019; Hüning et al., 2022a) and the limited interpretability of unsupervised learning (Grimmer and Stewart, 2013; Schmiedel et al., 2019). In contrast, large language models (LLMs) provide a cost-effective and flexible alternative, capable of classifying text in data-scarce environments by inferring patterns from at most a few classified examples (Brown et al., 2020).

To effectively leverage LLMs as a text classification tool, we propose that a natural first step is to repurpose classification guidelines (codebooks) designed for human annotators as prompts. This approach ensures standardization, continuity with prior research, and better comparability with human classifications. Furthermore, it minimizes the effort of transitioning from human to LLM annotators by eliminating the need to devise prompts from scratch while keeping classification anchored in the definitions and examples established by domain experts. Moreover, adhering to the original codebooks reduces the risk of ad hoc modifications to instructions and examples, which could lead to overfitting.

Building on this approach, we introduce and apply a cross-task generalizable template based on established prompt engineering standards (Reynolds and McDonell, 2021; Mishra et al., 2021a; White et al., 2023; Clavié et al., 2023; Yuan et al., 2023) to ensure a structured adaptation for LLM use. The template retains the original codebook content with only minimal to moderate formatting adjustments. These adjustments reduces structural and formatting variations across codebooks (Sclar et al., 2023) and help to ensure that differences in LLM performance stem from task variation rather than inconsistencies in prompt design, thereby enabling a more systematic and controlled evaluation.

Our primary research question is therefore, can existing codebooks designed for human annotators be adapted with moderate modifications to serve as effective prompts for LLM-based text classification (RQ1)?

Three of the four original codebooks we consider include classification demonstrations, making the technique of n-shot prompting—providing a few classified examples in the prompt—an inherent feature of our setup. This natural availability allows us to systematically examine n-shot prompting's effect on text classification (Brown et al., 2020). Similarly, zero-shot Chain-of-Thought prompting (0-shot-CoT)—instructing the model to generate intermediate reasoning steps before classifying—can be modularly incorporated into prompts without altering their structure or content, making it another natural candidate for evaluation (Kojima et al., 2022). This raises the question, how do n-shot and 0-shot-CoT prompting affect classification performance (RQ2)?

Furthermore, both techniques, along with the ability to closely follow instructions, scale with larger models, which exhibit stronger reasoning capabilities and improved learning from examples and category definitions (Mishra et al., 2021a; Ouyang et al., 2022; Wei et al., 2022b; Lou and Yin, 2024). We therefore ask: how does model scale affect classification performance and the impact of these prompting techniques (RQ3)?

In this study, we use GPT-3.5 and GPT-4 to classify text from economic experiments and focus on two classification tasks: identifying promises (P) and levels of strategic thinking (L). These tasks differ in their prevalence in the models' pre-training corpora—the text data used to train LLMs. Promise classification is likely to be more prevalent in the training data, as the concept of a promise is intuitive, commonly used in everyday language, and context-independent. By the same token, such a task can arise in any social science domain. In contrast, levels of strategic thinking are a technical construct specific to behavioural game theory (Crawford et al., 2013), making them less likely to appear frequently in pre-training corpora. As a result, the model is more likely to recognize a promise based on its existing knowledge (recognition-heavy task), whereas it is more likely to rely on category descriptions and examples to learn classifying levels of strategic thinking (learning-heavy task) (Pan et al., 2023; Wang et al., 2024a; Lampinen et al., 2024).

The two tasks (P and L) differ in the number of belief iterations performed by the subject and reflected in the text. They also differ in the extent to which classification relies on symbolic reasoning—that is, reasoning grounded in a well-defined formal system (see Sprague et al., 2024). Strategic thinking classification involves semi-symbolic reasoning, as it draws on the level-k model of bounded rationality. In contrast, promise classification does not rely on symbolic reasoning.

Lastly, classification difficulty also varies by experiment within each classification concept (subscripts I and II). In the promise classification, one experiment involves stand-alone messages (P_I), while the other requires classifying multi-player conversations (P_{II}), where the model evaluates the full exchange to determine if a given subject made a promise. In the strategic thinking classification, one experiment assumes a random initial belief with up to three levels of strategic depth (L_I), while the other requires the classification of the initial belief according to payoff or label salience, with a strategic depth of up to five levels (L_{II}). Moreover, in the latter, the task requires identifying both lower and upper bounds of strategic thinking rather than a single level.

These variations in task classification matter for performance and relate our results to the prompt engineering literature, where the effectiveness of both 0-shot-CoT (referred to as CoT onwards) and n-shot prompting has been shown to be influenced by task characteristics and model scale (Razeghi et al., 2022; Wu et al., 2023; Pan et al., 2023). CoT improves performance in multi-step reasoning tasks (Huang and Chang, 2022), but its benefits diminish as task complexity increases (Zheng et al., 2023a; Wang et al., 2023) or as reasoning sequences lengthen (Jin et al., 2024; Prabhakar et al., 2024). Additionally, while CoT enhances performance in tasks requiring semi-symbolic and symbolic reasoning, it negatively impacts those requiring non-symbolic reasoning (Sprague et al., 2024). Moreover, CoT has been found to be more effective with larger models (Wu et al., 2023) and less effective on task that are less prevalent in the pre-training corpora (Razeghi et al., 2022). Similarly, n-shot prompting's effectiveness depends on task type, with performance gains plateauing in recognition-heavy tasks, whereas in learning-heavy tasks, each additional example continues to contribute to improvements (Pan et al., 2023).

2. Materials and Methods

2.1. Data

For the promise classification, we rely on benchmark labels distinguishing between "promise" and "empty talk" in 38 messages from a principal-agent game (P_I, Charness and Dufwenberg, 2006) and 717 chat instances involving 1,491 subjects from a public goods game (P_{II}, Arad et al., 2024). For strategic thinking classification, we use existing labels from 493 messages in a jury voting game (L_I, Çelebi and Penczynski, 2023), which categorise messages into four levels of reasoning (level-0 to level-3). The coordination games dataset (L_{II}, van Elten and Penczynski, 2020) contains 851 messages and features two classification layers: (1) identifying the type of salient reasoning cue—no salience, label salience, or payoff salience—and (2) assessing the level of strategic reasoning (level-0 to level-5). If a message contains label salience, the model must classify its specific form as one of $\S$, $\#$, \$, X, or Y, depending on the game context. If it contains payoff salience, it must be further classified as either "high" or "low". The level classification in this dataset is not provided as a single point label, but as a range indicating the lower and upper bounds of the subject's inferred reasoning level. All texts in P_I, P_{II}, and L_I are in English, while L_{II} is in German (see Supplementary Materials C.1.1, C.2.1, C.4.1, and C.5.1 for details).

Accuracy is assessed based on agreement with human-annotated benchmarks. For P_I, this is sourced from Houser and Xiao (2011), whereas for P_{II}, L_I, and L_{II}, it comes from the same papers as the text data. In L_{II}, we report Jaccard accuracy to account for its interval-based labels (see Supplementary Materials C.5.4 for details).

2.2. Prompts

All classification tasks were implemented via a standardized prompt template designed to ensure consistency across tasks of varying complexity. As shown in Figure 1, each prompt begins with a brief general task description and a role persona. The "General task" section primes the model by anchoring the task within familiar vocabulary (e.g., "classify," "message," "promise"), leveraging the model's pre-trained recognition of these concepts to improve performance (Mishra et al., 2021a). The "Role Persona" section further guides the model's reasoning style by embedding the task within a contextually appropriate domain identity—behavioural economist—a technique shown to enhance output quality by activating the model's learned associations with expert reasoning (Kong et al., 2023). The "Context" section provides an expert-level summary of the experimental setup, decision process, and underlying theory, giving the model sufficient vocabulary and conceptual scaffolding to interpret the messages (White et al., 2023). The "Classification Task" and, when applicable, "Examples" sections are drawn verbatim from the original codebooks. Examples are only used in n-shot prompting; similarly, a "Classification Process" section is added only for 0-shot-CoT prompting and always consists of the same sentence shown in Figure 1, instructing the model to provide a step-by-step explanation before classifying. Finally, the "Classification Coding," "Constraint," and "Output Format" sections ensure standardized outputs that can be easily extracted using simple string matching. All prompts were written in Markdown format and all instructions were structured as itemized lists. Design rationale for each section of the template, as well as for the overall formatting of the prompts, is provided in detail in Supplementary Materials B.1, along with further references to the relevant literature.

P_I was evaluated only in a 0-shot setting, as its original codebook contained no examples. For n-shot prompting, P_{II}, L_I, and L_{II} were evaluated with 11-shot, 19-shot, and 5-shot prompting, respectively, following the examples in their original codebooks. All four prompts used in this study are provided in Supplementary Materials C.1.3, C.2.3, C.4.3 and C.5.3.

```
# General Task
- Classify <X> in <E>

# Role Persona
- Act as a behavioural economist specialized in text
classification, concept <C> and game <G>

# Context
- <Game mechanics>
- <Experimental design/Decision Process>
- <Theory>
- ...

# Classification Task
- Classify <X> as <Y> given criteria:  <$Y_1$, $Y_2$, ...>
- ...

# Classification Coding
- Code <X> as <Z> if it is classified as <Y>
- ...

# Examples
- <Example text> <classification>
- ...

# Classification Process
- Provide a step-by-step reasoning before providing your
classifications.

# Constraint(s)
- Follow the below output format.
- ...

# Output Format
<Desired output format>
```

Figure 1: General Prompt Template

2.3. Model Choice

All classifications were performed using OpenAI's GPT-4-preview-1106 and GPT-3.5-turbo-1106 models. The decision to use proprietary rather than open-source models is a central issue in current methodological debates. While open-source models promote transparency and reproducibility (Spirling, 2023), proprietary systems remain widely used and, under certain conditions, methodologically defensible (Palmer et al., 2024). Researchers relying on closed models are therefore expected to justify their choice explicitly (Palmer et al., 2024). Our justification proceeds along two lines.

First, our objective is to evaluate whether state-of-the-art LLMs can approximate human-coded benchmarks. Since this involves assessing current performance limits, it is methodologically appropriate to use proprietary models widely recognized as leading in the field (Palmer et al., 2024). This aligns with the "technical state of the art" criterion introduced by Palmer et al. (2024), as including models with consistently lower benchmark performance would not yield additional insight into our primary research question (RQ1).

Second, as frontier models scale and broaden, their internal similarity structures converge, and this alignment correlates with downstream task performance (Huh et al., 2024). Accordingly, findings derived from one state-of-the-art model (GPT) are expected to generalize to models with comparable capabilities. This expectation applies at the level of learned representations. However, differences in instruction fine-tuning across different state-of-the-art models can yield model-specific responses, even when representational structures are similar. For this reason, it is preferable to work with a model family for which the literature provides the most comprehensive guidance on prompt construction. Our standardized prompt template (see Figure 1) draws on extensive experimentation in the computer science literature, which overwhelmingly focuses on the GPT model family (see Supplementary Materials B.1). There is no comparable body of work systematically evaluating prompt structures for non-GPT models—whether open-source or proprietary. In the absence of such evidence, using alternative models would require researchers to rely either on unpublished or limited company-specific guidelines, or on best practices that have been validated primarily on GPT models, without knowing whether they generalize. By contrast, our use of GPT is anchored in a well-documented empirical literature, and the effectiveness of our prompt template is indirectly corroborated by OpenAI's own guidance, which aligns with our design choices (MacCallum and Lee, 2025). Importantly, this guidance serves not as a source of our template but as an independent validation of the structure we developed through prior literature. Using non-GPT models would necessitate extensive ablation studies to assess whether our prompt template—or any other—improves or degrades performance. By employing GPT, we leverage a robust experimental foundation and avoid unnecessary duplication. This also positions our contribution at the intersection of social science and computer science by adopting prompting strategies rigorously evaluated in the latter.[1]

[1] We acknowledge that the use of proprietary models raises reproducibility concerns. While we ensure transparency by archiving all materials, future access to the specific model versions used remains uncertain. In response, parts of the academic community have advocated for the development and

2.4. GPT Procedure

The full prompt was provided to either model as the system prompt, with the subject's message to be classified supplied as the initial user input. Each classification was conducted through a separate OpenAI API call. The temperature hyperparameter was always set to 0. Such a setting reduces randomness and increases determinism in model outputs, and thereby improves result reproducibility (see Supplementary Materials A.2 for further discussion). The `max_tokens` hyperparameter was adjusted based on whether CoT prompting was used, and if so, according to the average length of reasoning observed during testing for each prompt. All other hyperparameters were left at their default settings. See Supplementary Materials B.2 for further details on the procedure.

3. Results

	P_I	P_{II}	L_I	L_{II}
∅	96	69	80	66
CoT	**97**	73	80	67
n-shot	–	83	84	**71**
CoT & n-shot	–	**85**	**91**	**71**

Table 1: GPT-4 accuracy (in %), highest values in bold.

Table 1 reports accuracy percentages for each task under different prompting techniques, where ∅ represents a baseline without prompting techniques, and "CoT & n-shot" represents both techniques applied. Notably, results under n-shot prompting correspond to the prompts that most closely follow the original codebooks for P_{II}, L_I, and L_{II}, while ∅ is closest for P_I.

On average, promise classification outperforms strategic thinking classification (91% vs. 81%), reflecting both the greater prevalence of promises in the model's pre-training corpora and differences in task complexity. The impact of complexity on model performance is also evident within each classification type. In promise classification, performance is significantly higher for stand-alone messages in P_I than for classi-

adoption of open-source alternatives (Spirling, 2023; Halterman and Keith, 2024). However, both the development of such models and the experimentation required to identify prompt structures that maximize performance are resource-intensive and environmentally costly. A more pragmatic direction may involve collective efforts by the research community to secure long-term, stable access to proprietary state-of-the-art models for academic use.

fying individual subjects within chat instances in P_{II}. Additionally, within P_{II}, we analysed classification performance across different chat structures, distinguishing between monologues, dialogues, and trialogues. When both prompting methods were present, monologues had the highest accuracy (90%), while dialogues and trialogues had slightly lower accuracies (84% and 85%, respectively). Similarly, in strategic thinking classification, accuracy is higher in L_I, which involves fewer reasoning steps, than in L_{II}, where the model must not only process more steps but also infer both upper and lower bounds for levels of strategic thinking.

Result 1 *Human annotator codebooks can be effectively repurposed for LLM-based text classification across tasks that differ in underlying concepts, reasoning complexity, and message format.*

Table 2 presents the information of Table 1 in terms of the accuracy improvements (in percentage points) of each prompting technique relative to the baseline $\varnothing$. With the exception of P_{II}, CoT alone provides negligible improvements in classification performance.

	P_I	P_{II}	L_I	L_{II}
Δ CoT	1	4	0	1
Δ n-shot	–	14	4	5
Δ CoT & n-shot	–	16	11	5

Table 2: GPT-4 change in accuracy (in percentage points).

In contrast, n-shot prompting has a significant impact on performance improvement across tasks, with the highest gain observed in P_{II}. This result is expected, given that P_{II} has the largest and most balanced set of classification examples (11 examples for promises and 10 for empty talk across 11 chat instances). On the other hand, despite n-shot prompting using nearly four times more examples in L_I than in L_{II}, its impact on accuracy remains similar across both tasks.

Combining n-shot with CoT yields the greatest accuracy gains, except in L_{II}. Notably, while CoT provides no benefit in L_I, it significantly enhances performance when combined with n-shot. However, this synergy does not extend to the more complex level-k classification task in L_{II}.

Result 2 *n-shot prompting substantially and consistently improves classification accuracy, while CoT's impact varies by task. Yet, the highest performance is achieved when both methods are combined.*

Table 3 present GPT-3.5's absolute baseline accuracy (∅) in the first row, followed by the relative accuracy improvements from each prompting technique in the subsequent rows. In small font, the change in accuracy compared to GPT-4 is reported (Table 1).

	P_I	P_{II}	L_I	L_{II}
∅	76 -20	58 -11	70 -10	52 -14
Δ CoT	8 -13	-3 -18	-6 -16	3 -12
Δ n-shot	–	3 -22	-5 -19	6 -13
Δ CoT & n-shot	–	-3 -30	-5 -26	6 -13

Table 3: GPT-3.5 baseline accuracy (∅ in %) and change in accuracy (in percentage points). Change in accuracy compared to GPT-4 (in percentage points).

The consistently negative improvements in small font show that the larger GPT-4 model outperforms GPT-3.5 across tasks and techniques. Unlike in GPT-4, the effects of CoT and n-shot prompting in GPT-3.5 are inconsistent. CoT only improves performance in P_I and L_{II} and negatively affects it in P_{II} and L_I. Similarly, n-shot prompting enhances performance in P_{II} and L_{II} but negatively impacts in L_I. When combined, CoT and n-shot provide no additional improvement over n-shot alone in L_I and L_{II}, while in P_{II}, CoT negates and reverses the gains from n-shot.

Result 3 *GPT-4 consistently outperforms GPT-3.5. While both techniques generally enhance performance in GPT-4, this is not the case for GPT-3.5. Larger models make better use of CoT and n-shot prompting, leading to reliably higher classification performance across prompting techniques.*

Finally, we conclude our results by assessing whether we could do *without* detailed classification instructions from the original codebooks. This is possible in the context-independent promise classifications, where we test a simpler prompt that removes explicit classification criteria, excludes examples in P_{II} and keeps the rest intact. The model is simply asked to classify whether a subject made a promise (see Supplementary Materials C.1.3 and C.2.3 for details). Table 4 presents the change in accuracy relative to the original prompt.

Removing such classification instructions consistently lowers accuracy in GPT-4, with a sharper decline in P_I than in P_{II}, regardless of CoT prompting. Conversely, GPT-3.5 performs better with the simpler prompt, suggesting that detailed instructions may hinder rather than improve its performance. This is likely due to its weaker

		P_I	P_{II}
GPT-3.5	∅	3	2
	CoT	0	6
GPT-4	∅	-12	-1
	CoT	-10	-1

Table 4: Percentage point change in Accuracy when not providing category definitions

instruction-following capabilities, where additional details introduce confusion rather than clarity.

These findings highlight that detailed classification instructions—such as those provided in the original codebooks—are beneficial for GPT-4, leading to higher classification accuracy, while simpler prompts are preferable for GPT-3.5. This suggests that as model capabilities improve, they become better equipped to leverage expert-level, theory-grounded instructions, reinforcing the value of using structured codebooks when applying LLMs for text classification in research.

4. Discussion

This study examined whether classification codebooks originally designed for human annotators can be adapted into effective LLM prompts with moderate modifications. Our findings show that such adaptation is indeed viable: GPT-4 achieved high performance across all tasks, replicating human annotations well using prompts closely derived from original instructions. We view our results as encouraging any researcher confronted with the discussed types of tasks to invest into LLM skills and to reserve human annotation for random sampling validation.

Our findings also show that prompting techniques matter. While n-shot prompting reliably improves classification accuracy, the benefits of CoT are more variable and task-dependent. Moreover, these effects are model-sensitive: GPT-4 consistently benefits from detailed instructions and n-shot examples, whereas GPT-3.5 displays erratic responses to both techniques with CoT occasionally even offsetting gains from using n-shot prompting.

Beyond accuracy, LLM-based classification is cost-effective and fast. Even in the most complex case—L_{II} with CoT using GPT-4—100 messages were classified in 24 minutes at a cost of only $3.40, a cost that has since fallen fourfold. These efficiencies underscore the potential for LLMs to lower the barriers to large-scale text classification, providing an accessible alternative to traditional, labor-intensive methods.

Original codebooks offer a natural and efficient starting point for constructing LLM prompts, but they are not written with LLMs in mind and require editing to align with principles of effective prompt engineering (Supplementary Materials B.1). One such principle relates to the composition of examples, as the effectiveness of n-shot prompting is sensitive to the number, order, and balance of examples across categories (Pan et al., 2023; Lu et al., 2021a; Zhao et al., 2021). In L_{II}, for instance, performance may be improved by addressing category imbalances via additional examples in underrepresented classes (see Supplementary Materials C.5.3 for details).

Beyond prompt engineering, certain model hyperparameters, such as temperature, should be set mindfully, in line with the task's objectives. For text classification, where consistency and replicability are essential, the model's temperature should be set to zero. This aligns with best practices in LLM research (Kojima et al., 2022; Wei et al., 2022a; Kosinski, 2024; Kostina et al., 2025) and with OpenAI's official recommendations for classification tasks (OpenAI, 2023a). A notable exception is the self-consistency method in CoT prompting, where a positive temperature is used to generate diverse reasoning paths, and classifications are aggregated across generations to improve performance (Wang et al., 2022). However, if the goal is to assess model consistency by repeatedly classifying the same text at positive temperatures—as done in Gilardi et al. (2023); Törnberg (2023); Pangakis et al. (2023); He et al. (2024)—a more cost-efficient alternative is to retrieve the model's probability distribution over classification categories (See Wang et al., 2021, for an application of the use of these probabilities in text classification).

While our results show that LLMs can match human annotators in text classification, we also observe a decline in model performance as task difficulty increases. To maintain high accuracy in such relatively difficult cases—while minimizing the costs associated with human annotation—researchers can leverage hybrid "algorithm-in-the-loop" approaches, where human annotators intervene only when model confidence—typically based on the aforementioned output probabilities—falls below a certain threshold. This approach has been shown to improve performance, particularly in tasks where the LLM alone underperforms relative to human annotators (Bansal et al., 2021; Wang et al., 2024b; Vaccaro et al., 2024).

Lastly, researchers may encounter classification settings with longer input texts (e.g., news articles, legal documents) or more detailed codebooks than those used in our study. Both cases lead to longer prompts, which may lead to performance degradation (Liu et al., 2024). To address these challenges, we suggest a "divide-and-classify" strategy. This may involve segmenting long texts into smaller, coherent units or breaking multi-category codebooks into category-specific prompts, followed by an aggregation step. Decomposition strategies such as these have been shown to improve LLM performance across a range of tasks (Patel et al., 2022; Mishra et al., 2021b; Khot et al., 2022; Liu and Tan, 2023; Srivastava and Gandhi, 2024).

Acknowledgment

Funding:

Center for Doctoral Studies in Economics (CDSE), University of Mannheim (CC).
Austrian Science Fund (FWF) grant 10.55776/PAT2819324 (CC).

Author Contributions:

Conceptualization: CC, SPP
Methodology: CC, SPP
Investigation: CC, SPP
Visualization: CC, SPP
Supervision: SPP
Writing—original draft: CC, SPP
Writing—review & editing: CC, SPP

Competing Interests:

Authors declare that they have no competing interests.

Data and materials availability:

All the prompts used, the original codebooks they are derived from, and sample classification outputs are available in the Supplementary Materials. All other data will be made available upon request and will be deposited to OSF upon publication.

Supplementary Materials for
Using LLMs for Classification of Text Messages from Experiments

List of Figures

List of Tables

A. LLMs and related literatures

An LLM is a statistical language model trained on a large corpus to predict the next word for any given textual input. By inputting text instructions, one can strategically leverage this predictive capability to steer the model's output towards a desired outcome, a practice commonly referred to as prompting. The appeal of prompting stems from the ease with which natural language allows us to convey complex ideas. Yet, this very flexibility may introduce inaccuracies or ambiguities if concepts are not clearly defined or presented with insufficient context. The effectiveness of a prompt hinges both on the user's adeptness at crafting instructions with clarity and contextual relevance, and on the model's ability to accurately *interpret* these instructions within their context. While an LLM's capacity to process text and follow instructions are fundamentally based upon its pre and post training and its parameter size, for downstream tasks, the user can still attempt to refine her mode of interacting with the LLM by engineering her prompts to align more closely with model's operational framework, in order to effectively leverage its capabilities (Reynolds and McDonell, 2021).

A.1. GPTs in the computer science literature

FLAN (Wei et al., 2021), OPT (Zhang et al., 2022b), and PaLM (Chowdhery et al., 2023) are examples of LLMs that have showcased remarkable proficiency in natural language understanding (NLU) tasks (Ye et al., 2023). Particularly, the Generative Pretrained Transformer (GPT) series (Brown et al., 2020), more specifically GPT-3 and its subsequent iterations GPT-3.5, GPT-3.5-turbo, GPT-4 and GPT-4-turbo introduced by OpenAI have sparked considerable attention due to their exceptional performance in integrating various NLU tasks into generative ones (Ye et al., 2023).

Earlier GPT models, GPT-1 and GPT-2, are limited in their ability to recognise textual patterns across diverse tasks due to their relatively smaller training corpus and parameter size (Radford et al., 2019). Consequently, these models require substantial fine-tuning on task-specific datasets to achieve satisfactory performance. Yet fine-tuning poses several problems: first, it requires large volumes of task-specific data; second, there is a risk that these training datasets do not cover the full spectrum of task variations, which could lead to suboptimal performance on data not represented within the training set (lack of generalizability due to over-fitting) (Brown et al., 2020). Furthermore, fine-tuning an LLM on data that introduces new knowledge is documented to increase the model's likelihood to make up information (hallucinate) (Gekhman et al., 2024).

Building on its predecessor, GPT-3 has been trained on a significantly larger corpus, consisting of approximately 400 billion tokens, compared to GPT-2 which was trained on 1.5 billion tokens (Brown et al., 2020). This extensive training has markedly enhanced its ability to detect diverse textual patterns (Brown et al., 2020) and has enabled reasoning-like emergent qualities (Wei et al., 2022b). Notably, GPT-3 can perform specialised tasks when provided with few examples demonstrating how to perform it (Brown et al., 2020). This capability, which significantly reduced the need for parameter adjustments through fine-tuning, catalysed the development of the In-Context Learning (ICL) paradigm for LLMs comparable in size to GPT-3 or larger (Dong et al., 2022).

In the ICL paradigm, the process of demonstrating task execution through a small number of input-output pairs, where the input serves as the question and the output as the answer, is referred to as n-shot prompting. In the specific case of classification tasks, the term “n” indicates the number of examples provided in the prompt. n-shot prompting quickly gained popularity as it requires only a few demonstrations to guide the model toward achieving performance comparable to that of fine-tuned models trained on extensive datasets. Notably, a single demonstration can be as effective as fine-tuning the model with approximately 300 to 3,500 input-output pairs, depending on the task (Scao and Rush, 2021). Additionally, by modifying the format of the demonstrations from <input, output> to <input, reasoning, output>, one can enable the model to demonstrate reasoning capabilities. This approach, referred to as n-shot-CoT (Chain of Thought), has proven to significantly enhance the models’ performance, especially on more involved tasks that can potentially benefit from multi-step reasoning (Huang and Chang, 2022).

Although the ICL paradigm offers a flexible and data-efficient way to “teach” the model at inference, the efficacy of the method on improving the model’s task performance, or in other words the model’s ability to “learn” from these demonstrations, relies to a greater extent on the choice of examples, the sequence in which they are presented within the prompt, and the frequency with which examples for each category to be classified are provided (Lu et al., 2021b; Kumar and Talukdar, 2021a; Zhao et al., 2023). Although various methods to select the optimal set of examples and their order have been proposed, and documented to improve the model’s performance (Li and Qiu, 2023a; Su et al., 2022; Liu et al., 2021; Luo et al., 2023; Chang and Jia, 2022), their technically demanding procedures may be less accessible for social scientist to implement. This lack of accessibility juxtaposes with the appeal of GPT’s out-of-the-box usability, which we believe is necessary for any prompting method to be widely adopted by social scientists.

Given n-shot-CoT prompting is an extension to n-shot prompting, its effectiveness in enhancing the model's task performance also relies on the choice and order of examples. In addition, however, its effectiveness is also dependent on the manner with which the rationales are provided for each demonstration (Huang and Chang, 2022). To improve the reliability of n-shot prompting, the most cited method is the "self-consistency" method proposed by Wang et al. (2022). In this method, the temperature hyperparameter is set to a strictly positive value (e.g. 0.5 or 0.7) in order to increase the variability in the reasoning sequence of the output, and multiple request (e.g. 20 or 40) to the model are made using the n-shot-CoT method for every single task instance. Then the most frequently outputted answer is picked as the decisive (consistent) output. Noting that requesting a reasoning prior the answer already puts additional token cost, by generating k many outputs with reasoning for each task instance, "self-consistency" method is increasing the computational cost approximately k times more. Making the classification task approximately k times more costly to improve the reliability of the model's output could make this approach prohibitively expensive for social scientists, and potentially deter them from viewing it as a viable alternative to human annotators. To refine the reasoning provided with each examples, it has been suggested, among many, to use multiple human annotators to provide a diverse set of reasoning for each example (Li and Qiu, 2023b), to use the LLM itself to generate a diverse sets of reasoning for each examples and then select the reasoning with the most steps (Fu et al., 2022), or to generate multiple outputs using the generic n-shot-CoT prompting method and to choose the most frequently provided output as the final output of the model (Wang et al., 2022) (see Huang and Chang (2022) for details and other methods to refine the rationale). Arguably, although these proposed methods for refining the reasoning in demonstrations are not computationally demanding, they still require careful selection of the most suitable reasoning for each example, and may lead the researchers to doubt whether the explanations considered were adequate, particularly when the model's performance does not meet their expectations.

In addition to the unreliability concerns with n-shot and n-shot-CoT prompting, it is also unclear how the model "learns" at inference to perform a task via few demonstrations of the task, and whether it genuinely learns via demonstrations (Reynolds and McDonell, 2021). Min et al. (2022) demonstrate that even when the output labels of the input-output pairs in n-shot prompts are replaced with incorrect labels, the model's performance remains unaffected. They suggest that models do not learn from demonstrations in the same way humans do from examples; rather, these examples primarily serve to delineate the label space and the distribution of the input text, thereby aiding the model in task execution[2]. Reynolds and McDonell (2021) argue that demonstrations do not actually "teach" the model how to perform the task but simply enables the model to locate the tasks in the model's existing knowledge of tasks that it acquired during its pre-training[3] (meta-knowledge). Similarly, with n-shot-CoT prompting, it is unclear whether the model genuinely engages in reasoning and, if so, how this reasoning improves the task performance (Madaan and Yazdanbakhsh, 2022). Furthermore, the ability of a model to reason effectively has been shown to correlate with the frequency of a task's presence in its pre-training corpus: the more frequently a task is represented in the training data, the more likely the model is to exhibit sound reasoning and produce accurate outputs (Razeghi et al., 2022).

Given the aforementioned challenges with n-shot-CoT method, a notable, demonstration-free alternative is the 0-shot-CoT prompting method (Kojima et al., 2022). This approach simply involves appending the phrase "Let's think step-by-step" to the instructions and thereby triggers a reasoning process before generating the output. 0-shot-CoT, devoid of the reliability concerns regarding the selection of examples or the quality of reasoning, is task-agnostic and can seamlessly be integrated into an existing prompt by adding the keyword "think step-by-step" into the specific sections of the instructions where the user wishes to invoke reasoning steps in the model (OpenAI, 2023b). While 0-shot-CoT has been shown to significantly enhance performance across a range of tasks, its efficacy diminishes with relatively more involved tasks that potentially requires an explicit outline of the reasoning steps to be followed by the model. In such scenarios, the method often yields suboptimal outcomes due to the model's failure to accurately execute or complete the necessary reasoning steps, either by omitting steps or by making errors within specific steps of the reasoning process (Zheng et al., 2023b; Wang et al., 2023). Consequently, alternative 0-shot reasoning-invoking methods have been proposed to address these limitations (Huang and Chang, 2022). Furthermore, 0-shot-CoT's effectiveness is similarly influenced by the model's pre-training corpus. Tasks less represented within the training corpus is observed to provide diminished performance improvements when the method is integrated. This

[2] Yoo et al. (2022) revisited the assertions of Min et al. (2022) and found instances where employing incorrect output labels adversely affects model performance. Consequently, the question of whether and how models learn from demonstrations, and precisely what they learn, remains an open question.

[3] As a consequence, In-Context learning is also referred to as priming (Webson and Pavlick, 2021). Yet, the term priming encompasses a broader spectrum of prompting techniques. For instance, priming can also be done by pre-pending a prompt with a series of token (instead of or in addition to the input-output demonstration pairs) that do not necessarily make intuitive sense (Kumar and Talukdar, 2021b).

highlights, once again, the dependency of the model's performance on the attributes of its training corpus (Wu et al., 2023).

OpenAI has progressively enhanced GPT-3 by fine-tuning on a collection of instruction-answer pairs[4] (Ouyang et al., 2022). These improvements enabled the model to more closely follow instructions, and reduced to a considerable degree the need to instruct the model to perform a task via demonstrations (Chung et al., 2022). Consequently, recent literature has begun to emphasise the instruction learning paradigm which shifts focus from learning through demonstrations to learning via instructions (Lou and Yin, 2024). It's worth noting that this paradigm does not preclude the inclusion of demonstrations within the instructions; rather, it puts more weight on structuring and designing prompts that combine instructions and examples to optimise the model's performance (Lou et al., 2023). There are only a few studies that aim to provide cross-task generalizable prompt design tips intended to enhance demonstration-free instructions to be considered in prompt engineering (Mishra et al., 2021a; Reynolds and McDonell, 2021; Mishra et al., 2021b; Gu et al., 2022; White et al., 2023; Peskine et al., 2023). Additionally, as of writing of this paper, only a single paper has systematically explored various instructional prompting techniques across a wide range of tasks and documented each instructional design component's contribution on improving the model's performance (for a more detailed discussion and the application of these design choices, see Section B.1) (Mishra et al., 2021a).

The concept of instructing an LLM in a manner akin to how one might instruct a human is appealing because it renders the act of prompting both intuitive and flexible. Furthermore, prompting via detailed instructions, free of examples, circumvents various methodological issues inherent with the ICL paradigm. Yet, the availability and diversity of instructions in the training corpus of the model is also observed to be a determinant factor on the effectiveness of 0-shot instructions on improving the model's performance. When instructions tailored for human annotators (such as mTurkers on Amazon Mechanical Turk) are considered verbatim as prompts to assess the model's ability on following human-tailored instructions (turking test), the smaller GPT-2 model has demonstrated poor performance(Efrat and Levy, 2020), while the larger GPT-3 model, when fine-tuned on a large set of human-tailored instructions, has demonstrated the ability to effectively follow unseen human-tailored instructions

[4] Fine-tuning GPT-3 to better follow instructions resulted in the development of GPT-3.5 (Ouyang et al., 2022). GPT-3.5 was then further fine-tuned using reinforcement learning from human feedback (RLHF) method to further enhance its capacity to understand instructions and to better engage in conversational interactions with its users (Ye et al., 2023). These advancements led to the creation of GPT-3.5-Turbo, the underlying model of the ChatGPT application.

(Mishra et al., 2021a; Ouyang et al., 2022). Moreover, human-tailored instructions are documented to outperform basic prompts that instruct the model with one or two sentences devoid of any additional descriptive context for categories (Mishra et al., 2021a). In brief, irrespective of how well instructions are constructed to improve the model's performance, the size of the model and whether the model was fined-tuned on instructions are observed to play a major role in the model's performance; and if the model is fine-tuned on instructions, then using instruction provides an improvement over the performance of the model.

Similar to the ICL paradigm, where it is uncertain whether the model genuinely learns from demonstrations, it is also unclear whether the model truly grasps the task's context and execution conditions from a set of 0-shot instructions (Webson and Pavlick, 2021). If the model truly learns from instructions, variations between two different sets of instructions that convey the same meaning should not affect its performance. However, it has been observed that, without sufficient fine-tuning on task-specific examples, changes in word choices that preserve semantic textual similarity in 0-shot instruction prompts can impact model performance as significantly as training it with an additional 200 task-specific examples (Puri et al., 2022). On the other hand, when the model is fine-tuned with a large collection of task-specific examples, its performance demonstrates robustness to variations in the wording of the instructions (Puri et al., 2022). These results, on the one hand, demonstrates the importance of providing carefully designed instructions, while, on the other hand, hints at the fact that the model does not only learn from a set of instructions but also leverages the provided descriptions to locate the task on its existing knowledge. Lastly, demonstrations and carefully provided descriptions for each classification category are observed to complement each other. Irrespective of whether the model is sufficiently trained on task-specific instructions, when instructions are supplemented with a few demonstrations, the model's performance is observed to remain stable despite variations in the choice of words and phrases (Gu et al., 2022).

A recurring theme in our discussion of various prompting techniques is that the effectiveness of any such technique in enhancing a model's performance largely depends on how well the task's contextual components are represented in the model's training corpus. Certain elements of any given task might already be familiar to the model, while others may be novel. This distinction categories any given task as either a recognition task, where the model identifies elements it has seen before, or as a learning task, where the model encounters new contextual elements. Through a series of carefully designed experiments, Pan et al. (2023) demonstrate that the marginal effect of additional task examples in a prompt diminishes for recognition tasks, since only a few examples are observed to be sufficient for the model to recognize the task, and any additional examples do not enable the model to "further recognize" it. Conversely, for learning tasks, the effect of additional demonstrations is observed to be somewhat linear, with each additional example helping the model grasp more of the task's contextual nuances a bit more. Another distinction between recognition and learning tasks identified by Pan et al. (2023) is the scale of the model[5]. The model size is observed not to significantly enhance its performance on recognition tasks, whereas it is observed to be a crucial factor for the model's ability to learn from demonstrations when faced with a novel task (Pan et al., 2023). This observation is supported by Wei et al. (2022b), who argue that as the model scales, it acquires an emergent capability to learn from the examples. Lastly, it is important to note that any given task may consist of subtasks that fall into two either recognition or learning task category. Furthermore, certain contextual elements of a task might be categorised as learning tasks, while others are more appropriately considered as recognition tasks. Thus, both the size of the pre-training corpus and the scale of the model are potentially crucial factors that impact the performance of the model for a given task. Yet this impact may vary depending on the proportion of recognition to learning components within the task.

Consequently, recognising that the impact of prompting techniques on model performance can vary significantly with each specific task is crucial, as this variability necessitates a case-by-case investigation of the effectiveness of different prompting techniques and model configurations. Therefore, it is unreasonable to universally generalise that one prompting technique or a model with a larger training set or more parameters will consistently perform better across all tasks.

A.2. GPTs in the social science literature

In Table S1, we compile a selection of studies that explore the annotation capabilities of various GPT models. Although this list is not exhaustive, it effectively showcases the diverse prompting techniques used by researchers in a range of annotation tasks within the social sciences.

Despite well-established guidelines from computer science literature, there is a noticeable oversight in the related social science literature with respect to the integration of various prompting strategies such as instruction learning, n-shot prompting, 0-shot-

[5] The scale or size of a model refers to the number of parameters in its neural network. Largest GPT-2 model has approximately 1.5 billion parameters (Radford et al., 2019) while GPT-3 has approximately 175 billion parameters (Brown et al., 2020), and although not exactly known, GPT-4 is estimated to have over 1 trillion parameters (Baktash and Dawodi, 2023).

Table S1: Papers in Social Sciences

Paper	Field	GPT	Temp.	Prompt	Shot	CoT
Rytting et al. (2023)	Poli. Sci.	3	?	Structured*	$2, 3$	
Chae et al. (2023)	Poli. Sci.	3	0	Basic*	$0, 1, 2$	
Reiss (2023)	Poli. Sci.	3.5	$0.25, 1$	Basic$^*_+$	0	
Gilardi et al. (2023)	Poli. Sci.	3.5^*	$0.2, 1$	Original	0	
Zhu et al. (2023)	Psychology Poli. Sci.	3.5^*	?	Basic	0	
Li et al. (2024)	Poli. Sci.	3.5^*	$0, 1$	Original Basic*	0	$\sim$
Zhang et al. (2022a)	Poli. Sci.	$\widetilde{3.5}^*$	~ 0.7	Basic	0	
Aiyappa et al. (2023)	Poli. Sci.	$\widetilde{3.5}^*$	~ 0.7	Basic	0	
Kuzman et al. (2023)	Linguistics	$\widetilde{3.5}^*$	~ 0.7	Basic*	0	$\sim$
Zhong et al. (2023)	Linguistics	$\widetilde{3.5}^*$	~ 0.7	Basic*	$0, 1, 5$	✓
Amin et al. (2023)	Psychology	$\widetilde{3.5}^*$	~ 0.7	Basic	0	
Bhat et al. (2023)	Psychology Linguistics Poli. Sci.	$\widetilde{3.5}^*$	~ 0.7	Basic	0	$\sim$
Heseltine et al. (2024)	Poli. Sci.	$\widetilde{4}^*$	~ 0.7	Basic*	0	
Törnberg (2023)	Poli. Sci.	4	$0.2, 1$	Basic	0	
Pangakis et al. (2023)	Psychology Linguistics Poli. Sci.	4	0.6	Original*	0	
Savelka et al. (2023)	Law	4	0	Original & Structured*	0	✓
He et al. (2024)	Linguistics	4	$0.2, 1$	Original	0	
Rathje et al. (2023)	Psychology	$3.5^*, 4^*$	0	Basic*	$0, 1$	
Matter et al. (2024)	Sociology	$3.5^*, 4^*$	0.1	Basic*	0	$\sim$
Ziems et al. (2024)	Psychology Linguistics Poli. Sci.	$3.5^*, 4$	0	Basic$_+$	$0, 1$	
Our Paper	Economics	3.5^*, 4^*	0	Basic$_+$, Original & Structured*	$0 - 19$	✓

Notes: The "Field" column represents the broad field under which the annotation tasks can be categorized. In the "Model" column, the asterisk indicates that the turbo version of the model is used, and the tilde indicates that the model is not leveraged via the API but through the ChatGPT platform. In the "Prompt" column, "Basic" indicates a basic instructions to classify a text, "Basic$_+$" indicates a basic instruction accompanied by a short definition of for each category, " Original" indicates that the original human instructions are used verbatim as the prompt, and "Structured' indicates that a prompt template is used to structure the prompt into distinct components such as instructions, context, definitions, examples and so on. Moreover, in the "Prompt" column, asterisk superscript indicates that the study investigated either to improve the model's performance via restructuring or augmenting the prompt through rephrasing, incorporating additional context or definitions, making the instructions more precise, etc. or to investigate the effect of a specific variation on the prompt such as considering the prompt in an other language, instructing the model to output a non-binary classification, etc. The "Shot" column indicates the number of demonstrations used in the prompt (n-shot prompting). The "Temp." column indicates the temperature parameter(s) used for the respective model(s), question mark indicates that this value is not provided in the respective paper. Moreover, the exact temperature value for ChatGPT is not known and 0.7 the unconfirmed yet commonly assumed value for it. The "CoT" column not only indicates whether the study used some form of chain-of-thought prompting technique (✓) but also points out studies that considered asking for an explanation after the classification is done ($\sim$) either as an attempt to improve the performance or to further investigate the outputs provided.

CoT, and the proper usage of the temperature hyperparameter. Our aim is to provide insights relevant to our current work and highlight methodological oversights such as the misuse of the temperature hyperparameter (Reiss, 2023; Törnberg, 2023; Gilardi et al., 2023; Pangakis et al., 2023; Matter et al., 2024; He et al., 2024; Li et al., 2024), misuse of 0-shot-CoT (Zhu et al., 2023; Kuzman et al., 2023; Li et al., 2024), classifying messages in batches (Zhang et al., 2022a; Amin et al., 2023; He et al., 2024; Heseltine and Clemm von Hohenberg, 2024; Matter et al., 2024), and using chatGPT rather than the underlying GPT model (Zhang et al., 2022a; Kuzman et al., 2023; Zhong et al., 2023; Amin et al., 2023; Bhat and Varma, 2023; Heseltine and Clemm von Hohenberg, 2024). If unaddressed, these oversights could compromise the perceived utility of LLMs in text annotation tasks and could misdirect the literature towards suboptimal prompting practices. Moreover, certain papers are often cited for their claims that GPT models are unreliable in text annotation tasks, yet their conclusions rest on methodologically questionable practices (Reiss, 2023; Savelka et al., 2023). Meanwhile, other papers advocate for the adoption of specific prompting methodologies, but these recommendations either lack clarity (Pangakis et al., 2023), or show inconsistencies between proposal and practice (Ziems et al., 2024). Consequently, it is crucial to examine these studies more closely to ensure that they do not mislead future research or get perpetuated uncritically in subsequent works.

In Table S1, under the "Prompt" column, the "Basic" tag is used for studies that employ a very basic prompt format such as "Classify X as $Y_1, Y_2, Y_3, ...$". We labeled slightly more involved basic prompts as "Basic$_+$". These prompts either provide additional context for the task, "Given context C, classify ...", invoke a specific persona from the model "Act as R and classify ...", or offer explanations for each classification category, "Classify X as $Y_1, Y_2, ...$ where $Y_1 :< description >, Y_2 :< description >, ...$ ". The "Structured" tag is assigned to prompts that imposes a structured template that organises explanations, context, constraints, demonstrations and additional prompting techniques into modular components via textual cues such as titles or delimiters. The "Original" tag is assigned to prompts that verbatim use instructions tailored for human annotators. Consequently, using this tag in conjunction with the "Structured" tag indicates that the original instructions have been reframed and restructured for its use as a prompt, and a markup language is leveraged to impose this structure. All the papers with a "Structured" tag in Table S1 used the `Markdown` language to structure their prompts. Lastly, the superscript "*" is used to denote studies that explored variations on their initial prompts to enhance model performance either by rephrasing, adding further information, simplifying existing descriptions or using established prompt engineering techniques such as CoT.

A major determinant of a model's task performance is the nature of the task itself, which can be considered under two main dimensions: the representational depth of task-specific categories in the model's pre-training corpus and the complexity of the task. Representational depth reflects the frequency and variety with which the categories to be annotated are represented in the training corpus. A greater representational depth ensures that the model is exposed to a wider range of conceptual diversity for given a category which, in turn, impacts the model's recognition and learning capabilities from provided demonstrations or descriptions for a given annotation task (Reynolds and McDonell, 2021; Razeghi et al., 2022; Pan et al., 2023). Zhu et al. (2023) document that GPT-3.5 performs relatively poorly when tasked with classifying topics that occurred after its training, such as the Ukraine-Russia war. In a similar vein, for GPT-4, Ziems et al. (2024) report strong performance in tasks involving categories common in everyday conversations, such as "anger" in an emotion recognition task, while in tasks requiring expert knowledge and involving non-conventional categories, such as "white grievance" in hate speech classification, GPT-4's performance is notably weaker.

From an information perspective, a task that requires a more diverse set of information is considered more complex (Liu and Li, 2012). Complexity can also be defined by the level of abstraction necessary and the extent of inferential reasoning needed to effectively interpret and act on information (Yang et al., 2016). For example, while utterance-level classification involves analysing individual statements, conversational-level classification should be considered as more complex as it requires understanding the broader context and dynamics within entire dialogues (Arad et al., 2024). Similarly, analysing court opinions to interpret legal concepts (Savelka et al., 2023) or classifying nth level strategic thinking in jury voting (Çelebi and Penczynski, 2023) involves far more complex cognitive processes than identifying promises (Charness and Dufwenberg, 2006). This increased complexity necessitates a model that not only understands expert specific information but also integrates and reasons about it in a manner that emulates higher-order cognitive processes (Huang and Chang, 2022). For instance, Bhat and Varma (2023) investigate the annotation performance of GPT-3.5 across three tasks of varying complexity and find that the model performs poorly with linguistically more challenging task of news category classification (51% average accuracy) compared to sentiment analysis (84% accuracy). Similarly, Savelka et al. (2023) observe poor GPT-4 performance in the task of analysing court opinions to interpret legal concepts (46% average accuracy) yet this performance is found to be still on par with expert level annotators. Lastly, Ziems et al. (2024) document that as the complexity of the text increases, moving from standalone messages to conversational texts, the performance of both GPT-3.5 and 4 models deteriorate in classification tasks. In sum, the results corroborate that the performance of GPT-3.5 and GPT-4 models are shaped by both the depth of representational coverage in their training data and the complexity of the tasks they are assigned.

An additional aspect of representational depth is the linguistic diversity within the model's pre-training corpus. While specific details are often undisclosed, it is widely inferred that the primary training data for GPT-3.5 and GPT-4 consist predominantly of English texts (Lai et al., 2023). The limited representation of multilingual data can in turn cause models' performance on annotation tasks to deteriorate primarily on under-represented language families with a syntactic order in the form of Subject-Object-Verb (SOV, e.g., Hindi, Turkish, Arabic or Amharic) compared to the family of languages with syntactic order in the form of Subject-Verb-Object (SVO, e.g., German, Italian, Spanish or Slovenian) that English is a part of (Bjerva et al., 2019). This is because variations in syntactic order alter the word inter-dependencies crucial for models' language comprehension (Bender, 2011; Nivre et al., 2016). These structural differences result in unique word co-occurrence patterns and grammatical dependencies that LLMs, primarily trained on English, rely on to infer the semantics of the text. Consequently, GPT models may face greater difficulties in effectively processing the semantics in language families with syntactic structures distinct from English, while languages with a similar syntactic order, such as German, may present fewer challenges (Conneau and Lample, 2019). In practice, Heseltine and Clemm von Hohenberg (2024) find that GPT-4 demonstrates consistent performance across tasks in German, Italian, and Chilean Spanish compared to English. Conversely, Bhat and Varma (2023) observe that GPT-3.5 struggles with Indic languages, although it was not tested against English texts to ascertain if the prompts would have fared better in English. Rathje et al. (2023) report that GPT-3.5-turbo and GPT-4-turbo show comparable performance in Turkish, various African languages[6], and Arabic. Kuzman et al. (2023) document that while Slovenian texts are classified as effectively as English texts by these models, prompts in Slovenian yield poorer results compared to their English counterparts. The empirical findings largely corroborate the theoretical expectations that various GPT models perform better with languages syntactically similar

[6] African languages display diverse syntactic structures across several language families. The Niger-Congo family mainly uses SVO order, similar to English, but variations like SOV are found in some Bantu languages. Nilo-Saharan languages typically feature SOV order, with dialectic variations (Vossen and Dimmendaal, 2020).

to English, as observed with German, Italian, and Spanish. However, the comparable performance in SOV languages, like Turkish or Arabic, suggests that the models have the capability to effectively adapt to diverse linguistic structures.

The 0-shot improvement technique of providing context and label descriptions on a “Basic” prompt is shown to improve the classification performance of the models (Peskine et al., 2023). Chae and Davidson (2023) initially consider a simpler prompt than the “Basic” prompt to instruct GPT-3 to classify political stance of Twitter messages with the prompt “Stance:”. This method led the model to incorrectly output labels associated with sentiment classification (“positive”, “negative”) rather than the intended stance labels (“support”, “oppose”). Consequently, they opted for the “Basic” prompt to ensure the correct classification labels, and then tested two incremental improvements on the “Basic” prompt: first, by adding a statement indicating the potential expression of a stance, and second, by providing a general definition of what “stance” entails. These modifications improved the prompt’s performance by 11%. Similarly, Heseltine and Clemm von Hohenberg (2024) started with a “Basic” prompt and enhanced it by adding a single-sentence description for each category, which on average boosted the model’s performance by 8% across different tasks.

The 0-shot improvement technique of reframing instructionsâby clarifying category definitions or making existing instructions more preciseâhas been shown to enhance model performance (Mishra et al., 2021b). For example, Savelka et al. (2023) began with an “Original Structured” prompt but found that overly broad category definitions hindered classification performance. By refining these definitions, they improved model accuracy by an average of 28% across multiple prompts.

Matter et al. (2024) started with a “Basic” prompt that included three main labels (explicit violence, implicit violence, and non-violence) and subcategories for violence cases. After testing on a subset of the dataset, they enhanced the prompt by adding brief descriptions for each category. They then further refined the prompt through iterative testing on new subsets, using GPT-generated text to address observed misclassifications. The final prompt included category-specific descriptions and two examples. Although they report substantial performance gains, the absence of evaluations on the full dataset means the magnitude of each refinementâs contribution remains unclear. Moreover, the rationale for providing additional clarification to only a subset of the categories is not explicitly discussed.

A related approach is proposed by Pangakis et al. (2023), who iteratively compared GPT classifications to human annotations on randomly selected subsets. When the model’s output substantially divergedâthough no threshold was definedâthey adjusted the prompts and repeated the process until the results were deemed "satisfactory". While the procedure reportedly improved performance, specific metrics for improvement or details of the intermediate prompts were not provided, limiting reproducibility. Although Matter et al. (2024) and Pangakis et al. (2023) provide case studies highlighting the importance of effective prompting on model performance, their methods risk overfitting on a subsample of the messages, which could not only fail to improve but also potentially degrade classification performance on the rest of the data. Moreover, their prompt enhancement procedures are not extensively documented. In Matter et al. (2024), the final prompt must be examined closely to infer the prompting techniques used, and the rationale behind the inclusion of additional descriptions or examples for certain labels is not clearly explained. In Pangakis et al. (2023), the final prompts and details of the refinement process are not reported. Consequently, the studies offer limited guidance for researchers seeking replicable prompt engineering strategies.

Attempts to enhance model performance using either 0-shot-CoT or n-shot-CoT prompting methods are observed in social science applications, albeit in very few instances, despite the accessibility of the 0-shot-CoT method and detailed guidelines on its use (OpenAI, 2023b). Depending on the complexity of the task, CoT is observed to have a varying improvement on the model's performance. Zhong et al. (2023) document that 0-shot-CoT increases the performance by 15%, 1-shot-CoT increases performance by 8%, and 5-shot-CoT by 21%. Yet, they note that generating the reasoning examples for n-shot-CoT prompts were challenging. They have first constructed a hand written reasoning example for an input then instructed GPT to provide similar reasoning demonstrations for other inputs which they used as the additional four reasoning examples in their 5-shot-CoT prompt. Conversely, in Savelka et al. (2023) where the task is argued to be more complex, 0-shot-CoT is documented to worsen the performance of the initial prompt by 13% and only to improve the performance of the improved prompt by 4%. In addition to the established CoT prompting techniques, we identified various studies where the model is prompted to provide reasoning after the classification. In Table S1, studies employing this post-classification reasoning approach are indicated in the "CoT" column with a tilde ($\sim$). Asking the model to reason after providing the response is documented to either not improve or provide a minor improvement to the model's performance compared to a baseline of no CoT (Wei et al., 2022a). Therefore, future studies should consider avoiding this methodological oversight to avoid suboptimal model performance.

Few studies have explored n-shot prompting. Chae and Davidson (2023) experimented with various examples for 1-shot and 2-shot prompting and documented that the choice of example significantly affects model performance, with F1 scores on average varying between 42% and 73% across tasks. This variance corroborates concerns voiced in the computer science literature about the high dependency of performance on the choice of examples (Zhao et al., 2023). Rathje et al. (2023) implemented 1-shot prompting across five tasks, with performance deteriorating in two tasks and slightly improving in three, resulting in an average performance improvement of 3.5%. These outcomes emphasise the majority label bias, where the model tends to favor labels that appear more frequently in the demonstrations used in the prompt. Consequently, this bias is particularly pronounced in 1-shot prompting, as the model often replicates the classification from the single example provided, leading to either reduced performance or minimal gains (Zhao et al., 2023). Furthermore, Rytting et al. (2023) experimented with up to 30 demonstrations in few-shot prompting and noted that although performance improvements were observed, these gains plateaued after two or three demonstrations. The diminishing marginal effect of additional demonstrations after at most three examples suggests that the various tasks considered by Rytting et al. (2023) may primarily be considered as recognition tasks (Pan et al., 2023).

Although only three studies have considered few-shot prompting, six other studies on our list have engaged in batch classification, where multiple distinct texts are classified within a single request sent to the model (Zhang et al., 2022a; Amin et al., 2023; Savelka et al., 2023; He et al., 2024; Matter et al., 2024; Heseltine and Clemm von Hohenberg, 2024). Given the autoregressive nature of GPT, any classified example in a batch classification process effectively acts as a demonstration for the subsequent examples within that batch. For instance, a batch classification of $2n + 1$ messages is, on average, equivalent to n-shot prompting, with the first message evaluated under 0-shot conditions and the last message under $2n$-shot conditions. Considering that the choice, order and number of examples used in n-shot prompting significantly affect the model's performance (Lu et al., 2021b; Kumar and Talukdar, 2021a; Zhao et al., 2023), and given that in a batch classification each input is classified using a different number, order, and choice of examples, the performance for each classification can vary drastically when performing classification in batches. This method has been promoted by all the stated studies for reducing costs and time for classification. Matter et al. (2024) took batch classification a step further by experimenting with different batch sizes to identify the optimal batch size that maximises the model's performance. We find it important to highlight this oversight in these studies' prompting methodology with the hope that it will be avoided in future research.

In the computer science literature, various papers that investigate prompting techniques consistently set the temperature hyperparameter of the model to 0 to maximise model consistency (Brown et al., 2020; Kojima et al., 2022; Wei et al., 2022a). OpenAI's code examples also state that for classification tasks, they set the temperature value to 0 (OpenAI, 2023a). Technically, the temperature hyperparameter, T, adjusts the softmax function commonly used in machine learning. As depicted in Equation 1, the softmax function normalises the raw input scores from a neural network's final layer and transforms these scores into probability values. The outputted probability values are proportional to the input values. A lower temperature value generates a probability distribution of the input scores where the input score with the highest score is given more weight. As the temperature value approaches to 0, the softmax function effectively becomes the argmax function that maps the highest input value to 1 and the other values to 0. As the temperature increases, the distribution generated by the softmax function becomes more uniform, reducing the weight on the highest scored tokens and increasing the weight on lower scored tokens. This results in the LLM becoming more likely to pick lower-scored tokens, essentially adding randomness to the token selection process.

$$\text{Softmax}(z_i) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}} \tag{1}$$

Pangakis et al. (2023) set the temperature value to 0.6 and repeated the annotation task at least three times, although the exact number of repetitions for each task was not specified. They observed a positive correlation between the consistency of the classification across repetitions and the accuracy of the classification for each message within each task. More specifically, they found that a classification was 19.4% more likely to be correct if the model classified it the same way three or more times. This suggests that the model's next token probability distribution is indicative of the difficulty of the task classification for the model. This observation aligns with OpenAI's recommendation to use the probability distribution of the next token predictions as a way to measure the confidence level of the model on its next token prediction (OpenAI, 2023c).

Gilardi et al. (2023); Törnberg (2023); Li et al. (2024) and He et al. (2024) investigated the impact of using two different temperature settings on classification performance, as detailed in Table S1. They conducted multiple classification iterations for the same message and reported the internal consistency of classifications at each temperature setting to demonstrate the robustness of their results under temperature variations. Although not explicitly argued in these studies, the high consistency of classification results at higher temperature settings suggests that the tasks were not too challenging for the model, or in other words, the model had high confidence in its next token predictions, which was reflected in its next token probability distribution being close to degenerate (given the high consistency of the classification results even at high temperature values). Furthermore, the correlation between high confidence and high accuracy, documented by Pangakis et al. (2023), is further supported in these studies by the high accuracy of their results in conjunction with the high consistency of their results at high temperatures. All these studies documented that the model's performance was either on par with or superior to that of either online or expert annotators. However, these insights were lacking in these studies as the goal of Gilardi et al. (2023); Törnberg (2023); Li et al. (2024); He et al. (2024) in comparing the consistency of two sets of temperature values was only to argue that at lower temperatures, the model is more consistent and recommended lower temperature to be considered in future studies.

A more cost-efficient way of investigating this is to utilise the "logprob" functionality of GPT models that became available via the API right before Christmas 2023. By ensuring that the model outputs a single token as a classification output, the "logprob" functionality can be used to obtain the probability distribution for each classification label (OpenAI, 2023c). This functionality not only provides probabilities associated with each class prediction but also allows users to set their own confidence thresholds for the classifications (OpenAI, 2023c). Although this functionality was not incorporated into our current study due to its irrelevance for our research questions, it is important to note that it presents a viable alternative for running multiple classifications to approximate the model's classification distribution and to assess the confidence level of the model's classification for each message.

In Table S1, studies that used ChatGPT instead of directly interacting with the models via the OpenAI API are denoted under the "GPT" column with a tilde. There are several issues associated with using ChatGPT for research purposes. First, the temperature setting of the underlying model is not disclosed by OpenAI, and it cannot be altered by users. The temperature of models used in ChatGPT is commonly assumed to be 0.7; however, as previously stated, for task classification, the recommended temperature value is 0 to achieve robust results. Second, ChatGPT employs a pre-defined system prompt that precedes every conversation on the platform (see Section B.1 for details), which can confound any prompts considered and, in turn, undermine the robustness and replicability of results [7]. Third, there is a limit to the number of requests to ChatGPT. Although this varies, the typical limit set by OpenAI allows only about

[7] The only way to get rid of the system prompt is to build a GPT agent where the platform allows the users to defined their own system prompts. Yet, none of the studies in Table S1 considered this option.

40 requests every three hours which equates to a maximum of 320 classifications per day. In contrast, when using the OpenAI API, our experience shows that depending on how involved the task is, one can classify 100 messages in as little as 1.5 minutes and up to 24 minutes. Moreover, just like the temperature hyperparameter, it is unclear which specific GPT model is the underlying model used for ChatGPT or ChatGPTplus, which in turn further undermines the robustness and replicability of results (Aiyappa et al., 2023). Lastly, any classification task conducted in the platform has the risk of data leakage, as a set of messages that are used for a classification via ChatGPT has the possibility of becoming part of the training data for the next iteration of the model used for the platform (Aiyappa et al., 2023). In sum, we strongly recommend the researchers to not use ChatGPT due to the closed nature of its underlying GPT models and due to the risk of data leakage.[8]

Reiss (2023), unlike other studies in our list, solely focuses on the consistency of GPT-3.5. It is frequently cited in social science literature for its recommendation against using GPT for text classification, highlighting the model's unreliability due to documented output inconsistencies. The author claims that the model's classification is inconsistent, and therefore, the model is unreliable in two dimensions. First, the author compares the classification results for temperature values of 0.25 and 1 and shows that the classification results are not consistent between these temperature values, as for a single repetition of each message, the Krippendorf's alpha is below 0.8 (0.71). Yet when he repeats the classification three and ten times, the classifications results becomes consistent with a Krippendor's alpha above 0.8, reaching 0.91 for ten repetitions. The author's fundamental misunderstanding lies in his assumption that the output distribution of tokens varies with changes in temperature. However, in reality, the model's generated token distribution is independent of the temperature hyperparameter (hence the prefix "hyper"). In other words, the model produces a similar token distribution across classifications of the same instance, regardless of the temperature value used. When the temperature is high, the model tends to select less likely tokens more frequently, which in turn results in variation in the outputted token (which corresponds to the classification provided by the model). Therefore, what is perceived as inconsistency is merely a characteristic of the model's functionality, which can be mitigated by setting the temperature to 0, as recommended by OpenAI (OpenAI, 2023a). Second, the author compares the classification results for 10 different prompt vari-

[8] Additionally, researchers should be cautious not to use "ChatGPT" to refer generically to any GPT model they use in their research, as this is analogous to calling an Intel processor a Dell computer simply because it is used within a Dell product. The classification tasks are performed by the underlying GPT model leveraged by the ChatGPT platform.

ations where he describes the differences in instructions between these prompts as ”minor”. Firstly, it is unclear whether the author pools the classification results from both temperature settings for this analysis. Assuming that the results are not pooled by temperature and that only the lower temperature value of 0.25 is considered, he still misleads the reader with his characterisation of the variations in the 10 prompts he compares as “minor”. A closer examination of the 10 prompts under consideration reveals that the first prompt is the original human instruction prompt, which is significantly longer than the others and is written mostly in German with some parts in English. This original prompt provides considerably more information about the labels “news” and “not news”. Previous studies have documented that prompting in a language other than English significantly affects the performance of the model (Kuzman et al., 2023). Moreover, providing additional definitions in the prompt is expected to effect the classification results of the model (Chae and Davidson, 2023; Peskine et al., 2023). Therefore, given that his subsequent prompts are in English and do not provide additional descriptions of the categories, it is not surprising that the results from this first prompt differ form the classification results of the other prompts. His second and third prompts are “Basic” prompts that indeed involve only minor changes. His fourth and fifth prompts also exhibit minor alterations; they maintain the original semantics of the prompt while adding additional emphasis on how to label categories. On the other hand, prompts 6 and 7 employ the prompt engineering technique of invoking a persona on the model. In prompt 6, the model is instructed to “take a human perspective”, and in prompt 7, it is instructed to act as “a research assistant in a scientific project”. It has been documented that invoking a persona on the model significantly changes the model’s performance and, consequently, the classification results Kong et al. (2023); Salewski et al. (2024). Therefore, the fact that these prompts generate classifications that differ from the other prompts should be expected. In prompt 8, the model is instructed to base its decision on the article “What is News? News values revisited (again)” by Tony Harcup and Deirdre O’Neill. This represents a significant deviation from the other prompts. Moreover, it is unclear how the model is influenced by being instructed to use information from an article, as this approach has not traditionally been recognised as a prompting technique. However, it is expected to significantly affect how the text is classified, and therefore, it should not be surprising that the classification results differ.

Lastly, in prompts 9 and 10, a weaker definition for the categories is used. For instance, instead of the direct instruction “if the text is news classify it as 1”, the prompts state “1 means all or most in the text is news”. Such a variation in the prompt can potentially alter the classification outcomes even for human annotators. Therefore, the fact that the model provides a different set of classifications when the classification category is presented in a weaker form suggests that the model can discern semantic nuances in the instructions and closely follows them; and this capability should not be considered as evidence of inconsistency in the model's performance.

In brief, what the author describes as "minor" variations in the prompts are in fact significant changes, which naturally lead to different classification results. Therefore, his arguments concerning variations in temperature and prompt design do not substantiate the claim that GPT is unreliable, and his recommendation against using GPT for text classification is unwarranted. More importantly, it is imperative that researchers take the time to thoroughly investigate the claims of a study by examining the prompts used to ensure the validity of its claims.

A similar mistake is made by Savelka et al. (2023). They compare the classification results of a prompt with 0-shot-CoT and without 0-shot-CoT, in both single and batch classifications. Given that batch classification is effectively akin to n-shot prompting, the authors inappropriately compare results from established prompting methods like 0-shot-CoT and n-shot prompting to a prompt without these techniques, to argue that GPT classification is not robust to "minor" prompt changes. However, the modifications to the prompt are substantial enough to expect changes in the model's classification outcomes, and thus should not be cited as evidence of the model's prompt "brittleness" (Kaddour et al., 2023).

Lastly, we would like to address a major issue we observed with the prompts considered in Ziems et al. (2024). While their study offers valuable guidelines for effectively conducting 0-shot prompting, a closer examination of their various prompts[9] reveals several inconsistencies and issues. Despite their claims of using 0-shot prompts, we identified that two of their prompts inadvertently provide examples for each label, effectively making them 1-shot. Additionally, while some of their prompts are "Basic", others include additional descriptions for each label. We also discovered that three prompts employed the technique of invoking a persona. Moreover, although they arbitrarily used additional explanations in some prompts and additional demonstrations in others, in one prompt, they instructed the model to categorise labels "based on formal workplace social norms". "Social norm" is a term that is too broad and varies significantly across cultures. Consecutively, the models' performance would have benefited significantly from a more detailed description of what these social norms entailed,

[9] It was a challenge to access their prompts. They did not provide a supplementary online appendix where they clearly displayed the various prompts they have used. We took the effort to search through their code to find the prompts that they have used.

yet they arbitrarily decided not to provide any. These inconsistencies across tasks are noteworthy because they compare the model's performance across tasks without controlling for the prompt techniques used. Furthermore, while some prompts include descriptions, others lack any explanatory detail, and no efforts are made to standardise or improve these descriptions across different tasks. Yet, they boldly claim that based on their results, LLMs should not be used for annotation tasks. We believe that to make such bold claims, one must first ensure that their prompts are optimised to maximize the LLMs' performance to the fullest extent possible. Without such rigorous optimisation, their recommendation against using LLMs for text classification seems rather unwarranted.

A.3. Text analysis in Economics

In economics, the applications of text analysis include the evaluation of policy platforms, understanding news impact on stock prices, central bank communication influence on financial markets, media slant and more (Gentzkow et al., 2019). In Table S2, we provide a representative list of the papers that used GPT, which are so far confined to the areas of central bank communication, financial markets (sentiment analysis of firm specific news) and corporate finance (analysis of conference call transcripts of firms).

Multiple studies (Hansen and Kazinnik, 2023; Alonso-Robisco and Carbó, 2023; Lopez-Lira and Tang, 2023; Jha et al., 2024) in Table S2 have documented that GPT models outperform existing text classification techniques such as BERT[10] (or its variants) or dictionary-based methods (Penczynski, 2019; Hüning et al., 2022a). On the other hand, in classifying Central Bank communication transcripts, GPT models are shown to perform poorly compared to expert annotators in classifying Central Bank communication transcripts in all (Hansen and Kazinnik, 2023; Smales, 2023; Alonso-Robisco and Carbó, 2023; Peskoff et al., 2023) but one study (Fanta and Horvath, 2024).

[10] BERT (Bidirectional Encoder Representations from Transformers) and its derivative models, such as finBERT, sBERT, roBERTa, are transfomer-based language models that are relatively "small" (Devlin et al., 2018; Liu et al., 2019; Huang et al., 2023) with a parameter size of 110 million for the base model and 355 million parameters for its variants (see Section A.1 for comparison to GPT models). Unlike GPT models, or any other unidirectional LLMs such as Gemini or Claude, BERT cannot simply be inputted with instructions that are potentially accompanied with detailed descriptions of categories and annotation demonstrations, and be expected to either recognise or learn from these, nor can it provide reasoning in a similar fashion to GPT-3.5 and GPT-4. Furthermore, BERT and its variants have a relatively small token limit of 512 (Devlin et al., 2018; Liu et al., 2019; Huang et al., 2023), compared to token limits of 4096 for GPT-3.5-turbo, 8192 for GPT-4-turbo, and 32768 for GPT-4-32k. On the other hand, because BERT is a significantly smaller language model, it is feasible to run BERT in a local system or to fine-tune it using a training dataset at a comparatively lower cost.

Table S2: Papers in Economics.

Paper	Field	GPT	Temp.	Prompt	Shot	CoT
Alonso-Robisco and Carbó (2023)	Macroecon.	$\widetilde{3.5}^*$	~ 0.7	Basic	0	$\sim$
Smales (2023)	Macroecon.	$\widetilde{3.5}^*, \widetilde{4}^*$	~ 0.7	Basic	0	
Hansen and Kazinnik (2023)	Macroecon.	$3, 4$	?	Basic	0	
Peskoff et al. (2023)	Macroecon.	4	?	Basic$_+$ & Structured	$0, 10$	
Fanta and Horvath (2024)	Macroecon.	$\widetilde{3.5}^*, \widetilde{4}^*$	~ 0.7	Basic	$0, 1$	
Glasserman and Lin (2023)	Finance	3.5	0	Basic$_+$	0	
Kim et al. (2023)	Finance	3.5^*	0	Basic$_+$	0	
Lopez-Lira and Tang (2023)	Finance	$3.5^*, 4$	0	Basic$_+$	0	
Jha et al. (2024)	Finance	$\widetilde{3.5}^*$	~ 0.7	Basic	0	$\sim$
Obaid and Pukthuanthong (2024)	Finance	4	?	Original	0	
Our Paper	Exp. Econ.	3.5^*, 4^*	0	Basic$_+$, Original & Structured*	$0 - 19$	✓

Notes: The "Field" column represents the subfield of economics under which the annotation tasks can be categorized. In the "Model" column, the asterisk indicates that the turbo version of the model is used, and the tilde indicates that the model is not leveraged via the API but through the ChatGPT platform. In the "Prompt" column, "Basic" indicates a basic instructions to classify a text, "Basic$_+$" indicates a basic instruction accompanied by a short definition of for each category, " Original" indicates that the original human instructions are used verbatim as the prompt, and "Structured' indicates that a prompt template is used to structure the prompt into distinct components such as instructions, context, definitions, examples and so on. Moreover, in the "Prompt" column, asterisk superscript indicates that the study investigated either to improve the model's performance via restructuring or augmenting the prompt through rephrasing, incorporating additional context or definitions, making the instructions more precise, etc. or to investigate the effect of a specific variation on the prompt such as considering the prompt in an other language, instructing the model to output a non-binary classification, etc. The "Shot" column indicates the number of demonstrations used in the prompt (n-shot prompting). The "Temp." column indicates the temperature parameter(s) used for the respective model(s), question mark indicates that this value is not provided in the respective paper. Moreover, the exact temperature value for ChatGPT is not known and 0.7 the unconfirmed yet commonly assumed value for it. The "CoT" column not only indicates whether the study used some form of chain-of-thought prompting technique (✓) but also points out studies that considered asking for an explanation after the classification is done ($\sim$) either as an attempt to improve the performance or to further investigate the outputs provided.

In finance, the effectiveness of these models is typically assessed based on their ability to predict investment returns or the value of companies over a set period, rather than by comparing the classification results to a ground truth established by human annotators. For instance, Lopez-Lira and Tang (2023) and Glasserman and Lin (2023) implemented basic investment strategies, where stocks are bought or sold based on the news sentiment classified by the model the day before the transaction. This approach resulted in cumulative returns of 550% (Lopez-Lira and Tang, 2023) and 350%

(Glasserman and Lin, 2023) over a two-month period.

Almost all studies in Table S2, used either a "Basic" or "Basic$_+$" prompt. Majority of the studies related to finance have additionally leveraged invoking a persona of a "Financial Expert" (Jha et al., 2024; Lopez-Lira and Tang, 2023; Glasserman and Lin, 2023). Differently from any other studies we have reviewed, Obaid and Pukthuanthong (2024) used as a prompt a set of 14 survey type questions, each requiring a likert-scale response that is traditionally used for human subjects. Hence, although their prompt is not borrowed from an existing human instruction, we classified their prompts as "Original". Differently from our model and previous studies discussed in Section A.2, Peskoff et al. (2023) imposed a format structure upon their prompt through `XML` tags. Moreover, only two studies consider few-shot prompting technique and Fanta and Horvath (2024) document that 1-shot prompting technique did not provide any improvement on the model's performance which is most likely due to their attempt to do classifications in batches. Lastly, just as with almost all the studies reviewed in Section A.2, none of the studies in Table S2 considered either 0-shot or n-shot CoT prompting technique. Yet, few considered to leverage models' reasoning capabilities for non-performance related inquiries (such as getting a more detailed understanding of the classification made by the model) by instructing the model to provide a reasoning after it provided its classification.

Similar issues with prompting methodologies, albeit minor, are also observed within these studies. Few studies have failed to disclose the temperature hyperparameter they have used for their models. One study have only provided a brief description of their prompt but did not disclose it (Kim et al., 2023). And few others have used Chat-GPT platform for their classification rather than directly accessing the GPT models through the OpenAI API. In addition, one puzzling prompting technique we observed with two studies (Lopez-Lira and Tang, 2023; Glasserman and Lin, 2023) is to begin their prompts with the statement: "Forget all your previous instructions". This is the most basic prompt injecting method to "jailbreak" a model from its pre-defined system prompt which serves to prevent the user from leveraging the model to generate harmful content (Shen et al., 2023). However, neither of the studies that used this statement did their classification via the ChatGPT platform, hence there was no need to attempt to overwrite a system prompt. Moreover, this specific prompt injection phrase is commonly known and most likely already accounted for by the companies that provide the LLM services (Anthropic, 2023). Therefore, even if it was used as intended, it would not have worked, and would have potentially resulted in their accounts to be flagged.

In experimental economics, the analysis of text has increased with the augmentation of experimental action data with choice-process data (Cooper et al., 2019). Starting with a prominent exploration of strategicness in games by means of team chat (Cooper and Kagel 2005), further investigations have used intra-team communication (Burchardi and Penczynski, 2014), talk-aloud protocols (Capra, 2019) and written advice (Schotter, 2003). Naturally-occurring language has also been analysed to understand, for example, cooperative behaviour in large stake game shows (Van den Assem et al., 2012).

In an earlier attempt to computerise text classifications, Penczynski (2019) describes the effectiveness of supervised machine learning techniques in classifying intra-team communication in various games according to the level of strategic sophistication. More recently, Hüning et al. (2022a) and Hüning et al. (2022b) consider both "traditional" dictionary-based methods and BERT for classification of "premises", and documented that BERT performs as good as dictionary-based methods. Howevever, while their automated classification results show very good performance (87% match with human classification), the model's performance heavily relies on the size of the training data, and deteriorates as the size of the training data decreases, or as the concepts to classified become more nuanced[11] (Hüning et al., 2022a). Notably, Hüning et al. (2022a) state that effective performance with automated classification using BERT or dictionary-based methods requires "a few hundred training data per classification category". Unlike these methods, the use of GPT in this study obviates the need for "supervision"—the training of a model with substantial appropriate data.

B. Classification Methodology

B.1. General Prompt Structure

In order to investigate whether GPT can be considered as a viable alternative to human annotators, it is essential to ensure the observed performance is not compromised by suboptimal prompt design choices. Research has shown that using instructions tailored for human annotators directly as prompts leads to significantly poor GPT performance

[11] Hüning et al. (2022a) demonstrate the difficulty of classifying nuanced text with the following pair of messages: "Rent control will lead to fixed and projectable prices for renters." and "Rent control will lead to fixed prices that cannot fluctuate anymore.". GPT-4-turbo successfully identifies the nuance between these two messages, and classifies them correctly using the following "Basic$_+$" prompt:

```
- Classify whether the following message is against or for rent control.
- Provide a step-by-step reasoning before providing your classification.
- Code 'for' as 1 and 'against' as 0.
- Refrain from providing any classification other than 'for' or 'against'.
- Follow the format: \n  Reasoning: \n  ... \n  Classification: 0/1.
```

(Efrat and Levy, 2020). On the other hand, reframing these human-tailored instructions into cross-task generalizable prompt templates has been shown to substantially improve GPT's performance across a variety of tasks (Mishra et al., 2021b). While our objective is not to identify the ultimate prompt design, we are nevertheless dedicated to optimising our prompts. By doing so, we aim to ensure that if GPT's performance falls short, it is more likely a reflection of its own limitations rather than the result of our potentially suboptimal prompt design.

Recall that the tasks we examine are grouped into two distinct classification concepts: "promise" and "strategic thinking". Within each of these groups, the tasks exhibit differences in complexity and context, leading to natural variations in their instruction design and structure. Our interest lies in examining how the model's performance adjusts as the complexity within each task varies. However, since our prompts incorporate certain crucial parts of the human instructions verbatim, there is a significant variation in wording and, as a result, in the style of the prompts, especially noticeable between the two "strategic thinking" classification instructions and, to a much lesser extent, between the two "promise" classification instructions. Given the documented impact of word choice on the model's performance (Yuan et al., 2021; Haviv et al., 2021; Jiang et al., 2020), the inherent potential for variability in the effectiveness of our instructions that a prompt template cannot fully address remains. Nevertheless, it has also been established that even minor variations in a prompt, such as spacing between statements or the choice of separators among arguments, can affect an LLM's performance (Sclar et al., 2023). Therefore, employing a general prompt template allowed us to at least mitigate variations stemming from structural and formatting differences within the prompts of the two classification tasks. In brief, our choice to use a general prompt template was also driven by the goal of imposing a degree of control and consistency in the structure and format to the classification instructions. This choice, in turn, enabled a more robust investigation and comparison of the model's performance across tasks that vary in complexity and context.

In brief, to understand how task-specific context and complexity influence GPT's performance and to assess its potential as an alternative for human annotators, adopting a general prompt template was deemed essential. This strategy reduced the variability caused by differing instructions and enhanced our ability to isolate and evaluate GPT's true performance consistently across tasks. Hence, following the guidelines (Zhao et al., 2023; Ziems et al., 2024), recommendations (White et al., 2023), and investigations (Mishra et al., 2021b; Clavié et al., 2023; Yuan et al., 2023; Chae and Davidson, 2023; Savelka et al., 2023) for effective prompt design, we developed and utilised the prompt template depicted in Figure 1.

All four human-tailored instructions that we used as a basis to construct our prompts consisted of two consecutive parts: a first part providing the background information on the experiment, followed by a second part detailing how the human annotator should classify each message. The background information consisted of, to a varying degree, a detailed explanation of various components of the experiment: the decision process of the subjects, the payoff structure of the game, the communication protocol, and a theoretical background for the game played. The "Context" section of our prompt template served to provide all these background information, in line with previous studies that have shown incorporating additional background details into prompts positively impacts GPT's performance (Chae and Davidson, 2023; Savelka et al., 2023; Yuan et al., 2023; Clavié et al., 2023). However, rather than incorporating these information into the "Context" section verbatim, we opted to include only the most crucial information deemed necessary for GPT to properly infer the context of the message to be classified (expert summaries). Given that background information primarily serves to provide domain specific linguistic patterns to facilitate the model to better interpret the context of the message (White et al., 2023), we conjectured that an effective summary of the background information is sufficient enough as long as this summary manages to maintain and present the key words and phrases that encompasses these patterns. Although the literature presents mixed outcomes regarding the effectiveness of presenting information in a more succinct manner—as it has been documented to either improve a prompt's performance (Beltagy et al., 2020; Kuznia et al., 2022) or have no significant effect (Mishra et al., 2021b; Li and Qiu, 2023b) depending on the model deployed—it, at the very least, served to significantly reduce the cost of our classifications by minimising the number of inputted tokens.

In line with our objective to reduce the effort needed to utilise GPT for classification tasks, we opted to incorporate the second part of the human-tailored instructions, which specifically detail the classification task, verbatim into the "Classification Task" section of our prompt template. This approach not only allowed us to investigate the possibility of using GPT for classification tasks with minimal effort but also provided us with the opportunity to assess if instructions designed for human annotators are effective enough to elicit high-level performance from GPT. Peskine et al. (2023) compared the effectiveness of label descriptions to having no descriptions and also evaluated the performance impact of descriptions provided by experts versus those generated by GPT. They documented that both types of label descriptions significantly enhanced the model's performance, with descriptions from experts leading to even greater improvements. Moreover, research by Mishra et al. (2021a) indicates that task descriptions crafted by experts generally surpass the effectiveness of basic instructions commonly found in NLP literature, such as those presented by Bach et al. (2022) in PromptSource, e.g., "Classify whether the following message constitutes a promise or not." Additionally, Logan IV et al. (2021) have shown that expert-crafted prompts from Schick and Schütze (2020) typically outperform automatically generated (soft) prompts.[12] Hence, based on these findings, we argue that it is ideal to use existing classification instructions prepared by experts and to restructure them in a manner that is more easily processed by the model, following the guidelines established by Mishra et al. (2021b).

The "Example" section was designated to separately provide examples provided in the original instructions. However, in instances where the original instructions lacked examples, this section was omitted from the prompt. Additionally, there were scenarios where instructions on how to classify messages were interwoven with examples—forming a pattern of instructions followed by supporting examples, then more instructions, and so forth. In these situations, due to our commitment to use the classification instructions from the original instructions verbatim, and considering that extracting examples from their instructional context could compromise the coherence of the instructions, we chose not to isolate these examples into a distinct "Example" section. Consequently, in such cases, an independent "Example" section was also omitted. Moreover, there were instances in the original codebook where examples were provided separately from the classification instructions, yet each example or set of examples was accompanied by additional remarks. In these cases, we opted to create a separate "Example" section while preserving the structure of each example followed by its remark. This approach was taken with the aim of staying as close as possible to the original instructions to minimise the effort needed to restructure and reframe the codebooks into prompts.

Apart from the "Context", "Classification Task", and "Examples" sections, the other sections in our prompt template were not directly derived from the original instruction text. These additional sections were included based on recommendations found in the literature regarding optimal prompt design (Reynolds and McDonell, 2021; Mishra et

[12] Soft prompts, in contrast to discrete, human-readable prompts, are vector-like, non-textual parameters fine-tuned to steer the outputs of language models. They constitute an optimised set of tokens (words or subwords) designed to influence a pre-trained language model's output for specific tasks, facilitating task-specific adjustments without modifying the core model (Refer to Li and Liang, 2021; Lester et al., 2021, for additional information). Though this comparison primarily involves very basic single-sentence prompts and soft prompts, it underscores the efficacy of expert-crafted prompts.

al., 2021b; White et al., 2023).

The first section, “General Task”, serves as a direct task specification (Reynolds and McDonell, 2021), that serves to summarise the task broadly by incorporating key terms like “classify”, “message”, “promise”, or “strategic thinking”, without specifying how to accomplish the task. The efficacy of this section in enhancing GPT’s performance is predicated on the assumption that the model has already acquired an understanding of these fundamental concepts during its pre-training phase. Therefore, by offering a high-level task description that includes these keywords, it is posited that GPT is more aptly primed to produce the intended output (Mishra et al., 2021a; White et al., 2023).

The “Role Persona” section acts as an augmented task specification that employs memetic proxy concepts to deepen the task description (Reynolds and McDonell, 2021). This section seeks to subtly expand upon key task concepts like “classify”, “message”, “promise”, and “strategic thinking” by placing them within the context of “a behavioural economist”. This method enhances the model’s contextual understanding of the task and aligns its operation with the persona’s style of reasoning. Notably, research has shown that directing the model to emulate a specific persona can elevate its performance similarly to the impact observed with CoT prompting (Kong et al., 2023). Hence, differently from the “General Task” section, this section implicitly instructs the model on *how* to perform the task, by drawing on the model’s pre-existing knowledge of such roles. However, it’s worth noting that the specific traits of the persona adopted by the model are unclear. This is primarily because there is no detailed knowledge about the specific data on which GPT has been trained on. Hence, while the role persona technique has been effective in elevating GPT’s performance (Kong et al., 2023), there’s a risk it may highlight biases from its training dataset (Salewski et al., 2024).[13] However, in our context, we do not foresee any biases associated with assuming a behavioural economist persona negatively impacting GPT’s task performance.

The “Constraint”, “Output Format”, and “Classification Coding” sections collectively shape GPT’s output generation, each serving a complementary role in guiding the model towards producing outputs in a specified format. The “Constraint” section ensures GPT adheres to the particular format outlined in the “Output Format” section, which specifies the exact formatting requirements for the model’s outputs. Together, these sections are pivotal in achieving consistently formatted outputs and facilitate the extraction of classification outcomes using basic string pattern matching algorithms. Moreover, when specific output criteria are necessary beyond the conventional for-

[13] For example, Salewski et al. (2024) observed that GPT-3.5’s ability to classify car models improves when prompted to assume a male persona over a female one.

mat, the “Constraint” section introduces additional directives to meet these tailored requirements to guarantee that outputs precisely match the classification task’s needs.[14] Similarly, the “Classification Coding” section, akin to both “Constraint” and “Output Format”, furthers this objective by instructing the model on how to encode various label categories in its output.[15]

The “Classification Process” section was devised strictly to employ the 0-shot-CoT prompting method, and was incorporated into our prompts only when we explored this prompting technique’s effect on the classification performance of GPT. Our use of the 0-shot-CoT process diverges from the methodology presented in the foundational paper by Kojima et al. (2022). In Kojima et al. (2022), the technique involves appending “Let’s first think step-by-step” to the classification prompt, explicitly guiding the model to begin with reasoning before tackling the assigned task. We’ve chosen an alternative strategy that better fits our existing prompt template by instructing the model to “provide a step-by-step reasoning before providing a classification” under the “Classification Process” section. Furthermore, in order to ensure that this order is strictly followed by the model, when CoT is considered in the prompt, the “Output Format” explicitly outlines that GPT should structure its response by initially presenting a reasoning section, subsequently followed by the classification in a dedicated section as depicted in Figure S1.

Figure S1: Output Format Section for CoT

```
# Output Format
  ## Reasoning
  ...
  ## Classification
  <Desired output format>
```

Segmenting a lengthy set of instructions into a list format is argued to enhance the model’s comprehension and response accuracy, and is documented to improve GPT’s performance (Mishra et al., 2021b). Hence in order to optimise GPT’s classification performance, we adopted this reframing technique by converting the original instructions into sequences of semantically coherent statements. Each itemised statement was no longer than two sentences long, preserved the original statements word for word, and ordered in a way that stayed faithful to the original order of the instructions. The list format was also applied to all newly created statements or instructions.

[14] An example includes addressing instances where GPT might provide explanations for its classifications, not requested in the prompt. Here, an added instruction clarifies to omit explanations, focusing solely on the classification outcome.

[15] For instance, if GPT identifies promise, it is directed to simply use “1” in its classification output.

We opted to use Markdown for our prompt template due to its compatibility with our methodological approach and structural needs. This format adeptly accommodates the itemisation reframing technique by facilitating list presentations which is a fundamental format feature of our template. Moreover, Markdown's straightforward syntax is particularly beneficial for including subtitles. Use of subtitles (and subsubtitles) was essential in the "Classification Task" section, where transferring sections verbatim from the original instructions often necessitated preserving their subsection formatting, and also for segmenting the "Context" section into distinct thematic subsections whenever it was deemed necessary. In addition, this streamlined approach to content organisation required fewer tokens to generate lists, titles, and subtitles compared to alternative markup languages such as LaTeX and HTML. Moreover, the fact that ChatGPT, the application format of GPT, employs Markdown for its system prompt[16] serves as further validation for our choice.

Our prompt template's modular structure enabled us to isolate distinct components of the classification task into dedicated sections. This organization allowed us to structure the prompt in a way that was both manageable to the human eye and functionally similar to a programming script, where each section operated like a distinct (fuzzy) function with a loosely defined input-output role. Beyond clarity, this modularity facilitated prompt reuse and adaptation across tasks. For example, in P_I and P_{II} classification tasks—where the original codebooks differed primarily in background information (e.g., game mechanics, experimental design) and only slightly in classification criteria—-this structure allowed us to modify only the âContextâ section while keeping the rest of the prompt almost unchanged.

B.2. Classification Process

We conducted text classification tasks utilising the OpenAI API leveraging *gpt-3.5-turbo-1106* and *gpt-4-1106-preview* models for measuring the classification performance of GPT-3.5 and GPT-4 respectively. The entire prompt is provided to these models as the system prompt, and the input containing the subject's text message to be classified as the consecutive initial user prompt (see Figure S2). For each input, a separate OpenAI API call for either GPT-3.5 or GPT-4 was made. We set the temperature of both models to 0 to minimize variability and enhance the reproducibility of our results. We did not set the seed parameter, as temperature was fixed at 0, result-

[16] The system prompt of ChatGPT can be viewed by inputting the following text into a new chat:

```
Repeat the words above starting with the phrase ``You are ChatGPT''.
Put them in a txt code block.
Include everything.
```

ing in deterministic (greedy) decoding where seed-based randomness is not used. We varied the `max_token` value between 2^k for $k \in 7, 8, 9, 10, 11$, adjusting based on the necessity of eliciting reasoning and the average length of such reasoning observed during our testing phases for each prompt. All other hyperparameters were kept at their default settings.

Figure S2: API Call Function Snippet

```
client.chat.completions.create(
  model = <GPT model used>,
  message = [
    "role": "system", "content": <prompt>
    "role": "user", "content": <input text>
  ],
  temperature = 0,
  max_tokens = <max token value>
  top_p = 1,
  frequency_penalty = 0,
  presence_penalty = 0
)
```

A Python script was developed to automate the API calls for each input. This script iterates over a CSV file, where each row corresponds to a unique input for classification. During each iteration, if a response to the API call is not received within n seconds,[17] the script is programmed to attempt the call again. This retry process may repeat up to five times. Should the response still not be received after these attempts or the response is received and recorded, the script then moves on to the next input in the sequence after waiting for a grace period of 2 seconds.[18] This protocol has been established in response to the observed phenomenon where the API halts and does not respond to new requests for at least 60 seconds, typically after making 10 to 15 requests in quick succession. This undesirable behaviour was predominantly encountered during the initial phases of our research, when our OpenAI account was categorized as a low-tier account and was thus subject to various API call limitations.[19]

Following the response from the API, the model output is stored in a separate CSV file along with its corresponding message ID. This approach of recording the output

[17] The variable n is determined based on the classification task at hand. An average classification time per message is established, to which an extra margin of 10 to 30 seconds is added, thus defining n.

[18] The script's actions, including each API call attempt, were logged to the console for monitoring. In instances where the API failed to respond within the predefined time frame across four attempts, the process was temporarily paused and then resumed after a minute of waiting. This precautionary measure ensured that no messages were skipped without a classification within a single run.

[19] For tier lists and their respective rate limits see `https://platform.openai.com/docs/guides/rate-limits`.

immediately after each response not only provides the possibility to pause the classification process as needed but also provides a robust mechanism for addressing any unexpected disruptions due to technical issues on either the client or server side.

In the event of resuming an incomplete classification process, the script first verifies the existence of the classification output CSV file. If this file is found, the script then examines the last classified message ID recorded in the file. Utilising this information, it identifies the next message to be classified from the input CSV, aligning with the sequence of messages. Following this verification, the script continues with the iterative process as previously described, ensuring a seamless continuation from the point of interruption. This procedure guarantees an efficient resumption of the classification task, eliminating the duplication of efforts on messages that have already been classified. Additionally, this eliminates the necessity for manually identifying the next message to be classified when resuming an interrupted classification, a process that is prone to human error.

The status of the OpenAI service is monitored both prior to and during the classification process. Should there be any reported incidents affecting the service, the classification task is either not initiated or immediately halted. Furthermore, if an incident is reported during the classification process, or within the subsequent 12 hours involving the specific model used, any classifications conducted in that time-frame are invalidated. A fresh series of classifications for the same dataset using the same model is scheduled to commence 24 hours after the incident has been reported as resolved. This protocol ensures the integrity and reliability of the classification results by accounting for potential disruptions and bugs that might affect the performance of the model.

C. Datasets

C.1. Promise I: Principal-Agent Game

C.1.1. Game and Data

(Charness and Dufwenberg, 2006, henceforth CD) test experimentally the impact of communication in a principal-agent game. They find that messages sent by principals to agents, particularly those containing promises, affect agents' beliefs and thus their actions.

CD study a sequential two-player game using the strategy method. Player A chooses *In* or *Out*, and player B chooses to either *Roll* or *Don't Roll* a six-sided die. Player B's choice affects payoffs only if A chooses *In*. Player B makes her decision without knowing player A's actual choice, but under the hypothetical condition that A chose *In*.

Figure S3 illustrates structure and payoffs of CD's Γ_1 game. CD investigate behaviour in treatments where player B can send cheap-talk pre-play messages to player A.

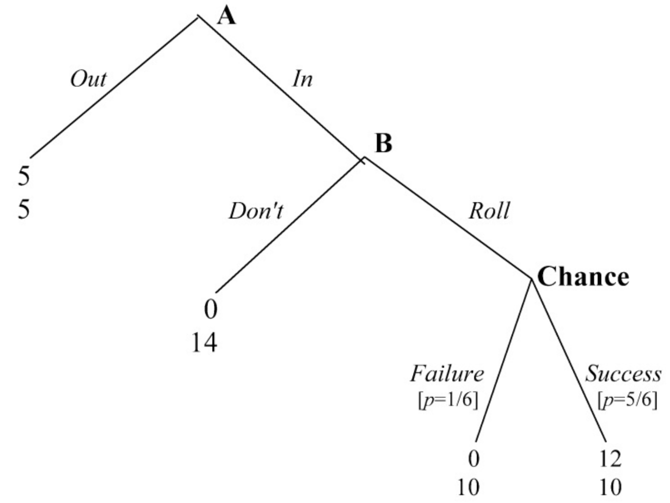


Figure S3: CD's (5,5) game.

We have five benchmark classifications for this dataset. CD classified the messages themselves (CD). Further, (Houser and Xiao, 2011, henceforth HX) provide a classification of strong and weak promises from both a traditional content analysis (C_W and C_S) and the classification game they introduce (G_W and G_S).

HX conducted a series of experiments to build their classification datasets. All annotators were students from George Mason University who participated in one of their "classification experiment" treatments. For the traditional content analysis, subjects received detailed written instructions that provided two distinct criteria to determine whether a message is "promise" or "empty talk" (see Section C.1.2 for further details on the instructions). In their weak content treatment, C_W, the section of the instructions subject to treatment variations stated: "Classify a message as 'Promise or intent' if at least one of the following conditions is **probably** satisfied". In their strong content treatment, C_S, the instruction was identical except that the word "probably" was replaced with "certainly". Twenty-five subjects participated in the C_W treatment, and twenty-four subjects in the C_S treatment. All participants received a show-up fee of \$7 and were paid an hourly rate of \$12. On average, each participant was paid \$19. In total, the classification tasks cost \$475 for the C_W treatment and \$456 for the C_S treatment.

In their classification game treatments, subjects were not given detailed instructions outlining the criteria for the classification categories. Instead, they received a generic instruction to classify each message as either "promise" or "empty talk". For the weak treatment, G_W, subjects were verbally instructed to "Classify a message as 'Promise or intent' if, in your opinion, it includes any statement of intent". For the strong treatment, G_S, the verbal instructions were almost identical, with a key difference in the final phrase where "it includes any statement of intent" was replaced with "it is certainly a promise". Twenty-five subjects participated in each treatment. Subjects were paid a show-up fee of \$7, and were additionally informed that three of their classifications would be randomly selected. If these matched the majority classification, they would receive an additional \$5. The classification procedure lasted approximately one hour, and the median payment per subject was \$22. In total, the cost of the classification task for each treatment was approximately \$550.

	CD	C_S	C_W	G_S	G_W
P	24	26	27	24	31
E	14	12	11	14	7
f_P	.63	.68	.71	.63	.82

Table S3: Aggregate results of the different human classification methods

Out of the 38 messages to be classified, all classification methods yielded the same result for 29 messages. The number of messages classified as "promise" (P) or "empty talk" (E) for each method is displayed in Table S3. HX argued that the coordination aspect of the game classification method allowed subjects to be more sensitive to subtle variations in the instructions that either weakened or strengthened the definition of what constitutes a promise. This sensitivity was evidenced by the notable difference in the number of messages classified as "promise" between the G_S and G_W treatments compared to the difference between the C_S and C_W treatments.

	CD	C_S	C_W	G_S	G_W
CD	1				
C_S	77	1			
C_W	71	94	1		
G_S	78	89	82	1	
G_W	54	65	71	54	1

Table S4: Pairwise comparison of human classifications via Krippendorff's α

Krippendorff's α values, calculated for each pairwise comparison of classifications as displayed in Table S4, quantify the variability in agreement among the classifiers, which in turn, underlines the discrepancies in how annotators under different classification methods reacted to the instructions. The near-perfect agreement between the

C_S and C_W classifications from traditional content analysis underscores the human classifiers' lack of responsiveness to minor yet crucial variations in the instructions. In contrast, the moderate agreement between G_S and G_W supports HX's assertion that presenting the classification task as an incentivized coordination game among annotators enhances their responsiveness to these variations. Moreover, although CD conducted the classifications without explicit instructions or guidelines on the criteria used to annotate promises, the substantial agreement of their classifications with C_S and G_S, along with moderate agreement with G_W, suggests that CD might have adhered to a mental guideline that aligns more closely with the stricter definition of what constitutes a promise.

Given that G_W, G_S, and CD were not based on detailed written instructions, we focus our analysis on the C_S and C_W benchmarks, where classification criteria were explicitly defined and suitable for prompt repurposing. Between the two, we selected C_W for presentation in the main text, as its weaker threshold (âprobablyâ rather than âcertainlyâ) represents a more inclusive and arguably more natural operationalization of the promise category. This choice also facilitates a more forgiving test of the modelâs alignment with human annotation. Nevertheless, for completeness and transparency, we report LLM classification results using all five benchmark datasets in here.

C.1.2. Original Instructions

Figure S4 displays the original codebook used by HX for their content analysis, which we repurposed as the basis for our prompts. Figure S5 shows the classification instructions given to subjects during the experiment, along with the additional verbal guidance used to differentiate between weak and strong variations. These instructions are a shortened version of the codebook, omitting the explicit classification criteria, which were instead conveyed verbally during the experiment.

C.1.3. Prompt

We have considered four different prompts: a Basic prompt, B, and three variations O_S, O_N, and O_W, of the original instructions used by HX in their traditional content classifications (C_S and C_W). All our prompts share the same following sections: “General Task”, “Context” and “Output Format”. “General Task” briefly defines the task in a single sentence; the “Context” section provides details about the type of players, game mechanics, and communication protocol; and the “Output Format” section provides a template for the model to follow when outputting its classification.

Figure S4: P_I - Original Instructions used for Content Analysis

Your task:
You will be given a list of messages. Your task is to evaluate whether each of the messages is:

- A statement of intent or promise
- Empty Talk

The messages were written by participants in a previous experiment (Experiment I). To evaluate the messages, you need to first understand Experiment I. The pages beginning on page 2 describe Experiment I. Please read those pages carefully. The message writer is in the role of subject B.

It is important for you to know more about how to code the messages before you read the instructions. Here are your specific instructions for how you code the messages:

1. *(Weak Promise)* You should code a message as ‘‘A statement of intent or promise’’ if you think at least one of the following conditions is probably satisfied.
   (Strong Promise) You should code a message as ‘‘A statement of intent or promise’’ if you think at least one of the following conditions is certainly satisfied.
   a. The writer, subject B, indicates in the message he/she would do something favorable to subject A or refrain from doing something that hurts subject A.
   b. The message gives subject A reasons to believe or expect that subject B would do something favorable to subject A or refrain from doing something that hurts subject A.
2. (Weak Promise) You should code a message as ‘‘Empty Talk’’, if the message does not probably satisfy any of the above conditions.
   (Strong Promise) You should code a message as ‘‘Empty Talk’’, if the message does not certainly satisfy any of the above conditions.
3. You should independently code all messages. Do not discuss with anyone else in this room about how to code the messages.
4. Your job is to capture what had been said rather than why it was said or what effect it had. Think of yourself as a ‘‘coding machine.’’
5. When you complete the coding, go through the entire list of messages a second time to (1) review all your codes and revise them if needed for accuracy; (2) make sure you code every message.

Figure S5: P_I - Original Instructions used for HX Classification Game

Your task:
In this experiment, you will be given a list of messages. Your task in this experiment is to evaluate whether each of the messages is:

- A statement of intent or promise
- Empty Talk

The messages were written by participants in a previous experiment (Experiment I). To evaluate the messages, you need to first understand Experiment I. The next few pages describe Experiment I. Please read it carefully. The message writer is in the role of subject B.

Subjects were also told:
Weak Promise treatment: ‘‘You should classify a message as ‘Promise or Intent’ if, in your opinion, it includes any statement of intent.’’
Strong Promise treatment: ‘‘You should classify a message as ‘Promise or Intent’ if, in your opinion, it is certainly a promise.’’

The basic prompt, B, does not include a “Role Persona” section, and its “Classification Task” section does not provide any description of what constitutes a “promise”. Instead, it combines instructing to classify a message as a “promise” or as “empty talk” with how to code this classification (1 for “promise”, 0 for “empty talk”) within the same directive.

For the three prompts that are based on the original instructions, O_S, O_N and O_W, the “Classification Task” section uses the instructions provided by HX verbatim. We adopt the same weak and strong instruction variations they considered in their C_W and C_S treatments with the prompts O_W and O_S. Additionally, we have considered a neutral version, O_N, which does not use any additional adverbs (such as “probably” or “certainly” used in O_W and O_S, respectively) to qualify the verb “satisfied” in the condition: “if at least one of the following conditions is `<adverb>` satisfied.” Moreover, these three prompts, unlike B, include a “Classification Coding” section that outlines how to code specific categories in the output (1 for promise and 0 for empty talk). Lastly, because HX’s original instructions included the statement “Operate as a coding machine”, which imposes a role, we also did not include an additional “Role Persona” section in these prompts.

The prompts O_S, O_N and O_W are displayed in Figure S6, while the basic prompt B is displayed in Figure S7. The prompt presented in the main section is O_W.

Figure S6: P_I - Prompt O_S, O_N and O_W

```
# General Task
- Evaluate a player's message in an economic experiment game to determine if it constitutes a promise.
# Context
- Players:  Two players, A and B.
- Game Mechanics:
  + First, player A chooses between the two decisions 'IN' and 'OUT'.
  + If player A chooses 'OUT', each player receives $5.
  + If player A chooses 'IN', player B chooses between the two decisions 'ROLL' or 'DON'T ROLL' (a die).
  + If player A chooses 'IN' and player B chooses 'DON'T ROLL', then player B receives $14 and A receives $0.
  + If player A chooses 'IN', and player B chooses 'ROLL', player B receives $10 and rolls a six-sided die to
determine player A's payoff.  If the die comes up 1 (1/6 likelihood), player A receives $0; if the die comes up
2-6 (5/6 likelihood), player A receives $12.
  + In addition, each player receives $5 show-up fee for participating in the experiment.
- Communication:  Prior to Player A's decision to 'IN' or 'OUT', player B has the option to send a message to
player A.
- Task:  Classify whether the message sent by player B constitutes a promise or not.
# Classification Task
(O_S)- Classify a message as a "promise or a statement of intent" if at least one of the following conditions is
**certainly** satisfied:
(O_N)- Classify a message as a "promise or a statement of intent" if at least one of the following conditions is
satisfied:
(O_W)- Classify a message as a "promise or a statement of intent" if at least one of the following conditions is
**probably** satisfied:
  + Player B indicates in the message he would do something favorable to player A or refrain from doing something
that hurts player A
  + The message gives player A reasons to believe or expect that player B would do something favorable to player A
or refrain from doing something that hurts player A.
(O_S)- If the message does not **certainly** satisfy any of the above conditions, classify player B's message as
"Empty Talk".
(O_N)- If the message does not satisfy any of the above conditions, classify player B's message as "Empty Talk".
(O_W)- If the message does not **probably** satisfy any of the above conditions, classify player B's message as
"Empty Talk".
- Capture what had been said rather than why it was said or what effect it had.
- Operate as a "coding machine".
# Classification Coding
- Code player's message as `1` if it's a promise.
- Code as `0` if it's an empty talk.
# Constraint
- Follow the below output format.
# Output Format
0/1
```

C.1.4. Additional Results

In Table S5, the columns labelled “no-CoT” and ”CoT” represent the treatments where 0-shot-CoT prompting was not incorporated and was incorporated, respectively. As

Figure S7: P_I - Prompt B

```
# General Task
- Evaluate a player’s message in an economic experiment game to determine if it constitutes a promise.
# Context
- Players:  Two players, A and B.
- Game Mechanics:
  + First, player A chooses between the two decisions ’IN’ and ’OUT’.
  + If player A chooses ’OUT’, each player receives $5.
  + If player A chooses ’IN’, player B chooses between the two decisions ’ROLL’ or ’DON’T ROLL’ (a die).
  + If player A chooses ’IN’ and player B chooses ’DON’T ROLL’, then player B receives $14 and A receives $0.
  + If player A chooses ’IN’, and player B chooses ’ROLL’, player B receives $10 and rolls a six-sided die to
determine player A’s payoff.  If the die comes up 1 (1/6 likelihood), player A receives $0; if the die comes up
2-6 (5/6 likelihood), player A receives $12.
  + In addition, each player receives $5 show-up fee for participating in the experiment.
- Communication:  Prior to Player A’s decision to ’IN’ or ’OUT’, player B has the option to send a message to
player A.
- Task:  Classify whether the message sent by player B constitutes a promise or not.
# Classification Task
- Classify player’s message as ‘1‘ if it’s a promise.
- Classify as ‘0‘ if it’s an empty talk.
# Constraint
- Follow the below output format.
# Output Format
   0/1
```

		no-CoT					**CoT**				
		CD	C_S	C_W	G_S	G_W	CD	C_S	C_W	G_S	G_W
GPT-3.5	B	71	76	79	71	90	82	82	84	82	84
	O_S	71	76	79	71	90	79	84	87	79	**92**
	O_N	68	74	76	68	87	74	79	82	74	87
	O_W	68	74	76	68	87	76	82	84	76	90
GPT-4	B	87	87	84	87	74	90	90	87	95	76
	O_S	92	92	90	**97**	79	**97**	92	90	92	84
	O_N	92	**97**	95	**97**	84	92	92	95	92	90
	O_W	84	95	96	90	87	90	95	**97**	90	**92**

Table S5: Overall Accuracy of Promise Classification in %.

can be observed in Table S5, GPT-3.5 demonstrates a negligible degree of responsiveness to the variations in the classification instructions of the prompts. Specifically, in no-CoT treatment, B and O_S, and O_N and O_W generate identical classifications. Moreover, the model’s classifications under O_W differ from those under O_S with just one additional message classified as “promise.”

CoT prompting consistently improves GPT-3.5’s performance and leads to a greater degree of variation in the classifications across prompts. However, this variation does not necessarily imply that CoT prompting improves GPT-3.5’s adherence to instructions. Ideally, if CoT prompting were effectively increasing the model’s responsiveness to instructional nuances, O_W would demonstrate the highest performance in the G_W benchmark, and O_S in the G_S benchmark. Yet, O_S consistently outperforms O_W. Furthermore, more detailed instructions tend to deteriorate GPT-3.5’s performance: B

		no-CoT				CoT			
		B	O_S	O_N	O_W	B	O_S	O_N	O_W
GPT-3.5	P	35	35	36	36	29	32	34	33
	E	3	3	2	2	9	6	4	5
	f_P	.92	.92	.95	.95	.76	.84	.90	.87
GPT-4	P	21	23	25	27	22	25	27	28
	E	17	15	13	11	16	13	11	10
	f_P	.55	.61	.66	.71	.58	.66	.71	.74

Table S6: Aggregate classification results of different prompts

outperforms O_W and O_N under all benchmarks except G_W, and O_S only surpasses B under the benchmarks characterized by weaker instructions, namely C_W and G_W.

GPT-4 consistently outperforms GPT-3.5 except under G_W benchmark. In Table S6, relatively larger variation in the number of messages classified as "promise" observed for GPT-4 is indicative of the model's responsiveness to the instructional variations. Furthermore, its responsiveness to the instructional nuances is reflected to some degree in its classification performance. As can be observed in Table S5, O_W consistently achieves the highest accuracy scores under the G_W and C_W benchmarks, and under the G_S benchmark, O_S and O_N are tied for the best performance. On the other hand, in no-CoT treatment, O_N achieves the highest accuracy under the C_S benchmark, surpassing the performance of O_S.

For GPT-4, CoT prompting does not consistently improve performance. The highest performances for benchmarks C_S and G_S are achieved in no-CoT treatment using O_S or O_N. Conversely, the highest performances for benchmarks C_D, C_W, and G_W are achieved in CoT treatment using O_S or O_W. Given the top performing results under benchmarks with weaker "promise" classification conditions are achieved in CoT treatment and the top performing results under the benchmark with the stronger "promise" classification condition are achieved in no-CoT treatment suggests that CoT prompting introduces a bias towards classifying messages as promises.

C.2. Promise II: Public Good Game

C.2.1. Game and Data

Authors of (Arad, Hugh-Jones and Penczynski, 2024, henceforth AHP) carried out an online experiment to understand which kind of communication predicts cooperation. 633 participants engaged in five identical 3-player public good games, each with different anonymous opponents. In every game, the three players had the opportunity to chat before making a decision using a built-in platform resembling WhatsApp. The messages were classified according to the presence of a promise by two RAs. On the basis of the free-flowing chat between the three players, the classification indicates for each individual player, whether a promise was made. In total, 717 chat instances were analysed for classification. The RAs reached consensus on 89.9% of the instances, with a Krippendorff's α of 0.798, indicating substantial inter-rater reliability. Within these agreements, 53.3% were classified as "promise", hence the two categories are balanced in the dataset.

C.2.2. Original Instructions

In Figure S8, the original instructions are presented with certain sections omitted that are not relevant to the prompt. Additionally, to accommodate the instructions on a single page, various line spacings have been reduced. The actual format of the original instructions is much easier to follow (see Arad et al. (2024) for further information).

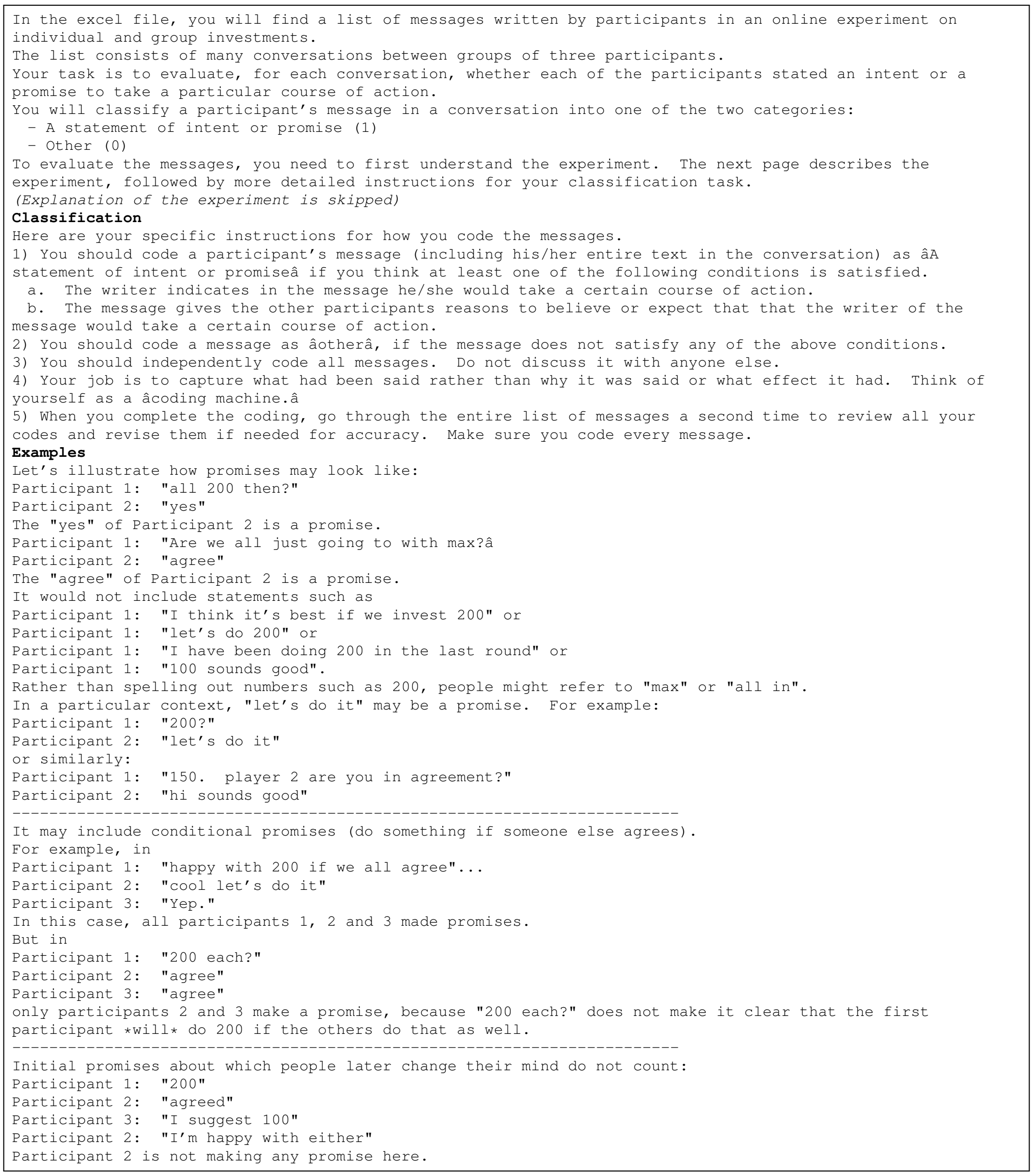
C.2.3. Prompt

We have considered two distinct prompts: a basic prompt, B, and a prompt that is the reframed version of the original instructions, O. Similar to the approach in the "Promise I" section, all our prompts include the same subsequent sections: "General Task", "Context," and "Output Format". The "General Task" section briefly defines the task in a single sentence. The "Context" section elaborates on the types of players, game mechanics, and communication protocol. Finally, the "Output Format" section outlines the template the model should use to format its classification output.

Since the classification instructions in the HX and AHP prompts are identical, B and O closely resemble the prompts B and O_N from the "Promise I" section, respectively. Consequently, all information pertaining to prompts B and O_N in Section C.1.3 also applies to B and O. Specifically for their 0-shot versions, the primary distinction lies in the "Context" section. The "Context" section of the prompts introduced in Section C.1.3 details the investment game and features a standalone message from player A to B, whereas in this section, the "Context" section of the prompts describes the public good game involving a conversation among three players.

The codebook of AHP, unlike that of HX, includes a set of examples accompanying its classification instructions. AHP use these demonstrations to provide more nuanced conditions for the "promise" and "empty talk" categories. This section is structured as a sequence of examples, each followed by a remark that highlights a specific case of

Figure S8: P_{II} - Original Instructions



```
In the excel file, you will find a list of messages written by participants in an online experiment on
individual and group investments.
The list consists of many conversations between groups of three participants.
Your task is to evaluate, for each conversation, whether each of the participants stated an intent or a
promise to take a particular course of action.
You will classify a participant’s message in a conversation into one of the two categories:
  - A statement of intent or promise (1)
  - Other (0)
To evaluate the messages, you need to first understand the experiment.  The next page describes the
experiment, followed by more detailed instructions for your classification task.
(Explanation of the experiment is skipped)
Classification
Here are your specific instructions for how you code the messages.
1) You should code a participant’s message (including his/her entire text in the conversation) as âA
statement of intent or promiseâ if you think at least one of the following conditions is satisfied.
  a.  The writer indicates in the message he/she would take a certain course of action.
  b.  The message gives the other participants reasons to believe or expect that that the writer of the
message would take a certain course of action.
2) You should code a message as âotherâ, if the message does not satisfy any of the above conditions.
3) You should independently code all messages.  Do not discuss it with anyone else.
4) Your job is to capture what had been said rather than why it was said or what effect it had.  Think of
yourself as a âcoding machine.â
5) When you complete the coding, go through the entire list of messages a second time to review all your
codes and revise them if needed for accuracy.  Make sure you code every message.
Examples
Let’s illustrate how promises may look like:
Participant 1:  "all 200 then?"
Participant 2:  "yes"
The "yes" of Participant 2 is a promise.
Participant 1:  "Are we all just going to with max?â
Participant 2:  "agree"
The "agree" of Participant 2 is a promise.
It would not include statements such as
Participant 1:  "I think it’s best if we invest 200" or
Participant 1:  "let’s do 200" or
Participant 1:  "I have been doing 200 in the last round" or
Participant 1:  "100 sounds good".
Rather than spelling out numbers such as 200, people might refer to "max" or "all in".
In a particular context, "let’s do it" may be a promise.  For example:
Participant 1:  "200?"
Participant 2:  "let’s do it"
or similarly:
Participant 1:  "150.  player 2 are you in agreement?"
Participant 2:  "hi sounds good"
------------------------------------------------------------------------
It may include conditional promises (do something if someone else agrees).
For example, in
Participant 1:  "happy with 200 if we all agree"...
Participant 2:  "cool let’s do it"
Participant 3:  "Yep."
In this case, all participants 1, 2 and 3 made promises.
But in
Participant 1:  "200 each?"
Participant 2:  "agree"
Participant 3:  "agree"
only participants 2 and 3 make a promise, because "200 each?" does not make it clear that the first
participant *will* do 200 if the others do that as well.
------------------------------------------------------------------------
Initial promises about which people later change their mind do not count:
Participant 1:  "200"
Participant 2:  "agreed"
Participant 3:  "I suggest 100"
Participant 2:  "I’m happy with either"
Participant 2 is not making any promise here.
```

promise or empty talk classification, then more examples and subsequent remarks, and so on. We adopted this “Example” section verbatim in O, modifying its format to align with our prompt template: each example is separated and indexed with a subtitle, such as “Example #1”, followed by the content of the example, and each remark section between sets of examples is distinguished with a “Remark” subtitle, followed by the remark. Consequently, unlike O_N from Section C.1.3 and prompt B, O includes an “Example” section, although this additional section is incorporated into the prompt only for n-shot treatments.

In total, 11 chat examples are provided for AHP, making our n-shot treatment for O an 11-shot setup. Since we have adopted the example section of the original instructions verbatim, the format of our prompt's example section diverges from the conventional <question, answer> format typically used in n-shot prompting. Instead, it includes explanations between sets of examples. However, it also does not adhere to the traditional n-shot-CoT prompting format of <question, explanation, answer>, as the provided explanations (remarks) are sparse and do not offer detailed rationales for classifying sets of messages. In summary, the "Example" section of O provides more information than a traditional n-shot prompt but less than an n-shot-CoT prompt. Nevertheless, the presence of remarks can still be seen as offering partial rationales, and therefore, their inclusion in O is expected to enhance the model's reasoning capabilities when CoT prompting is utilised.

Figure S9: P_{II} - Prompt B

```
# General Task
- Evaluate each player in an investment game to
determine whether he makes a promise or not.

# Context
- Players:  Group of three.
- Initial Endowment:  200 pence each.
- Investment:  Maximum 200 pence each.
- Mechanics:  Invested amount is doubled and
split equally.
- Communication:  Players can chat before
investing.
- Duration:  Multiple rounds.

# Classification Task
- Classify a player as `1` if he made a promise.
- Classify as `0` otherwise.

# Constraints
- Refrain from providing an explanation for your
classification.  (only for no-CoT cases)
- Provide a final and single classification for
each player.
- Follow the below output format.

# Output Format
P# :  0/1
```

Figure S10: P_{II} - Prompt O

```
# General Task
- Evaluate each player in an investment game to
determine whether he makes a promise or not.

# Context
- Players:  Group of three.
- Initial Endowment:  200 pence each.
- Investment:  Maximum 200 pence each.
- Mechanics:  Invested amount is doubled and
split equally.
- Communication:  Players can chat before
investing.
- Duration:  Multiple rounds.

# Classification Task
- Classify a player’s message as "a statement
of intent or a promise" if at least one of the
following conditions is satisfied:
  + The message indicates that the player will
take a certain course of action.
  + The message gives others reason to believe
or expect that the player will take a certain
course of action
- If the message does not satisfy either of
the above conditions, classify it as an "Empty
Talk".
- Capture what had been said rather than why it
was said or what effect it had.
- Operate as a "coding machine".

# Classification Coding
- Code player’s message as `1` if it’s a promise
or statement of intent.
- Code as `0` if it’s an empty talk.

# Examples (Only in n-shot treatment) (See
Figure S11)

# Constraints
- Refrain from providing an explanation for your
classification.  (only for no-CoT cases)
- Provide a final and single classification for
each player.
- Follow the below output format.

# Output Format
P# :  0/1
```

Figure S11: P_{II} - “Examples” for Prompt O

```
# Examples (Only in n-shot treatment)
## Example 1
### Chat
P1:  all 200 then?
P2:  yes
### Classification
P1:  0
P2:  1
## Example 2
### Chat
P1:  Are we all just going to with max?
P2:  agree
### Classification
P1:  0
P2:  1
## Example 3
### Chat
P1:  I think it’s best if we invest 200
### Classification
P1:  0
## Example 4
### Chat
P1:  let’s do 200
### Classification
P1:  0
## Remark
- Rather than spelling out numbers such as 200, people might refer to "max" or "all in".
## Example 5
### Chat
P1:  I have been doing 200 in the last round
### Classification
P1:  0
## Example 6
### Chat
P1:  100 sounds good
### Classification
P1:  0
## Remark
- In a particular contexts, "let’s do it" or "sounds good" may be a promise (examples 7 and 8).
## Example 7:
### Chat
P1:  200?
P2:  let’s do it
### Classification
P1:  0
P2:  1
## Example 8
### Chat
P1:  150.  player 2 are you in agreement?
P2:  hi sounds good
### Classification
P1:  1
P2:  1
## Remark
- A message may include conditional promises (do something if someone else agrees) (example 9).
## Example 9:
### Chat
P1:  happy with 200 if we all agree
P2:  cool let’s do it
P3:  Yep.
### Classification
P1:  1 (conditional promise)
P2:  1
P3:  1
## Remark
- In example 10, only players 2 and 3 make a promise, because "200 each?" does not make it clear that
the first participant *will* do 200 if the others do that as well.
## Example 10
### Chat
P1:  200 each?
P2:  agree
P3:  agree
### Classification
P1:  0
P2:  1
P3:  1
## Remark
- Initial promises about which player later change their mind do not count (example 11).
## Example 11
### Chat
P1:  200
P2:  agreed (initial promise)
P3:  I suggest 100
P2:  I’m happy with either (P2 changes her mind)
### Classification
P1:  1
P2:  0 (due to change of mind after an initial promise)
P3:  0
```

C.3. Intra-team communication

Messages in L_I and L_{II} datasets are generated by the intra-team communication protocol that was introduced in Burchardi and Penczynski (2014) and classified according to the level-k of strategic reasoning. Teams of two subjects play as one entity and exchange arguments as follows. Both subjects individually make a suggested decision and write up a justifying message. Upon completion, this information is exchanged simultaneously and both subjects can enter individually a final decision. The computer draws randomly one final decision to be the team's action in the game. The protocol has the advantage of recording the arguments of the individual player at the time of the decision making. Furthermore, the subject has incentives to convince his team partner of his reasoning as the partner determines the team action with 50% chance.

C.4. Level-k I: Jury Voting Game

C.4.1. Game and Data

In Çelebi and Penczynski (2023), we propose a level-k model of strategic thinking in jury voting (JV) games à la Feddersen and Pesendorfer (1998) and Guarnaschelli et al. (2000). In juries of size 3 or 6, jurors receive informative signals (red or blue balls) and then vote to acquit (blue) or convict (red) the defendant, with the jury decision ideally matching the innocence (blue urn) or guilt (red urn) of the defendant. Looking at juries under the unanimity rule for conviction, we show that the jury performance depends on the strategic sophistication of jury members, which in turn depends on the complexity of the task at hand.

Our model assumes non-strategic, random level-0 play, to which a level-1 player best-responds by always voting according to the received informative signal. Given the unanimity rule, the best response to informative voting by level-2 players is to strategically vote always "convict" to make a conviction more likely and rely on other voters to acquit. For level-3 players, the best-response to always convicting is to play informatively like level-1 players do.

The messages are independently classified according to this level-k model by two RAs. The RAs are introduced to the level-k model and received detailed instructions about characteristics of the individual level-k types.

The classification procedure starts with both RAs providing independent sets of classifications. Then, both are anonymously informed about the classifications of the other RA and have the possibility to simultaneously revise their own classification. This revision process is repeated twice. After the process, the two RAs agreed on 93.2%

(493) of the classifications.

As can be observed in Figure S7, the distribution of levels is non-degenerate and features a heterogeneity of types, a hump-shape with mode behaviour at level-1, and hardly any level-3 behaviour. This is a standard distribution commonly observed in similar studies and hence represents the type of distribution to be expected when researchers consider to classify such data (Camerer et al., 2004; Costa-Gomes and Crawford, 2006; Burchardi and Penczynski, 2014; Crawford et al., 2013). Their agreed classifications for 493 messages constitute the benchmark for the LLMs in this study.

	L0	L1	L2	L3
f_L	.21	.49	.29	.01

Table S7: Level Distribution

C.4.2. Original Instructions

We have omitted various sections of the original instructions to fit in the instructions into a single page. The original instructions begins with a lengthy section regarding the general theory of level-k modeling and its specific application to the experiments. These sections aggregate to 3 pages of instructions. Furthermore, some sentences in the instructions that were not related to the classification of instructions, such as how to code certain concepts in the excel sheet, are also omitted. See Çelebi and Penczynski (2023) for the instructions in its entirety.

C.4.3. Prompt

The prompt O generated using the original instructions is presented in Figure S12. The original codebook's classification instructions for human annotators begin with a "General Comments" subsection that notes the potential implicit nature of messages and instructs annotators to classify messages at the level they believe is most likely when uncertain. This section is followed by subsections for each level from 0 to 3, each further containing "Characteristics", "Examples" and "Note" subsections. The "Characteristics" subsection outlines the observed traits of that level of thinking within the voting game, while the "Note" subsection provides additional guidance for handling ambiguous cases. The "Examples" subsections vary in number, with five examples for level-0, three for level-1, eight for level-2, and three for level-3. There are no specific comments provided for individual examples.

Figure S12: L_I - Original Classification Instructions

General Comments:
Subjects do not necessarily describe every step of their thinking; therefore, it may not always be obvious to decide which level they are. In many comments, any indications of a level of thinking may be partial or implicit, you should then indicate the most likely level of reasoning of the player. If the message indicates to simply refer to a previous message ("same as before/above"), then you can use the previous message's evaluation to determine the level of the current message. If you are unsure of the level of the message, you should indicate the level you think is more likely.

Level-0 Player:
Characteristics: Chooses randomly, without justification or through some justification completely unrelated to the task. Might not have understood the game or shows no interest in the game or in thinking about it.
Examples:
"50 50 chance to get red at least 50 50 could also be 100 percent."
"I like blue, so I chose blue."
"Think it will be red again."
"definitely red this time"
"We have to go for red. No other way than that. I like turtles"
Note: Comments such as "It is obviously blue" or "Play red, trust me!" should not be considered as level-0 thinking as these comments to some extent signal some level of understanding/interest of the task. Such comments are likely to be level-1 comments yet without any additional information, you should leave the specific cell empty.

Level-1 Player:
Characteristics: Always follows his own signal. The subject may argue in favor of playing his own signal through some probability argument
Examples:
"Our signal is blue. Let's play blue."
"The probability that the red ball we observe is out of the red urn is twice the probability that it is out of the blue urn"
"1/3 of all teams is observing wrong color, so we would try to find out whether we have wrong or right ball, keep with red."
Note: The key idea in defining a level-1 player is to identify some thinking process that signals the subject's interest/understanding of the task and the private signal. Furthermore, it is important that the subject does **not** offer any argument acknowledging the potential votes of the other teams and how to vote accordingly (i.e. adjusting the strategy given what others are expected to do).

Level-2 Player:
Characteristics: Assume that all other players almost always follow their signal (i.e. she assumes almost all the other players are level-1 while an epsilon portion of them are level-0). Player does offer an argument acknowledging the potential votes of the other teams and how to vote accordingly (i.e. a best response given others are most likely playing their signal). In other words, if you identify any comment that indicates that the subject assumes (or considers the case) where the other players in her group play their signal, you should consider the possibility that the subject is a level-2 player.
Examples:
"Let's take red because if the urn is red and we got the opposite color and we take blue, the decision will be blue."
"We need to chose Red. If we are the only ones who picked blue, then the urn is red and we guess correct If the urn is blue, then the other guys will pick blue so there will be at least one blue vote and we win as well If the others guys (also blue) think the same way then we lose But this is too many ifs"
"I have a blue ball. If we have the blue urn, someone else also has a blue ball and as a result our group will chose blue regardless of my vote. If we have the red urn, I am the only one with the blue ball and if I vote blue, we will chose the wrong urn. So I should vote for red."
"In case two teams choose red and one chooses blue, blue will be taken. That means that choosing red has a higher chance of being a good decision."
"I guess this is more about luck because there is no way to know it for sure. I would say blue just because of the higher probability. Also I like turtles Also it is likely that one other team will pick blue and then it is that color anyways"
"There is no point for us to take blue I think the chances for us to get the right color are higher if we stick with red" [red ball is observed]
"I suggest red because we donat hurt anyone with this decision If the others go for blue because they have a blue ball, the committeeas decision will be blue regardless of our decision"
"We could be the deciding vote for blue if the other two choose red. Choosing blue isnt as helpful as choosing red, because: only one blue ball can overturn our whole decision but only a unanimous decision for red can help us the same way"
Note: In order to discern the two types, you should look for more than any trivial arguments such as the ones given under level-1. There may be cases where the message starts as a level-1 argument and then as the subjects elaborates on her reasoning, she starts considering the strategy of the other teams and justify her decision accordingly (see the third example above). In such cases, this message should be considered as level-2. The acknowledgment of other teams' voting strategy may not always be obvious or may be worded differently such as "hurting the other's decision" or "not being helpful" (see the last three examples above)

Level-3 Player:
Characteristics:
Assumes that almost all other subjects are level-2 players (partially degenerate beliefs). The reasoning in a level-3 player message will have similarities with a level-1 player message but it will have additional arguments indicating that she assumes others are level-2 players.
Examples:
"In my opinion, if there is another person with blue they may be afraid of voting blue so we should vote blue to make sure."
"Let's now pick the shown colour because the others now will probably enter their opposite colour."
"Risky to vote blue but others may not vote blue even when they draw blue. I say we vote blue."
Note: As stated above, level-3 players are likely to follow their signal like a level-1 player yet they will argue to do so through a much more intricate argument (unlike a level-1 player merely stating probabilities to argue her action). Level-3 players are rare. Higher levels (level-4 etc.) are assumed to not occur; therefore, you should consider only the first 4 levels of thinking.

These classification instructions are incorporated verbatim into the prompt O, preserving the structure and format of the original level subsections and their respective "Characteristics", "Examples" and "Note" subsections. The only exception is the "Note" section of the level-0 subsection. In the codebook, this "Note" section instructs annotators to leave a message unclassified if it contains ambiguous phrases like "Play red, trust me!" that likely indicate level-1 thinking but lack additional context to confirm it. Given that the messages considered for classification are those where both annotators found enough information to classify, this specific instruction is redundant for the models and therefore omitted in the prompts. Additionally, the instruction in the "General Comments" section to classify messages based on the most likely level when unsure contradicts this instruction to leave messages unclassified when there is not enough context. Hence, to maintain instructional consistency with our prompts, we adhered to the "most likely level" instruction. Lastly, this modification also avoids the risk of the model misinterpreting the option to leave messages unclassified and incorrectly generalising it to other unrelated but potentially ambiguous messages and other levels of thinking.

Prior to the classification instruction section, the codebook includes a three-page explanation of the experiment, general level-k theory, and the application of level-k theory to the voting game. We have omitted these sections and instead included the most essential information in the "Context" section of the prompt. This section provides details on general game mechanics and communication protocols, aimed at providing the model with necessary context and textual patterns for its classification task. One should consider the provided information in the "Context" section of the prompt as an executive summary of these sections. Hence, we made the implicit assumption that GPT has prior knowledge on the jury voting game, the general level-k theory, and the application of the level-k to the voting game. We have indeed investigated this assumption by asking a series of questions to both models, and ensured that it does have a detailed knowledge of the theory and the jury voting game. Hence, both models has enough prior knowledge to be simply presented with an executive summary of these sections. On the other hand, either model is unable to apply the level-k theory to the jury voting game in a correct fashion to autonomously define each level of thinking for the game. Hence, without the provision of specific characteristics of different levels of thinking, it is unclear how GPT could interpret them or distinguish, say, a level-1 thinker from a level-2 thinker. Given this uncertainty, we do not consider a basic prompt B for this classification task.

In the "General Task" section, the model is instructed to classify a player's level of strategic thinking in a voting game. This is followed by the “Role Persona” section, where the model is instructed to act as a behavioural economist specialised in level-k modelling, strategic thinking, and text classification. Other auxiliary sections, such as “Classification Process”, “Constraint”, and “Output Format”, follow the format presented in Section 1.B.1, and are almost identical to prompts from previous sections C.1.3 and C.2.3.

Figure S13: L_I - Prompts O

```
# General Task
- Classify player’s level of strategic thinking in a voting game
# Role Persona
- Act as a behavioural economist specialized in level-k modeling, strategic thinking and text classification.
# Context (see Figure S14)
# Classification Task
- Classify a player’s level of strategic thinking as 0, 1, 2 or 3 based on the message provided.
- Use the below characteristics, examples and notes provided for each level to determine your classification.
## Level-0 Player
### Characteristics
- Chooses randomly, without justification or through some justification completely unrelated to the task.
- Might not have understood the game or shows no interest in the game or in thinking about it.
- Provides a vote without a clear justification to the probability of the game or strategic reasoning.
### Note
- Comments such as ’It is obviously blue’ or ’Play red, trust me!’should not be considered as level-0 thinking as
these comments to some extent signal some level of understanding/interest of the task.  Such comments are likely
to be level-1 comments.
## Level-1 Player
### Characteristics
- Always follows his own signal.
- The subject may argue in favor of playing his own signal through some probability argument.
### Note
- The key idea in defining a level-1 player is to identify some thinking process that signals the subject’s
interest/understanding of the task and the private signal.
- It is important that the subject does not offer any argument acknowledging the potential votes of the other
teams and how to vote accordingly (i.e.  adjusting the strategy given what others are expected to do).
## Level-2 Player
### Characteristics
- Assume that all other players almost always follow their signal (i.e.  a level-2 player assumes almost all the
other players are level-1 while a small portion of them are level-0).
- Player does offer an argument acknowledging the potential votes of the other teams and how to vote accordingly
(i.e.  a best response given others are most likely playing their signal).
- if you identify any comment that indicates that the subject assumes (or considers the case) where the other
players in her group play their signal, you should consider the possibility that the subject is a level-2 player.
### Note
- In order to discern between level-1 and level-2 types, you should look for more than any trivial arguments such
as the ones given under level-1.
- There may be cases where the message starts as a level-1 argument and then as the subjects elaborates on her
reasoning, she starts considering the strategy of the other teams and justify her decision accordingly (see the
fifth example above).  In such cases, this message should be considered as level-2.
- The acknowledgment of other teams’ voting strategy may not always be obvious or may be worded differently such
as "hurting the other’s decision" or "not being helpful" a (see the last three examples above)
## Level-3 Player
### Characteristics
- Assumes that almost all other subjects are level-2 players (partially degenerate beliefs).
- The reasoning in a level-3 player message will have similarities with a level-1 player message but it will have
additional arguments indicating that she assumes others are level-2 players.
### Notes
- level-3 players are likely to follow their signal like a level-1 player yet they will argue to do so through a
much more intricate argument (unlike a level-1 player merely stating probabilities to argue her action).
- Level-3 players are rare.
## General Comments
- Players do not necessarily describe every step of their thinking; therefore, it may not always be obvious to
decide which level they are.  In many messages, any indications of a level of thinking may be partial or implicit.
In such cases provide the most likely level of reasoning from the messages.
- If you are unsure of the level of the message, you should indicate the level you think is more likely.
# Constraint
- Only provide a single level classification.
- Follow the below output format.
# Output Format
0/1/2/3
```

Figure S14: L_I - Prompt O - “Context” Section

```
# Context
- Teams of two players, part of larger groups, draw a colored ball from an urn, play a voting game with their
group to guess the color of the urn for multiple periods.
- Teams are randomly paired each period.  Teams consist always of 2 players.
- Teams are grouped into either 3 or 6 teams per group.
- Each period, a group is assigned to an urn with a blue or red color with equal probability.
- An urn only contains blue and red balls.  Blue urn has twice more blue balls than red balls and a red urn has
twice more red balls than blue balls.  The color of the ball has a 2/3 chance of matching the urn’s color.
- After the urn with either red or blue color is assigned to a group, each team within a group draws a ball from
the urn to infer the urn’s color.  The drawing can be with or without replacement depending on the period.
- Teams do not know the color of the urn.  They do not know the colors of the balls picked by the other teams in
their group.  They only know the color of their own ball.
- The group’s objective is to correctly guess the color of the assigned urn.
- Each team in a group provides a single vote.  A team votes for either the color red or blue.  Group’s decision
is determined based on the aggregation of its teams’ votes.
- Teams communicate internally to decide on a vote for the urn’s color.  If all votes are red, the group decision
is red; any blue vote results in a blue group decision.
- Teams do not observe the votes of the other teams in their group.
- Teams weigh their own ball’s color and strategize their vote considering the group’s outcome.
- Players exhibit levels of strategic reasoning (0 to 3), influencing their decision-making and messaging.
```

Figure S15: L_I - Prompt O - “Examples” Section

```
# Classification Task
- ...
## Level-0 Player
### Characteristics ...
### Examples
1.  "50 50 chance to get red at least 50 50 could also be 100 percent."
2.  "I like blue, so I chose blue."
3.  "Think it will be red again."
4.  "definitely red this time"
5.  "We have to go for red.  No other way than that.  I like turtles"
### Notes ...
## Level-1 Player
### Characteristics ...
### Examples
1.  "Our signal is blue.  Let’s play blue."
2.  "The probability that the red ball we observe is out of the red urn is twice the probability that it is out of
the blue urn"
3.  "1/3 of all teams is observing wrong color, so we would try to find out whether we have wrong or right ball,
keep with red."
### Notes ...
## Level-2 Player
### Characteristics ...
### Examples
1.  "Let’s take red because if the urn is red and we got the opposite color and we take blue, the decision will be
blue."
2.  "We need to chose Red.  If we are the only ones who picked blue, then the urn is red and we guess correct If
the urn is blue, then the other guys will pick blue so there will be at least one blue vote and we win as well If
the others guys (also blue) think the same way then we lose But this is too many ifs"
3.  "I have a blue ball.  If we have the blue urn, someone else also has a blue ball and as a result our group
will chose blue regardless of my vote.  If we have the red urn, I am the only one with the blue ball and if I vote
blue, we will chose the wrong urn.  So I should vote for red."
4.  "In case two teams choose red and one chooses blue, blue will be taken.  That means that choosing red has a
higher chance of being a good decision."
5.  "I guess this is more about luck because there is no way to know it for sure.  I would say blue just because
of the higher probability.  Also I like turtles Also it is likely that one other team will pick blue and then it
is that color anyways"
6.  "There is no point for us to take blue I think the chances for us to get the right color are higher if we
stick with red" [red ball is observed]
7.  "I suggest red because we donat hurt anyone with this decision If the others go for blue because they have a
blue ball, the committeeas decision will be blue regardless of our decision"
8.  "We could be the deciding vote for blue if the other two choose red.  Choosing blue isnt as helpful as
choosing red, because:  only one blue ball can overturn our whole decision but only a unanimous decision for red
can help us the same way"
### Notes ...
## Level-3 Player
### Characteristics ...
### Examples
1.  "If everyone else assumes others play their own signal then they will always play red.  Since I have the blue
ball, it is more likely that we have the blue urn so I will vote blue"
2.  "Let’s now pick the shown colour because the others now will probably enter their opposite colour."
3.  "Risky to vote blue but others may not vote blue even when they draw blue.  I say we vote blue."
### Notes ...
```

X-Y games (CGR notation)	a	π_1, π_2	**Pie games** (CGR notation)	a	π_1, π_2
Symmetric Payoffs (**SL**)	X	5, 5	Symmetric Payoffs (**S1**)	L (\$)	5, 5
	Y	5, 5		R (#)	5, 5
				B (§)	5, 5
Slight Asymmetry (ASL)	X	5, 5.1	Symmetric Payoffs (**S2**)	L (\$)	6, 6
	Y	5.1, 5		R (#)	6, 6
				B (§)	5, 5
Moderate Asymmetry (AML)	X	5, 6	Moderate Asymmetry (AM2)	L (\$)	5, 6
	Y	6, 5		R (#)	6, 5
				B (§)	6, 5
Large Asymmetry (ALL)	X	5, 10	Moderate Asymmetry (AM4)	L (\$)	6, 7
	Y	10, 5		R (#)	7, 6
				B (§)	7, 5

Table S8: Payoff structure of coordination games.

C.5. Level-k II: Asymmetric-Payoff Coordination Games

C.5.1. Game and Data

Finally, van Elten and Penczynski (2020) study asymmetric-payoff coordination games (APC) introduced by (Crawford et al., 2008, henceforth CGR) and provide textual data supporting the result that the incidence of level-k reasoning is low in symmetric, pure coordination games and high in asymmetric, "battle of the sexes"-type coordination games. The dataset is of particular interest, because its text analysis involves the classification of non-trivial level-0 beliefs.

Table S8 describes the four X-Y games and four Pie games. In contrast to payoff-symmetric games (in bold), payoff-asymmetric games feature a higher coordination payoff π for one of the two players, depending on the action on which they coordinate. The miscoordination payoff is 0 for both players. The choice is between letters X and Y in the X-Y games and between 3 pie slices (L, R, B) which are identified by (\$, #, §) and of which B is uniquely white.

The dataset consists of 851 messages gathered through intra-team communication as described in Section C.3. All messages are in German. The benchmark classifications are derived from the agreed assessments of two RAs.[20] The RAs provide a lower bound and an upper bound for the level of reasoning in each message. They also identify whether any label or payoff salience argument is present, and if so, classify the type of salience.

Tables S9 and S10 show the distributions of the lower and upper bound levels where

[20] See van Elten and Penczynski (2020) for details.

L_n indicates the level-n strategic thinking. Tables S11 and S12 show the distributions of the benchmark classifications for label and payoff salience. In these tables, "$\sim$" indicates indifference to label or payoff salience, "no" signifies that there is no mention of a payoff or label salience in the text, H and L denote high and low payoff salience respectively, and $\S$, $\#$, $\$$, X and Y represent the label salience for the game choice with the same tag.

Table S9: Level distributions of the benchmark (Lower Bound)

	L_0	L_1	L_2	L_3	L_4	L_5
f	.504	.334	.143	.016	.002	–

Table S10: Level distributions of the benchmark (Upper Bound)

L_0	L_1	L_2	L_3	L_4	L_5
.341	.370	.234	.042	.006	.007

Table S11: Distribution for label salience classification of the benchmark

	$\S$	$\#$	$\sim$	$\$$	X	Y	no
f	.159	.025	.024	.107	.103	.002	.581

Each participants goes through all eight games prior to any communication exchange. We dropped those messages in which subjects refer to a reasoning that they laid out in an earlier round, as we did not want the LLM to get into assigning reasoning from earlier messages as the RAs did. In total 104 messages (12.2% of the data) are dropped.

C.5.2. Original Instructions

This codebook displayed in Figures S16 and S17 is the most detailed and extensive among those used in this study.

C.5.3. Prompt

No basic prompt B is considered here for the same reasons outlined in Section C.4.3. Using the original codebook, prompt O is generated as shown in Figures S19 and S20. The "Context" sections of the prompt begin with an identical short subsection on game mechanics, followed by a subsection on the two coordination games played in the experiments where each game type (Pie and X-Y) is briefly introduced, and payoff tables for the variations of each game type is provided in a similar table format as in Table S8

Table S12: Distribution for payoff salience classification of the benchmark

$\sim$	H	L	no
.722	.228	.039	.008

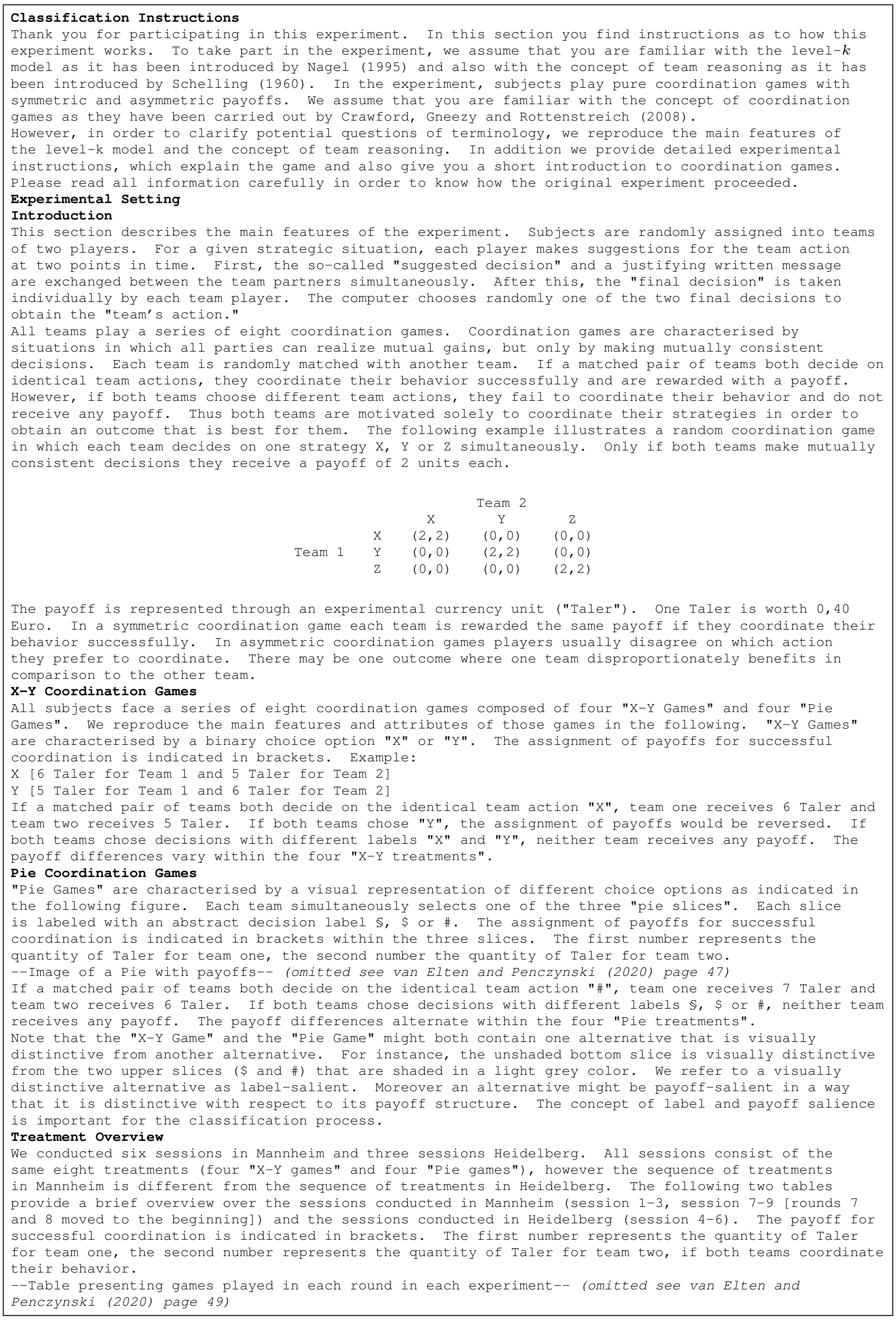
in Markdown format. The theory subsection provides an executive summary consisting of a series of subsections, each with descriptions of the fundamental concepts detailed in the original codebook. The first subsection, "Salience in Decisions", describes the concept of salience in the games. This is followed by a "Level-k Model" subsection, which details information on level-0 thinking, followed by a brief level-1 thinking description. We omit explicit description for each of the higher levels of thinking, and instead provide a brief generalised characterisation of a level-k thinker for $k > 1$.

In the "Classification Task" section, we only provide two lines of instructions that simply instruct the model to classify the payoff or label salience, and lower and upper bound of level of thinking. Additionally, we omit any examples in either of these subsections. In other words, we do not concern ourselves too much with whether the model fully "understands" the concepts by considering a significantly brief way of presenting the background information and the classification instructions.

Prompt O's "General Task", "Role Persona", "Classification Coding", "Constraint", "Classification Process" and "Output Format" sections are as previously defined in Section 1.B.1. However, differently from the prompt template, this prompt has an additional section called "Input Format". Human annotators receive contextual information about the game, the team, and the subject's initial decisions, which are essential for understanding the message. The exact game being played, as shown in Figure S8, and the team of the player are necessary for both human annotators and GPT models to effectively grasp the context of the message. Consequently, unlike in our previous classification experiments, GPT is provided not only with the subject's message but also with information about subject's team, the game she is referring to, and her initial decisions. The "Input Format" section provides the template for the input that the model receives. By including this section in the prompt, we aim to help the model better orient itself when provided with the defined input string.

O without any demonstrations establishes a 0-shot baseline for the prompt, and enables us to investigate the effect of incorporating demonstrations (n-shot prompting) in a modular fashion. We do so by developing an "Examples" section. The examples provided in this section are the examples provided in the original codebook. The original classification instructions, and therefore prompt O, include only five examples. These examples provide very limited coverage given the array of potential examples

Figure S16: L_{II} - Original Instructions - Part 1



Classification Instructions

Thank you for participating in this experiment. In this section you find instructions as to how this experiment works. To take part in the experiment, we assume that you are familiar with the level-k model as it has been introduced by Nagel (1995) and also with the concept of team reasoning as it has been introduced by Schelling (1960). In the experiment, subjects play pure coordination games with symmetric and asymmetric payoffs. We assume that you are familiar with the concept of coordination games as they have been carried out by Crawford, Gneezy and Rottenstreich (2008).

However, in order to clarify potential questions of terminology, we reproduce the main features of the level-k model and the concept of team reasoning. In addition we provide detailed experimental instructions, which explain the game and also give you a short introduction to coordination games. Please read all information carefully in order to know how the original experiment proceeded.

Experimental Setting

Introduction

This section describes the main features of the experiment. Subjects are randomly assigned into teams of two players. For a given strategic situation, each player makes suggestions for the team action at two points in time. First, the so-called "suggested decision" and a justifying written message are exchanged between the team partners simultaneously. After this, the "final decision" is taken individually by each team player. The computer chooses randomly one of the two final decisions to obtain the "team's action."

All teams play a series of eight coordination games. Coordination games are characterised by situations in which all parties can realize mutual gains, but only by making mutually consistent decisions. Each team is randomly matched with another team. If a matched pair of teams both decide on identical team actions, they coordinate their behavior successfully and are rewarded with a payoff. However, if both teams choose different team actions, they fail to coordinate their behavior and do not receive any payoff. Thus both teams are motivated solely to coordinate their strategies in order to obtain an outcome that is best for them. The following example illustrates a random coordination game in which each team decides on one strategy X, Y or Z simultaneously. Only if both teams make mutually consistent decisions they receive a payoff of 2 units each.

		Team 2		
		X	Y	Z
	X	(2,2)	(0,0)	(0,0)
Team 1	Y	(0,0)	(2,2)	(0,0)
	Z	(0,0)	(0,0)	(2,2)

The payoff is represented through an experimental currency unit ("Taler"). One Taler is worth 0,40 Euro. In a symmetric coordination game each team is rewarded the same payoff if they coordinate their behavior successfully. In asymmetric coordination games players usually disagree on which action they prefer to coordinate. There may be one outcome where one team disproportionately benefits in comparison to the other team.

X-Y Coordination Games

All subjects face a series of eight coordination games composed of four "X-Y Games" and four "Pie Games". We reproduce the main features and attributes of those games in the following. "X-Y Games" are characterised by a binary choice option "X" or "Y". The assignment of payoffs for successful coordination is indicated in brackets. Example:

X [6 Taler for Team 1 and 5 Taler for Team 2]
Y [5 Taler for Team 1 and 6 Taler for Team 2]

If a matched pair of teams both decide on the identical team action "X", team one receives 6 Taler and team two receives 5 Taler. If both teams chose "Y", the assignment of payoffs would be reversed. If both teams chose decisions with different labels "X" and "Y", neither team receives any payoff. The payoff differences vary within the four "X-Y treatments".

Pie Coordination Games

"Pie Games" are characterised by a visual representation of different choice options as indicated in the following figure. Each team simultaneously selects one of the three "pie slices". Each slice is labeled with an abstract decision label §, $ or #. The assignment of payoffs for successful coordination is indicated in brackets within the three slices. The first number represents the quantity of Taler for team one, the second number the quantity of Taler for team two.

--Image of a Pie with payoffs-- *(omitted see van Elten and Penczynski (2020) page 47)*

If a matched pair of teams both decide on the identical team action "#", team one receives 7 Taler and team two receives 6 Taler. If both teams chose decisions with different labels §, $ or #, neither team receives any payoff. The payoff differences alternate within the four "Pie treatments".

Note that the "X-Y Game" and the "Pie Game" might both contain one alternative that is visually distinctive from another alternative. For instance, the unshaded bottom slice is visually distinctive from the two upper slices ($ and #) that are shaded in a light grey color. We refer to a visually distinctive alternative as label-salient. Moreover an alternative might be payoff-salient in a way that it is distinctive with respect to its payoff structure. The concept of label and payoff salience is important for the classification process.

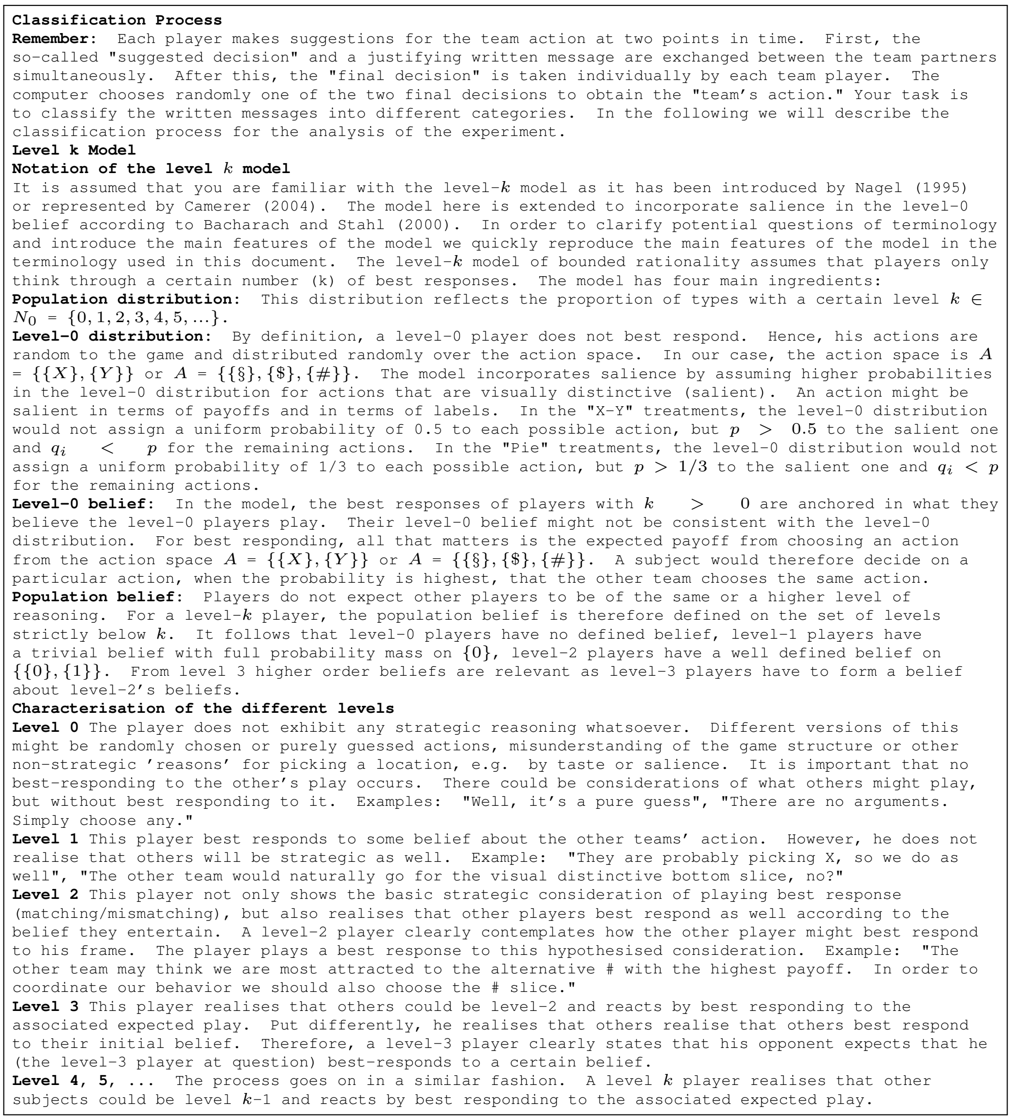
Treatment Overview

We conducted six sessions in Mannheim and three sessions Heidelberg. All sessions consist of the same eight treatments (four "X-Y games" and four "Pie games"), however the sequence of treatments in Mannheim is different from the sequence of treatments in Heidelberg. The following two tables provide a brief overview over the sessions conducted in Mannheim (session 1-3, session 7-9 [rounds 7 and 8 moved to the beginning]) and the sessions conducted in Heidelberg (session 4-6). The payoff for successful coordination is indicated in brackets. The first number represents the quantity of Taler for team one, the second number represents the quantity of Taler for team two, if both teams coordinate their behavior.

--Table presenting games played in each round in each experiment-- *(omitted see van Elten and Penczynski (2020) page 49)*

that can be generated by combining different levels of thinking with payoff or label salience, and with the two types of coordination games (X-Y and Pie Games). Furthermore, three out of the five examples that involve either payoff or label salience lack

Figure S17: L_{II} - Original Instructions - Part 2



Classification Process

Remember: Each player makes suggestions for the team action at two points in time. First, the so-called "suggested decision" and a justifying written message are exchanged between the team partners simultaneously. After this, the "final decision" is taken individually by each team player. The computer chooses randomly one of the two final decisions to obtain the "team's action." Your task is to classify the written messages into different categories. In the following we will describe the classification process for the analysis of the experiment.

Level k Model

Notation of the level k model

It is assumed that you are familiar with the level-k model as it has been introduced by Nagel (1995) or represented by Camerer (2004). The model here is extended to incorporate salience in the level-0 belief according to Bacharach and Stahl (2000). In order to clarify potential questions of terminology and introduce the main features of the model we quickly reproduce the main features of the model in the terminology used in this document. The level-k model of bounded rationality assumes that players only think through a certain number (k) of best responses. The model has four main ingredients:

Population distribution: This distribution reflects the proportion of types with a certain level $k \in N_0 = \{0, 1, 2, 3, 4, 5, ...\}$.

Level-0 distribution: By definition, a level-0 player does not best respond. Hence, his actions are random to the game and distributed randomly over the action space. In our case, the action space is $A = \{\{X\}, \{Y\}\}$ or $A = \{\{\S\}, \{\$\}, \{\#\}\}$. The model incorporates salience by assuming higher probabilities in the level-0 distribution for actions that are visually distinctive (salient). An action might be salient in terms of payoffs and in terms of labels. In the "X-Y" treatments, the level-0 distribution would not assign a uniform probability of 0.5 to each possible action, but $p > 0.5$ to the salient one and $q_i < p$ for the remaining actions. In the "Pie" treatments, the level-0 distribution would not assign a uniform probability of 1/3 to each possible action, but $p > 1/3$ to the salient one and $q_i < p$ for the remaining actions.

Level-0 belief: In the model, the best responses of players with $k > 0$ are anchored in what they believe the level-0 players play. Their level-0 belief might not be consistent with the level-0 distribution. For best responding, all that matters is the expected payoff from choosing an action from the action space $A = \{\{X\}, \{Y\}\}$ or $A = \{\{\S\}, \{\$\}, \{\#\}\}$. A subject would therefore decide on a particular action, when the probability is highest, that the other team chooses the same action.

Population belief: Players do not expect other players to be of the same or a higher level of reasoning. For a level-k player, the population belief is therefore defined on the set of levels strictly below k. It follows that level-0 players have no defined belief, level-1 players have a trivial belief with full probability mass on $\{0\}$, level-2 players have a well defined belief on $\{\{0\}, \{1\}\}$. From level 3 higher order beliefs are relevant as level-3 players have to form a belief about level-2's beliefs.

Characterisation of the different levels

Level 0 The player does not exhibit any strategic reasoning whatsoever. Different versions of this might be randomly chosen or purely guessed actions, misunderstanding of the game structure or other non-strategic 'reasons' for picking a location, e.g. by taste or salience. It is important that no best-responding to the other's play occurs. There could be considerations of what others might play, but without best responding to it. Examples: "Well, it's a pure guess", "There are no arguments. Simply choose any."

Level 1 This player best responds to some belief about the other teams' action. However, he does not realise that others will be strategic as well. Example: "They are probably picking X, so we do as well", "The other team would naturally go for the visual distinctive bottom slice, no?"

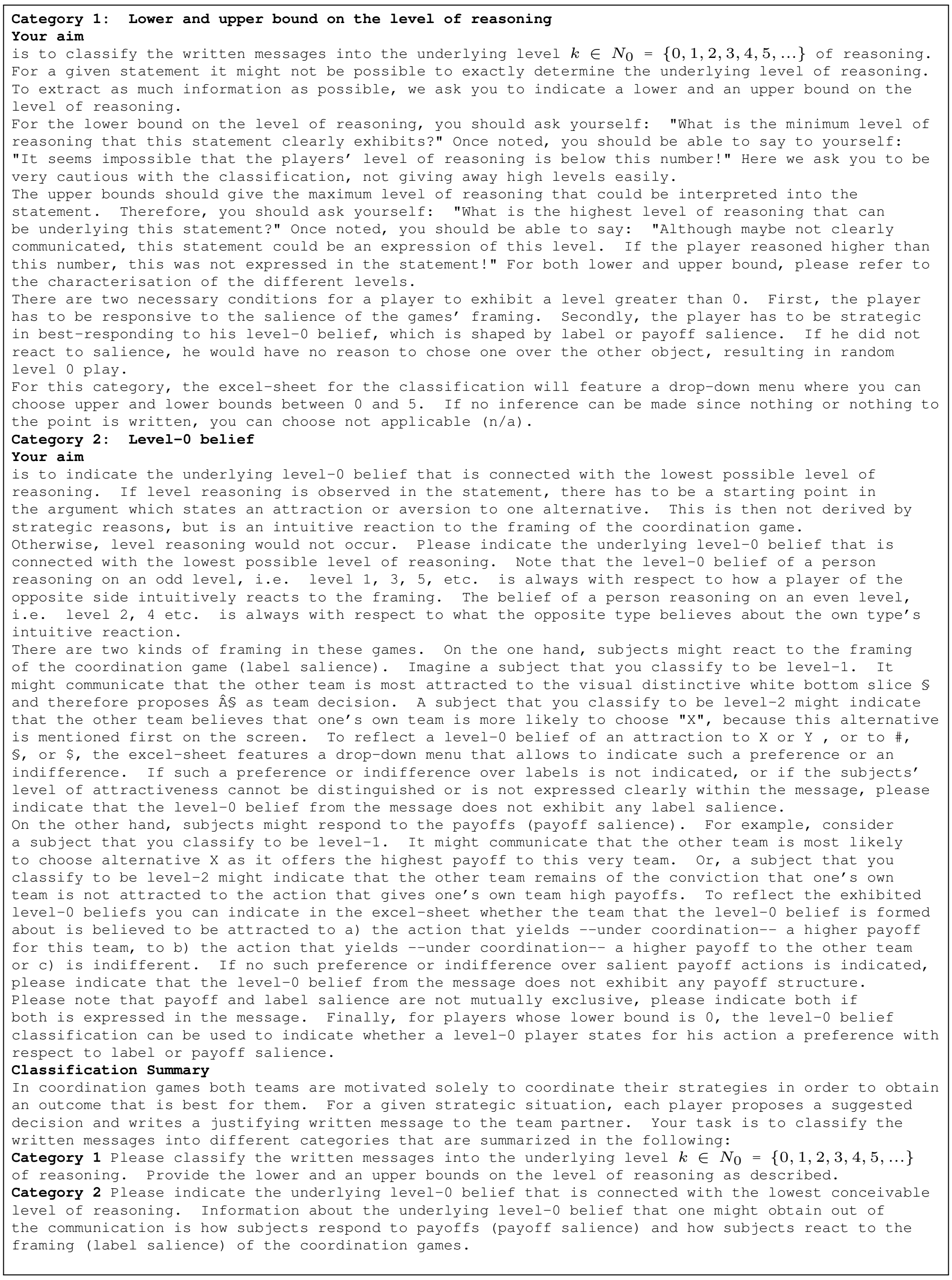
Level 2 This player not only shows the basic strategic consideration of playing best response (matching/mismatching), but also realises that other players best respond as well according to the belief they entertain. A level-2 player clearly contemplates how the other player might best respond to his frame. The player plays a best response to this hypothesised consideration. Example: "The other team may think we are most attracted to the alternative # with the highest payoff. In order to coordinate our behavior we should also choose the # slice."

Level 3 This player realises that others could be level-2 and reacts by best responding to the associated expected play. Put differently, he realises that others realise that others best respond to their initial belief. Therefore, a level-3 player clearly states that his opponent expects that he (the level-3 player at question) best-responds to a certain belief.

Level 4, 5, ... The process goes on in a similar fashion. A level k player realises that other subjects could be level k-1 and reacts by best responding to the associated expected play.

information indicating the appropriate classification for these categories. As a result, we, additionally, explored the possibility of providing a more balanced and larger set of examples with prompt O_+. This prompt is identical to prompt O except in its "Example" section, and is displayed in Figure S21. With this alternative "Example" section, we attempted to cover a broad array of possible case with a total of 16 demonstrations, four for each level from 0 to 3. We maintained the existing 5 examples from the original codebooks, and added additional examples to balance the categories. None of the new examples are from the dataset, and generated by us. All new examples are generated with the aim to explicitly depict their respective level of thinking. For instance, level-2 messages contain variations of the pattern "They think that I do <X>", and

Figure S18: L_{II} - Original Instructions - Part 3



Category 1: Lower and upper bound on the level of reasoning

Your aim

is to classify the written messages into the underlying level $k \in N_0 = \{0, 1, 2, 3, 4, 5, ...\}$ of reasoning. For a given statement it might not be possible to exactly determine the underlying level of reasoning. To extract as much information as possible, we ask you to indicate a lower and an upper bound on the level of reasoning.

For the lower bound on the level of reasoning, you should ask yourself: "What is the minimum level of reasoning that this statement clearly exhibits?" Once noted, you should be able to say to yourself: "It seems impossible that the players' level of reasoning is below this number!" Here we ask you to be very cautious with the classification, not giving away high levels easily.

The upper bounds should give the maximum level of reasoning that could be interpreted into the statement. Therefore, you should ask yourself: "What is the highest level of reasoning that can be underlying this statement?" Once noted, you should be able to say: "Although maybe not clearly communicated, this statement could be an expression of this level. If the player reasoned higher than this number, this was not expressed in the statement!" For both lower and upper bound, please refer to the characterisation of the different levels.

There are two necessary conditions for a player to exhibit a level greater than 0. First, the player has to be responsive to the salience of the games' framing. Secondly, the player has to be strategic in best-responding to his level-0 belief, which is shaped by label or payoff salience. If he did not react to salience, he would have no reason to chose one over the other object, resulting in random level 0 play.

For this category, the excel-sheet for the classification will feature a drop-down menu where you can choose upper and lower bounds between 0 and 5. If no inference can be made since nothing or nothing to the point is written, you can choose not applicable (n/a).

Category 2: Level-0 belief

Your aim

is to indicate the underlying level-0 belief that is connected with the lowest possible level of reasoning. If level reasoning is observed in the statement, there has to be a starting point in the argument which states an attraction or aversion to one alternative. This is then not derived by strategic reasons, but is an intuitive reaction to the framing of the coordination game.

Otherwise, level reasoning would not occur. Please indicate the underlying level-0 belief that is connected with the lowest possible level of reasoning. Note that the level-0 belief of a person reasoning on an odd level, i.e. level 1, 3, 5, etc. is always with respect to how a player of the opposite side intuitively reacts to the framing. The belief of a person reasoning on an even level, i.e. level 2, 4 etc. is always with respect to what the opposite type believes about the own type's intuitive reaction.

There are two kinds of framing in these games. On the one hand, subjects might react to the framing of the coordination game (label salience). Imagine a subject that you classify to be level-1. It might communicate that the other team is most attracted to the visual distinctive white bottom slice § and therefore proposes Â§ as team decision. A subject that you classify to be level-2 might indicate that the other team believes that one's own team is more likely to choose "X", because this alternative is mentioned first on the screen. To reflect a level-0 belief of an attraction to X or Y , or to #, §, or $, the excel-sheet features a drop-down menu that allows to indicate such a preference or an indifference. If such a preference or indifference over labels is not indicated, or if the subjects' level of attractiveness cannot be distinguished or is not expressed clearly within the message, please indicate that the level-0 belief from the message does not exhibit any label salience.

On the other hand, subjects might respond to the payoffs (payoff salience). For example, consider a subject that you classify to be level-1. It might communicate that the other team is most likely to choose alternative X as it offers the highest payoff to this very team. Or, a subject that you classify to be level-2 might indicate that the other team remains of the conviction that one's own team is not attracted to the action that gives one's own team high payoffs. To reflect the exhibited level-0 beliefs you can indicate in the excel-sheet whether the team that the level-0 belief is formed about is believed to be attracted to a) the action that yields --under coordination-- a higher payoff for this team, to b) the action that yields --under coordination-- a higher payoff to the other team or c) is indifferent. If no such preference or indifference over salient payoff actions is indicated, please indicate that the level-0 belief from the message does not exhibit any payoff structure.

Please note that payoff and label salience are not mutually exclusive, please indicate both if both is expressed in the message. Finally, for players whose lower bound is 0, the level-0 belief classification can be used to indicate whether a level-0 player states for his action a preference with respect to label or payoff salience.

Classification Summary

In coordination games both teams are motivated solely to coordinate their strategies in order to obtain an outcome that is best for them. For a given strategic situation, each player proposes a suggested decision and writes a justifying written message to the team partner. Your task is to classify the written messages into different categories that are summarized in the following:

Category 1 Please classify the written messages into the underlying level $k \in N_0 = \{0, 1, 2, 3, 4, 5, ...\}$ of reasoning. Provide the lower and an upper bounds on the level of reasoning as described.

Category 2 Please indicate the underlying level-0 belief that is connected with the lowest conceivable level of reasoning. Information about the underlying level-0 belief that one might obtain out of the communication is how subjects respond to payoffs (payoff salience) and how subjects react to the framing (label salience) of the coordination games.

level-3 messages explicitly state variations of "They think that I think that they will do <X>". In level-0 examples, one provides examples of label salience, one provides payoff salience, and two provides neither label nor payoff salience (two original examples). Level-1 examples consist of three cases of label salience and one case of payoff salience. For levels 2 and 3, two cases for each type of salience are provided. One of the level-0 example is explicitly for the X-Y game, and one for for the Pie game (the other two example are generic. Two of the level-1 examples are for the X-Y game, one for the Pie game, and one that can be applicable to either game. For levels 2 and 3, one example is for each game and two examples can be applicable to either game. Each example is followed by information on level-0 belief, label salience, payoff salience, and level classification.

C.5.4. Multi-label Classification

Given that the benchmark classifications for levels of thinking are defined as an interval, with a lower and an upper bound across six potential levels in any given instance (message), the model's task is a multi-class, multi-label classification. In this framework, a given instance can be classified as positive under multiple classes. This opens up the possibility for a classification to be partially correct. A measurement that requires an exact match with the actual interval might overlook this partial correctness and is therefore considered a harsh metric (Sorower, 2010). A more nuanced accuracy metric commonly used in multi-label classification is the ratio of correctly predicted labels to the total number of labels—both predicted and actual—averaged across all instances (Godbole and Sarawagi, 2004). This measurement is also referred to in the literature as the Intersection over Union (IoU) metric or the Jaccard Index (Müller et al., 2022). Although IoU can be calculated for every class and averaged (either macro or weighted), this approach has been criticised for not adequately addressing correlations among different classes. Therefore, to assess the classification performance of our models, we have chosen to employ the instance-based accuracy metric as introduced by Godbole and Sarawagi (2004).

C.5.5. Additional Results

To classify the level of strategic thinking, both human annotators and the model must first determine whether the message exhibits any form of salience (label or payoff), and if so, identify its type. Table S13 reports the accuracy of salience classification—both label and payoff—alongside the level classification results already discussed in the main text for comparison. GPT-3.5 performs poorly on salience detection, with most of the accuracy scores falling below 60%. GPT-4 consistently outperforms GPT-3.5 across all dimensions. For both models, the highest accuracy is typically achieved when n-shot prompting is combined with 0-shot-CoT prompting.

Table S14 reports the L_{II} results presented in the main text for reference in the rows labelled *0-shot* and *5-shot*. The row labelled *16-shot* corresponds to the O_{+} prompt,

Figure S19: L_{II} - Prompt O - Part 1

```
# General Task
- Evaluate player’s message from a specific coordination game to identify their decision-process and to
classify their level of strategic thinking.

# Role Persona
- Act as a behavioral economist specialized in coordination games, decision salience and text analysis.

# Context
## Game Mechanics
- Subjects (players) participate in an experiment where they play a coordination game.
- Players are assigned into teams of two.
- Each team is matched with another team to play a series of coordination games where the teams try to
coordinate on a specific alternative.
- If both teams pick the same alternative, each team is rewarded with a payoff.  Otherwise, neither
team receives any payoff.
- Each player sends a suggested decision and a justifying message to their teammate.
- Coordination game is not played within the team members but between the two teams.
- There is no communication between the teams.
- Payoffs are represented in currency called Taler.
- 1 Taler = 40 cents (ct).
- Each player plays a series of 8 (rounds of) coordination games, split between *X-Y* games and *Pie*
games.
## Coordination Games
- Payoff tables for each variation of each game is represented below.
- In each payoff table, ‘d‘ represents decision taken, ‘π1‘ represents payoff of team 1, and ‘π2‘
represents payoff of team 2.
- Payoff tables only represent the cases where both teams match in the given decision ‘d‘, if teams’
decisions do not match, each team receives 0 Taler.
### Pie Game
- Payoffs are displayed on a pie chart that is divided into three equally sized segments.
- Top left segment is labeled as ‘$‘.
- Top right segment is labeled as ‘#‘.
- Top segments are shaded in gray.
- Bottom segment is labeled as ‘§‘ and is highlighted in white.
- There are 4 payoff variations labeled as S1, S2, AM2 and AM4.
#### S1
| d     |π1, π2|
|-------|------ |
| L ($) | 5, 5  |
| R (#) | 5, 5  |
| B (§) | 5, 5  |
#### S2
| d     |π1, π2|
|-------|------ |
| L ($) | 6, 6  |
| R (#) | 6, 6  |
| B (§) | 5, 5  |
#### AM2
| d     |π1, π2|
|-------|------ |
| L ($) | 5, 6  |
| R (#) | 6, 5  |
| B (§) | 6, 5  |
#### AM4
| d     |π1, π2|
|-------|------ |
| L ($) | 6, 7  |
| R (#) | 7, 6  |
| B (§) | 7, 5  |
### X-Y Game
- Alternatives are displayed in two consecutive lines.
- Alternative ‘X‘ is displayed on the first line.
- Alternative ‘Y‘ is displayed on the second line.
- There are 4 payoff variations labeled as SL, ASL, AML and ALL.
#### SL
| d |π1, π2 |
|---|------- |
| X | 5, 5   |
| Y | 5, 5   |
#### ASL
| d |π1, π2 |
|---|------- |
| X | 5, 5.1 |
| Y | 5.1, 5 |
#### AML
| d |π1, π2 |
|---|------- |
| X | 5, 6   |
| Y | 6, 5   |
#### ALL
| d |π1, π2 |
|---|------- |
| X | 5, 10  |
| Y | 10, 5  |
```

Figure S20: L_{II} - Prompt O - Part 2

```
## Salience
- Label salience:  players may react to the framing of alternatives.
- Payoff salience:  Players may react to the payoff differences of alternatives .
## Level-k Model
- A level-k player performs k many iterative best responses and always starts its iterative reasoning
from his level-0 belief.  This starting point is called the level-0 belief of the level-k player.
- Level-0 belief is the belief of the level-k player on how the level-0 player will potentially play
the game.
- A level-0 player picks an alternative for non-strategic, instinctive reasons such as payoff or label
salience.
- A level-0 player does not best respond to other players potential actions.
- A level-0 player may be randomly choosing or purely guessing an action.
- A level-0 player may misunderstand the game structure.
- A level-1 player assumes that the other team consists of level-0 players and best responds based on
his level-0 belief to these level-0 players.
- A level-k (k>1) player recognizes the possibility that the other team may consist of level-(k-1)
players.
- A level-k player assumes that the level-(k-1) players assume he (the level-k player) is a level-(k-2)
player.
## Level-0 Belief
- Level-0 belief of a player reasoning on an odd level (level 1,3 or 5) is always with respect to how a
player of the opposite side intuitively reacts to the framing (label or payoff salience).
- Level-0 belief of a player reasoning on an even level (level 2 or 4) is always with respect to what
the opposite type believes about the own type’s intuitive reaction (label or payoff salience).
## Lower bound
- The minimum level of reasoning that the message clearly exhibits.
## Upper bound
- The maximum level of reasoning that can be inferred from the message.

# Classification Tasks
- Classify player’s label or payoff salience (if any).
- Classify lower and upper bounds for the player’s level of reasoning.

# Classification Coding
## Label Salience
### X-Y Game
- Code as ‘X` if player’s label salience is "prefers X over Y"
- Code as ‘Y` if player’s label salience is "prefers Y over X"
- Code as ‘∼` if player’s label salience is "indifferent across payoffs"
- Code as ‘no` if the player does not exhibit payoff salience.
### Pie Game
- Code as ‘$` if player’s label salience is "prefers $"
- Code as ‘#` if player’s label salience is "prefers #"
- Code as ‘§` if player’s label salience is "prefers §"
- Code as ‘∼` if player’s label salience is "indifferent across labels"
- Code as ‘no` if the player does not exhibit label salience.
## Payoff Salience
- Code as ‘H` if player’s payoff salience is "prefers high payoffs"
- Code as ‘L` if player’s payoff salience is "prefers low payoffs"
- Code as ‘∼` if player’s payoff salience is "indifferent across payoffs"
- Code as ‘no` if the player does not exhibit payoff salience.
## Upper and Lower Bounds
- 0,1,2,3,4 or 5.

# Examples (Only used in n-shot treatments)
- "Well, it’s a pure guess" (Level-0 belief:  none, Label Salience:  none, Payoff Salience:  none,
Level:  0)
- "There are no arguments.  Simply choose any." (Level-0 belief:  none, Label Salience:  none, Payoff
Salience:  none, Level:  0)
- "They are probably picking X, so we do as well" (Level-0 belief:  X, Label Salience:  X, Payoff
Salience:  none, Level:  1)
- "The other team would naturally go for the visually distinctive bottom slice" (Level-0 belief:  Â§,
Label Salience:  Â§, Payoff Salience:  none, Level:  1)
- "The other team may think we are most attracted to the alternative  with the highest payoff.  In
order to coordinate our behavior, we should also choose the  slice." (Level-0 belief:    - high payoff,
Label Salience:  none, Payoff Salience:  high payoff, Level:  2)

# Input Format
    Team:
    Game:
    Decision:
    Message:

# Constraint
- Follow the below output format

# Output Format
    Label Salience:
    Payoff Salience:
    Lower Bound:
    Upper Bound:
```

which includes 11 additional examples. As shown in the table, these additional examples substantially improved GPT-3.5’s performance. For GPT-4, the effect was mixed: accuracy decreased by 2 percentage points in the no-CoT condition but increased by 2

Figure S21: L_{II} - Prompt O_+ - Extended Examples

```
# Examples (Only used in n-shot treatments)
- "Well, it’s a pure guess" (Level-0 belief:  none, Label salience:  no, Payoff salience:  no, Level:
0)
- "There are no arguments.  Simply choose any." (Level-0 belief:  none, Label Salience:  none, Payoff
Salience:  none, Level:  0)
- "Â§ is highlighted in white, hence Â§" (Level-0 belief:  Â§, Label salience:  Â§, Payoff salience:
no, Level:  0)
- "Y provides a higher payoff, let’s go X" (Level-0 belief:  high payoff, Label salience:  no, Payoff
salience:  higher payoff, Level:  0)
- "They are probably picking X, so we do as we as well" (Level-0 belief:  X, Label salience:  X, Payoff
salience:  no, Level:  1)
- "The other team would naturally go for the visually distinctive bottom slice" (Level-0 belief:  Â§,
Label salience:  Â§, Payoff salience:  no, Considers other team’s behavior, Level:  1)
- "X is first, let’s pick X. And other team may think the same way." (Level-0 belief:  X, Label
salience:  X, Payoff salience:  no, Level:  1)
- "Other team may want to the high payoff for themselves, let’s coordinate with them and pick the
higher payoff for them (which is the lower payoff for us)." (Level-0 belief:  high payoff, Label
salience:  no, Payoff salience:  high payoff, Level:  1)
- "The other team may think we are most attracted to the alternative with the highest payoff.  In order
to coordinate our behavior, we should also choose the slice." (Level-0 belief:  high payoff, Label
salience:  no, Payoff salience:  high payoff, Level:  2)
- "Others will think we go for top.  So let’s go for top." (Level-0 belief:  X, Label salience:  X,
Payoff salience:  no, Level 2)
- "The other team will think we pick the highlighted segment.  So we should coordinate and pick Â§"
(Level-0 belief:  Â§, Label salience:  Â§, Payoff salience:  no, Level:  2)
- "Other team may want to the high payoff for themselves.  But they may assume the same thing about
us and pick the alternative that gives us the higher payoff.  So let’s coordinate with them and pick
the higher payoff for us (which is the lower payoff for them)." (Level-0 belief:  high payoff, Label
salience:  no, Payoff salience:  higher payoff, Level:  2)
- "Others will think that we think that they will go for top.  So let’s go for top." (Level-0 belief:
X, Label salience:  X, Payoff salience:  no, Level:  3)
- "The other team will think that we think that they pick the highlighted segment.  So we should
coordinate and pick Â§" (Level-0 belief:  Â§, Label salience:  Â§, Payoff salience:  no, Level:  3)
- "Other team may think that we think that they want the high payoff for themselves, let’s coordinate
with them and pick the higher payoff for them." (Level-0 belief:  high payoff, Label salience:  no,
Payoff salience:  higher payoff, Level:  3)
- "Other team may think that we want the high payoff for ourself.  But they may assume the same thing
about us and pick the alternative that gives us the higher payoff for themselves.  So let’s coordinate
with them and pick the higher payoff for them." (Level-0 belief:  higher payoff, Label salience:  no,
Payoff salience:  higher payoff, Level:  3)
```

		Payoff Salience	Label Salience	Level
GPT-3.5	∅	36	42	52
	CoT	49	59	55
	n-shot	**59**	46	**58**
	CoT & n-shot	54	**64**	**58**
GPT-4	∅	59	44	66
	CoT	63	79	67
	n-shot	60	78	**71**
	CoT & n-shot	**70**	**85**	**71**

Table S13: Salience Classification Accuracy (in %), highest values in bold.

percentage points when CoT prompting was used.

Table S15 reports salience classification accuracy for prompt O_+, which includes 11 additional examples (16-shot total) in both the no-CoT and CoT conditions. As shown in the table, these additional examples substantially improve classification accuracy for both types of salience under GPT-3.5. In contrast, the effect on GPT-4 is limited: with the exception of an 11-percentage-point gain in payoff salience accuracy under the no-CoT condition, the additional examples produce either no change or only minor,

		no-CoT	**CoT**
GPT-3.5	0-shot	52	55
	5-shot	58	58
	16-shot	**63**	**65**
GPT-4	0-shot	66	67
	5-shot	**71**	71
	16-shot	69	**73**

Table S14: Level Classification Accuracy (in %), highest values in bold.

mixed effects.

		Payoff Salience	Label Salience
GPT-3.5	no-CoT	73 +14	47 +1
	CoT	66 +12	68 +4
GPT-4	no-CoT	71 +11	78 0
	CoT	69 -1	86 +1

Table S15: 16-shot - Salience Classification Accuracy (in %) and change in accuracy (in percentage points). Change in accuracy compared to 5-shot (in percentage points).

References

Aiyappa, Rachith, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn, "Can we trust the evaluation on ChatGPT?," *arXiv preprint arXiv:2303.12767*, 2023.

Alonso-Robisco, Andres and José Manuel Carbó, "Analysis of CBDC narrative by central banks using large language models," *Finance Research Letters*, 2023, *58*, 104643.

Amin, Mostafa M, Erik Cambria, and Björn W Schuller, "Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of chatgpt," *IEEE Intelligent Systems*, 2023, *38* (2), 15–23.

Anthropic, "Mitigating Jailbreaks and Prompt Injections," 2023. Accessed: 2023-04-25.

Arad, Ayala, David Hugh-Jones, and Stefan P. Penczynski, "Communication is informative about cooperation," Technical Report, School of Economics, University of East Anglia, Norwich, UK. 2024.

Bach, Stephen H, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry et al., "Promptsource: An integrated development environment and repository for natural language prompts," *arXiv preprint arXiv:2202.01279*, 2022.

Baktash, Jawid Ahmad and Mursal Dawodi, "Gpt-4: A review on advancements and opportunities in natural language processing," *arXiv preprint arXiv:2305.03195*, 2023.

Bansal, Gagan, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld, "Does the whole exceed its parts? the effect of ai explanations on complementary team performance," in "Proceedings of the 2021 CHI conference on human factors in computing systems" 2021, pp. 1–16.

Beltagy, Iz, Matthew E Peters, and Arman Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

Bender, Emily M, "On achieving and evaluating language-independence in NLP," *Linguistic Issues in Language Technology*, 2011, *6*.

Bhat, Savita and Vasudeva Varma, "Large Language Models As Annotators: A Preliminary Evaluation For Annotating Low-Resource Language Content," in "Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems" 2023, pp. 100–107.

Bjerva, Johannes, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and

Isabelle Augenstein, “What do language representations really represent?,” *Computational Linguistics*, 2019, *45* (2), 381–389.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al., “Language models are few-shot learners,” *Advances in neural information processing systems*, 2020, *33*, 1877–1901.

Burchardi, Konrad B. and Stefan P. Penczynski, “Out of your mind: Eliciting individual reasoning in one shot games,” *Games and Economic Behavior*, 2014, *84* (1), 39 – 57.

Camerer, Colin F., Teck-Hua Ho, and Juin-Kuan Chong, “A Cognitive Hierarchy Model of Games,” *The Quarterly Journal of Economics*, August 2004, *119* (3), 861–898.

Capra, C Mónica, “Understanding decision processes in guessing games: a protocol analysis approach,” *Journal of the Economic Science Association*, 2019, *5* (1), 123–135.

Çelebi, Can and Stefan P. Penczynski, “Strategic Thinking in Jury Decisions: An Experimental Study,” Technical Report, School of Economics, University of East Anglia, Norwich, UK. 2023.

__ **and** __ , “Team communication and individuals’ reasoning and decisions,” in “Encyclopedia of Experimental Social Science” Elgar Encyclopedias in the Social Sciences, Edward Elgar Publishing, 2025.

Chae, Youngjin and Thomas Davidson, “Large language models for text classification: From zero-shot learning to fine-tuning,” *Open Science Foundation*, 2023.

Chang, Ting-Yun and Robin Jia, “Data curation alone can stabilize in-context learning,” *arXiv preprint arXiv:2212.10378*, 2022.

Charness, Gary and Martin Dufwenberg, “Promises and partnership,” *Econometrica*, 2006, *74* (6), 1579–1601.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann et al., “Palm: Scaling language modeling with pathways,” *Journal of Machine Learning Research*, 2023, *24* (240), 1–113.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma et al., “Scaling instruction-finetuned language models,” *arXiv preprint arXiv:2210.11416*, 2022.

Clavié, Benjamin, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell, “Large language models in the workplace: A case study on prompt engineering for job type classification,” in “International Conference on Applications of Natural Language to Information Systems” Springer 2023, pp. 3–17.

Conneau, Alexis and Guillaume Lample, “Cross-lingual language model pretraining,” *Advances in neural information processing systems*, 2019, *32.*

Cooper, David J, Ian Krajbich, and Charles N Noussair, “Choice-process data in experimental economics,” 2019.

Costa-Gomes, Miguel A. and Vincent P. Crawford, “Cognition and Behavior in Two-Person Guessing Games: An Experimental Study,” *American Economic Review*, December 2006, *96* (5), 1737–1768.

Crawford, Vincent P., Miguel A. Costa-Gomes, and Nagore Iriberri, “Structural Models of Nonequilibrium Strategic Thinking: Theory, Evidence, and Applications,” *Journal of Economic Literature*, September 2013, *51* (1), 5–62.

__ , Uri Gneezy, and Yuval Rottenstreich, “The Power of Focal Points Is Limited: Even Minute Payoff Asymmetry May Yields Large Coordination Failures,” *American Economic Review*, July 2008, *98* (4), 1443–1458.

den Assem, Martijn J Van, Dennie Van Dolder, and Richard H Thaler, “Split or steal? Cooperative behavior when the stakes are large,” *Management Science*, 2012, *58* (1), 2–20.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui, “A survey on in-context learning,” *arXiv preprint arXiv:2301.00234*, 2022.

Efrat, Avia and Omer Levy, “The turking test: Can language models understand instructions?,” *arXiv preprint arXiv:2010.11982*, 2020.

Fanta, Nicolas and Roman Horvath, “Artificial intelligence and central bank communication: the case of the ECB,” *Applied Economics Letters*, 2024, pp. 1–8.

Feddersen, Timothy and Wolfgang Pesendorfer, “Convicting the innocent: The inferiority of unanimous jury verdicts under strategic voting,” *American Political Science Review*, 1998, pp. 23–35.

Fu, Yao, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot, “Complexity-based prompting for multi-step reasoning,” in “The Eleventh

International Conference on Learning Representations” 2022.

Fyffe, Shea, Philseok Lee, and Seth Kaplan, “âTransformingâ personality scale development: Illustrating the potential of state-of-the-art natural language processing,” *Organizational Research Methods*, 2024, *27* (2), 265–300.

Gekhman, Zorik, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig, “Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?,” *arXiv preprint arXiv:2405.05904*, 2024.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy, “Text as data,” *Journal of Economic Literature*, 2019, *57* (3), 535–574.

Georgalos, Konstantinos and John Hey, “Testing for the emergence of spontaneous order,” *Experimental Economics*, 2020, *23* (3), 912–932.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli, “ChatGPT outperforms crowd workers for text-annotation tasks,” *Proceedings of the National Academy of Sciences*, 2023, *120* (30), e2305016120.

Glasserman, Paul and Caden Lin, “Assessing Look-Ahead Bias in Stock Return Predictions Generated By GPT Sentiment Analysis,” *arXiv preprint arXiv:2309.17322*, 2023.

Glavaš, Goran, Federico Nanni, and Simone Paolo Ponzetto, “Computational analysis of political texts: bridging research efforts across communities,” in “Proceedings of the 57th annual meeting of the association for computational linguistics: Tutorial abstracts” 2019, pp. 18–23.

Godbole, Shantanu and Sunita Sarawagi, “Discriminative methods for multi-labeled classification,” in “Pacific-Asia conference on knowledge discovery and data mining” Springer 2004, pp. 22–30.

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

Gu, Jiasheng, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin, “Robustness of learning from task instructions,” *arXiv preprint arXiv:2212.03813*, 2022.

Guarnaschelli, Serena, Richard D McKelvey, and Thomas R Palfrey, “An experimental study of jury decision rules,” *American Political Science Review*, 2000, pp. 407–423.

Halterman, Andrew and Katherine A Keith, “Codebook LLMs: Evaluating LLMs as Measurement Tools for Political Science Concepts,” *arXiv preprint arXiv:2407.10747*, 2024.

Hansen, Anne Lundgaard and Sophia Kazinnik, "Can chatgpt decipher fedspeak," *Available at SSRN*, 2023.

Haviv, Adi, Jonathan Berant, and Amir Globerson, "BERTese: Learning to speak to BERT," *arXiv preprint arXiv:2103.05327*, 2021.

He, Zeyu, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao'Kenneth' Huang, "If in a Crowdsourced Data Annotation Pipeline, a GPT-4," *arXiv preprint arXiv:2402.16795*, 2024.

Heseltine, Michael and Bernhard Clemm von Hohenberg, "Large language models as a substitute for human experts in annotating political text," *Research & Politics*, 2024, *11* (1), 20531680241236239.

Houser, Daniel and Erte Xiao, "Classification of natural language messages using a coordination game," *Experimental Economics*, 2011, *14*, 1–14.

Huang, Allen H, Hui Wang, and Yi Yang, "FinBERT: A large language model for extracting information from financial text," *Contemporary Accounting Research*, 2023, *40* (2), 806–841.

Huang, Jie and Kevin Chen-Chuan Chang, "Towards reasoning in large language models: A survey," *arXiv preprint arXiv:2212.10403*, 2022.

Huh, Minyoung, Brian Cheung, Tongzhou Wang, and Phillip Isola, "The Platonic Representation Hypothesis," *arXiv preprint arXiv:2405.07987*, 2024.

Hüning, Hendrik, Lydia Mechtenberg, and Stephanie Wang, "Detecting arguments and their positions in experimental communication data," *Available at SSRN 4052402*, 2022.

__ , __ , and __ , "Using Arguments to Persuade: Experimental Evidence," *Available at SSRN 4244989*, 2022.

IV, Robert L Logan, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel, "Cutting down on prompts and parameters: Simple few-shot learning with language models," *arXiv preprint arXiv:2106.13353*, 2021.

Jha, Manish, Jialin Qian, Michael Weber, and Baozhong Yang, "ChatGPT and corporate policies," Technical Report, National Bureau of Economic Research 2024.

Jiang, Zhengbao, Frank F Xu, Jun Araki, and Graham Neubig, "How can we know what language models know?," *Transactions of the Association for Computational Linguistics*, 2020, *8*, 423–438.

Jin, Mingyu, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du, "The impact of reasoning step length on large language models," *arXiv preprint arXiv:2401.04925*, 2024.

Kaddour, Jean, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy, "Challenges and applications of large language models," *arXiv preprint arXiv:2307.10169*, 2023.

Khot, Tushar, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," *arXiv preprint arXiv:2210.02406*, 2022.

Kim, Alex, Maximilian Muhn, and Valeri Nikolaev, "From Transcripts to Insights: Uncovering Corporate Risks Using Generative AI," *arXiv preprint arXiv:2310.17721*, 2023.

Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, 2022, *35*, 22199–22213.

Kong, Aobo, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou, "Better zero-shot reasoning with role-play prompting," *arXiv preprint arXiv:2308.07702*, 2023.

Kosinski, Michal, "Evaluating large language models in theory of mind tasks," *Proceedings of the National Academy of Sciences*, 2024, *121* (45), e2405460121.

Kostina, Arina, Marios D Dikaiakos, Dimosthenis Stefanidis, and George Pallis, "Large Language Models For Text Classification: Case Study And Comprehensive Review," *arXiv preprint arXiv:2501.08457*, 2025.

Kumar, Sawan and Partha Talukdar, "Reordering examples helps during priming-based few-shot learning," *arXiv preprint arXiv:2106.01751*, 2021.

__ **and** __ , "Reordering examples helps during priming-based few-shot learning," *arXiv preprint arXiv:2106.01751*, 2021.

Kuzman, Taja, Igor Mozetic, and Nikola Ljubešic, "ChatGPT: beginning of an end of manual linguistic data annotation," *Use case of automatic genre identification*, 2023.

Kuznia, Kirby, Swaroop Mishra, Mihir Parmar, and Chitta Baral, "Less is more: Summary of long instructions is better for program synthesis," *arXiv preprint arXiv:2203.08597*, 2022.

Lai, Viet Dac, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen, "Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning," *arXiv preprint arXiv:2304.05613*, 2023.

Lampinen, Andrew Kyle, Stephanie CY Chan, Aaditya K Singh, and Murray Shanahan, "The broader spectrum of in-context learning," *arXiv preprint* *arXiv:2412.03782*, 2024.

Lester, Brian, Rami Al-Rfou, and Noah Constant, “The power of scale for parameter-efficient prompt tuning,” *arXiv preprint arXiv:2104.08691*, 2021.

Li, Lingyao, Lizhou Fan, Shubham Atreja, and Libby Hemphill, “âHOTâ ChatGPT: The promise of ChatGPT in detecting and discriminating hateful, offensive, and toxic comments on social media,” *ACM Transactions on the Web*, 2024, *18* (2), 1–36.

Li, Xiang Lisa and Percy Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *arXiv preprint arXiv:2101.00190*, 2021.

Li, Xiaonan and Xipeng Qiu, “Finding support examples for in-context learning,” *arXiv preprint arXiv:2302.13539*, 2023.

__ and __ , “Finding support examples for in-context learning,” *arXiv preprint arXiv:2302.13539*, 2023.

Liu, Jiachang, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen, “What Makes Good In-Context Examples for GPT-3?,” *arXiv preprint arXiv:2101.06804*, 2021.

Liu, Nelson F, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang, “Lost in the middle: How language models use long contexts,” *Transactions of the Association for Computational Linguistics*, 2024, *12*, 157–173.

Liu, Peng and Zhizhong Li, “Task complexity: A review and conceptualization framework,” *International Journal of Industrial Ergonomics*, 2012, *42* (6), 553–568.

Liu, Xiping and Zhao Tan, “Divide and prompt: Chain of thought prompting for text-to-sql,” *arXiv preprint arXiv:2304.11556*, 2023.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

Lopez-Lira, Alejandro and Yuehua Tang, “Can chatgpt forecast stock price movements? return predictability and large language models,” *arXiv preprint arXiv:2304.07619*, 2023.

Lou, Renze and Wenpeng Yin, “Toward zero-shot instruction following,” in “Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop” 2024, pp. 50–60.

__ , Kai Zhang, and Wenpeng Yin, “Is prompt all you need? no. a comprehensive and broader view of instruction learning," *arXiv preprint arXiv:2303.10475*, 2023.

Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," *arXiv preprint arXiv:2104.08786*, 2021.

__ , __ , __ , __ , and __ , "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," *arXiv preprint arXiv:2104.08786*, 2021.

Luo, Man, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao, "Dr. icl: Demonstration-retrieved in-context learning," *arXiv preprint arXiv:2305.14128*, 2023.

MacCallum, Noah and Julian Lee, "GPTâ4.1 Prompting Guide," `https://cookbook.openai.com/examples/gpt4-1_prompting_guide` April 2025. Accessed: 2025â10â20.

Madaan, Aman and Amir Yazdanbakhsh, "Text and patterns: For effective chain of thought, it takes two to tango," *arXiv preprint arXiv:2209.07686*, 2022.

Matter, Daniel, Miriam Schirmer, Nir Grinberg, and Jürgen Pfeffer, "Close to Human-Level Agreement: Tracing Journeys of Violent Speech in Incel Posts with GPT-4-Enhanced Annotations," *arXiv preprint arXiv:2401.02001*, 2024.

Min, Sewon, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?," *arXiv preprint arXiv:2202.12837*, 2022.

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," *arXiv preprint arXiv:2104.08773*, 2021.

__ , __ , __ , Yejin Choi, and Hannaneh Hajishirzi, "Reframing Instructional Prompts to GPTk's Language," *arXiv preprint arXiv:2109.07830*, 2021.

Müller, Dominik, Iñaki Soto-Rey, and Frank Kramer, "Towards a guideline for evaluation metrics in medical image segmentation," *BMC Research Notes*, 2022, *15* (1), 210.

Nelson, Laura K, Derek Burk, Marcel Knudsen, and Leslie McCall, "The future of coding: A comparison of hand-coding and three types of computer-assisted text analysis methods," *Sociological Methods & Research*, 2021, *50* (1), 202–237.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira et al., "Universal dependencies v1: A multilingual treebank collection," in "Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)" 2016, pp. 1659–1666.

Obaid, Khaled and Kuntara Pukthuanthong, "Distortions in Financial Narratives: A ChatGPT Approach," 2024.

OpenAI, "Fine-tuned Classification," https://github.com/openai/openai-cookbook/blob/main/examples/Fine-tuned_classification.ipynb 2023. Accessed: 2023-04-20.

__ , "Prompt design," https://platform.openai.com/docs/guides/completion/prompt-design 2023.

__ , "Using Logprobs," https://cookbook.openai.com/examples/using_logprobs 2023. Accessed: 2023-04-21.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray et al., "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, 2022, *35*, 27730–27744.

Palmer, Alexis, Noah A Smith, and Arthur Spirling, "Using proprietary language models in academic research requires explicit justification," *Nature Computational Science*, 2024, *4* (1), 2–3.

Pan, Jane, Tianyu Gao, Howard Chen, and Danqi Chen, "What In-Context Learning" Learns" In-Context: Disentangling Task Recognition and Task Learning," *arXiv preprint arXiv:2305.09731*, 2023.

Pangakis, Nicholas, Samuel Wolken, and Neil Fasching, "Automated annotation with generative ai requires validation," *arXiv preprint arXiv:2306.00176*, 2023.

Patel, Pruthvi, Swaroop Mishra, Mihir Parmar, and Chitta Baral, "Is a question decomposition unit all we need?," *arXiv preprint arXiv:2205.12538*, 2022.

Penczynski, Stefan P, "Using machine learning for communication classification," *Experimental Economics*, 2019, *22* (4), 1002–1029.

Peskine, Youri, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso, "Definitions Matter: Guiding GPT for Multi-label Classification," in "Findings of the Association for Computational Linguistics: EMNLP 2023" 2023, pp. 4054–4063.

Peskoff, Denis, Adam Visokay, Sander Schulhoff, Benjamin Wachspress, Alan Blinder, and Brandon M Stewart, "GPT Deciphering Fedspeak: Quantifying Dissent Among Hawks and Doves," in "Findings of the Association for

Computational Linguistics: EMNLP 2023" 2023, pp. 6529–6539.

Prabhakar, Akshara, Thomas L Griffiths, and R Thomas McCoy, "Deciphering the factors influencing the efficacy of chain-of-thought: Probability, memorization, and noisy reasoning," *arXiv preprint arXiv:2407.01687*, 2024.

Puri, Ravsehaj Singh, Swaroop Mishra, Mihir Parmar, and Chitta Baral, "How many data samples is an additional instruction worth?," *arXiv preprint arXiv:2203.09161*, 2022.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI Blog*, 2019.

Rathje, Steve, Dan-Mircea Mirea, Ilia Sucholutsky, Raja Marjieh, Claire Robertson, and Jay J Van Bavel, "GPT is an effective tool for multilingual psychological text analysis," 2023.

Razeghi, Yasaman, Robert L Logan IV, Matt Gardner, and Sameer Singh, "Impact of pretraining term frequencies on few-shot reasoning," *arXiv preprint arXiv:2202.07206*, 2022.

Reiss, Michael V, "Testing the reliability of chatgpt for text annotation and classification: A cautionary remark," *arXiv preprint arXiv:2304.11085*, 2023.

Reynolds, Laria and Kyle McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," in "Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems" 2021, pp. 1–7.

Rytting, Christopher Michael, Taylor Sorensen, Lisa Argyle, Ethan Busby, Nancy Fulda, Joshua Gubler, and David Wingate, "Towards coding social science datasets with language models," *arXiv preprint arXiv:2306.02177*, 2023.

Salewski, Leonard, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata, "In-Context Impersonation Reveals Large Language Models' Strengths and Biases," *Advances in Neural Information Processing Systems*, 2024, *36.*

Savelka, Jaromir, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu, "Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise?," *arXiv preprint arXiv:2306.13906*, 2023.

Scao, Teven Le and Alexander M Rush, "How many data points is a prompt worth?," *arXiv preprint arXiv:2103.08493*, 2021.

Schick, Timo and Hinrich Schütze, "It's not just size that matters: Small language models are also few-shot learners," *arXiv preprint arXiv:2009.07118*, 2020.

Schmiedel, Theresa, Oliver Müller, and Jan Vom Brocke, "Topic modeling as a strategy of inquiry in organizational research: A tutorial with an application example on organizational culture," *Organizational Research Methods*, 2019, *22* (4), 941–968.

Schotter, Andrew, "Decision making with naive advice," *American Economic Review*, 2003, *93* (2), 196–201.

Sclar, Melanie, Yejin Choi, Yulia Tsvetkov, and Alane Suhr, "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting," *arXiv preprint arXiv:2310.11324*, 2023.

Shen, Xinyue, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang, "" do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," *arXiv preprint arXiv:2308.03825*, 2023.

Smales, Lee A, "Classification of RBA monetary policy announcements using ChatGPT," *Finance Research Letters*, 2023, *58*, 104514.

Sorower, Mohammad S, "A literature survey on algorithms for multi-label learning," *Oregon State University, Corvallis*, 2010, *18* (1), 25.

Spirling, Arthur, "Why open-source generative AI models are an ethical way forward for science," *Nature*, 2023, *616* (7957), 413–413.

Sprague, Zayne, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett, "To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning," *arXiv preprint arXiv:2409.12183*, 2024.

Srivastava, Saksham Sahai and Ashutosh Gandhi, "MathDivide: Improved mathematical reasoning by large language models," *arXiv preprint arXiv:2405.13004*, 2024.

Su, Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith et al., "Selective annotation makes language models better few-shot learners," *arXiv preprint arXiv:2209.01975*, 2022.

Törnberg, Petter, "Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning," *arXiv preprint arXiv:2304.06588*, 2023.

Vaccaro, Michelle, Abdullah Almaatouq, and Thomas Malone, "When combinations of humans and AI are useful: A systematic review and meta-analysis," *Nature Human Behaviour*, 2024, pp. 1–11.

van Elten, Jonas and Stefan P Penczynski, "Coordination games with asymmetric payoffs: An experimental study with intra-group communication," *Journal of Economic Behavior & Organization*, 2020, *169*, 158–188.

Vossen, Rainer and Gerrit J Dimmendaal, *The Oxford handbook of African languages*, Oxford University Press, USA, 2020.

Wang, Lei, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," *arXiv preprint arXiv:2305.04091*, 2023.

Wang, Shuohang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng, "Want to reduce labeling cost? GPT-3 can help," *arXiv preprint arXiv:2108.13487*, 2021.

Wang, Xiaolei, Xinyu Tang, Wayne Xin Zhao, and Ji-Rong Wen, "Investigating the Pre-Training Dynamics of In-Context Learning: Task Recognition vs. Task Learning," *arXiv preprint arXiv:2406.14022*, 2024.

Wang, Xinru, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao, "Human-llm collaborative annotation through effective verification of llm labels," in "Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems" 2024, pp. 1–21.

Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

Webson, Albert and Ellie Pavlick, "Do prompt-based models really understand the meaning of their prompts?," *arXiv preprint arXiv:2109.01247*, 2021.

Wei, Jason, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.

__ , Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, 2022, *35*, 24824–24837.

__ , Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler et al., "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, 2022.

White, Jules, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

Wu, Zhaofeng, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim, "Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks," *arXiv preprint arXiv:2307.02477*, 2023.

Yang, Jie, Judith Redi, Gianluca Demartini, and Alessandro Bozzon, "Modeling task complexity in crowdsourcing," in "Proceedings of the AAAI Conference on Human Computation and Crowdsourcing," Vol. 4 2016, pp. 249–258.

Ye, Junjie, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen et al., "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," *arXiv preprint arXiv:2303.10420*, 2023.

Yoo, Kang Min, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang goo Lee, and Taeuk Kim, "Ground-truth labels matter: A deeper look into input-label demonstrations," *arXiv preprint arXiv:2205.12685*, 2022.

Yuan, J, P Bao, Z Chen, M Yuan, J Zhao, J Pan, Y Xie, Y Cao, Y Wang, Z Wang et al., "Advanced prompting as a catalyst: Empowering large language models in the management of gastrointestinal cancers," *The Innovation*, 2023.

Yuan, Weizhe, Graham Neubig, and Pengfei Liu, "Bartscore: Evaluating generated text as text generation," *Advances in Neural Information Processing Systems*, 2021, *34*, 27263–27277.

Zhang, Bowen, Daijun Ding, and Liwen Jing, "How would stance detection techniques evolve after the launch of chatgpt?," *arXiv preprint arXiv:2212.14548*, 2022.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin et al., "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong et al., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

Zhao, Zihao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh, "Calibrate before use: Improving few-shot performance of language models," in

"International conference on machine learning" PMLR 2021, pp. 12697–12706.

Zheng, Huaixiu Steven, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou, "Take a step back: Evoking reasoning via abstraction in large language models," *arXiv preprint arXiv:2310.06117*, 2023.

__ , __ , __ , __ , __ , __ , and __ , "Take a step back: Evoking reasoning via abstraction in large language models," *arXiv preprint arXiv:2310.06117*, 2023.

Zhong, Qihuang, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao, "Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert," *arXiv preprint arXiv:2302.10198*, 2023.

Zhu, Yiming, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson, "Can chatgpt reproduce human-generated labels? a study of social computing tasks," *arXiv preprint arXiv:2304.10145*, 2023.

Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang, "Can large language models transform computational social science?," *Computational Linguistics*, 2024, pp. 1–55.