Text Annotation via Inductive Coding: Comparing Human Experts to LLMs in Qualitative Data Analysis

Angelina Parfenova[1,2], **Andreas Marfurt**[1], **Alexander Denzler**[1], and **Juergen Pfeffer**[2]

[1]Lucerne University of Applied Sciences and Arts
[2]Technical University of Munich

Abstract

This paper investigates the automation of qualitative data analysis, focusing on inductive coding using large language models (LLMs). Unlike traditional approaches that rely on deductive methods with predefined labels, this research investigates the inductive process where labels emerge from the data. The study evaluates the performance of six open-source LLMs compared to human experts. As part of the evaluation, experts rated the perceived difficulty of the quotes they coded. The results reveal a peculiar dichotomy: human coders consistently perform well when labeling complex sentences but struggle with simpler ones, while LLMs exhibit the opposite trend. Additionally, the study explores systematic deviations in both human and LLM generated labels by comparing them to the golden standard from the test set. While human annotations may sometimes differ from the golden standard, they are often rated more favorably by other humans. In contrast, some LLMs demonstrate closer alignment with the true labels but receive lower evaluations from experts.

1 Introduction

Qualitative data analysis (QDA) is an important research method across various fields such as marketing, media studies, social science, psychology, medical research, and others (Avjyan, 2005; Creswell, 2016; Mohajan et al., 2018; Flick, 2018; Leeson et al., 2019; Brennen, 2021). Unlike quantitative research, which relies on numerical data and statistical analysis, qualitative research captures the richness and complexity of human experiences, behaviors, and social phenomena (Denzin and Lincoln, 2005; Patton, 2014). It explores research questions in more details, providing insights that are often missed by quantitative methods. Yet, this depth of understanding comes at a cost—QDA is naturally labor-intensive, requiring thorough manual work that are both time-consuming and sensitive to inconsistencies and subjective biases (Morse, 2015; Bumbuc, 2016).

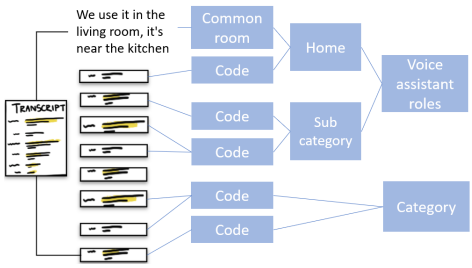


Figure 1: Coding in *thematic analysis*. The source text is split into quotes. The main idea of a paragraph is extracted and becomes a *code* (open coding). Then, this list of codes is hierarchically grouped into more abstract categories (axial coding).

One of the most critical and demanding stages of QDA, specifically of *thematic analysis*, is the process of *coding*. Coding involves the systematic identification and labeling of significant themes, ideas, attitudes, and topics within a body of text (Charmaz, 2014; Glaser and Strauss, 2017). This method consists of two stages: open coding and axial coding (see Figure 1) which aim to summarize the main ideas from sentences into *codes* and then categorize them, establishing hierarchy (Saldana, 2016). Despite its importance, coding is time-consuming, often taking weeks to complete for large datasets (Alshenqeeti, 2014; Hennink et al., 2020). Moreover, the manual nature of this process makes it prone to subjective interpretation, which can lead to variability in the results (Ryan and Bernard, 2003; MacQueen et al., 2008).

Automating QDA is increasingly important, as traditional methods like Topic Modeling and Wordnet hierarchies capture keywords but often miss deeper insights (Leeson et al., 2019; Parfenova, 2024). Advances in NLP, especially LLMs, offer potential for reducing manual effort in coding, though their ability to match human analysis remains uncertain. This paper explores how LLMs

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 6471–6484

can automate the open coding process, comparing their performance to human experts through experiments in zero-shot, few-shot, and fine-tuning scenarios. A key finding of this study is that fine-tuning with as few as 100 examples can achieve sufficient performance, which is particularly beneficial for computational social science research, where data collection remains a challenge (Lazer et al., 2020).

2 Related Work

Qualitative coding involves the systematic categorization of textual data to identify patterns, themes, and insights. In this process, each significant statement or segment of text is assigned a *code* that encapsulates its core idea. According to the definition by Saldana (2016), a code is "often a word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based or visual data." In one of the most popular methods of QDA, *thematic analysis*, once these segments are coded, they are grouped into broader categories that highlight underlying hierarchy between codes. The data itself can consist of interviews, documents, field notes, or any other source of qualitative information. To explain the process more simply, we first summarize the main idea of each quote (sentence or paragraph). Then, we group these summaries into larger categories. This involves examining all the ideas we've identified and determining how they fit together into broader themes.

Methods similar to coding One of the most extensively studied approaches is the use of topic modeling and word embeddings. These techniques are often compared directly to traditional open coding methods to evaluate their effectiveness. For instance Leeson et al. (2019) used Latent Dirichlet Allocation (LDA) by Blei et al. (2003) to extract topics from text data, assigning weights to words that represent the identified topics. In this, the words in topics were compared to the codes created by human coders. A more recent development in this area involves using BERT embeddings with *hierarchical density-based spatial clustering of applications with noise (HDBSCAN)* (Grootendorst, 2022; Parfenova, 2024). This method provides a more detailed and contextually aware representation of the data. However, this technique still extracts only existing words from the text rather than generating new ideas based on the context.

Another approach explored in the literature involves leveraging WordNet (Miller et al., 1990), a lexical database that represents the semantic relationships between words in a hierarchical structure (Wei et al., 2015; Guetterman et al., 2018). However Wordnet has limited lexical coverage and is not actuvely maintained. ConceptNet, on the other hand, extends beyond WordNet by capturing common sense knowledge and broader connections between concepts, making it more suitable for qualitative coding (Liu and Singh, 2004), but still using only words present in the text, instead of generating new ones.

As for the automation of coding using LLMs, it is worth mentioning that there are two primary approaches: deductive and inductive. Deductive coding is theory-driven, where predefined codes are applied to the data. In contrast, inductive coding is data-driven, allowing codes to emerge organically from the data. Some studies, such as Xiao et al. (2023); Spinoso-Di Piano et al. (2023); Matter et al. (2024); Ziems et al. (2024); Fischer and Biemann (2024), have explored the use of LLMs for automatic deductive code generation where labels are predefined. However, our approach utilizes an inductive coding method based on *grounded theory* (Glaser and Strauss, 2017), allowing insights to naturally emerge from the data.

Existing software for coding Qualitative researchers usually utilize specialized software such as Atlas.ti[*], Dedoose[†], MAXQDA[‡] to aid in manual coding. These tools provide a user-friendly interface for tagging, categorizing, and organizing data. While these platforms offer significant convenience and streamline the workflow, they do not perform coding itself. Instead, they serve as digital extensions of traditional qualitative methods, making it easier to manage large volumes of data.

3 Dataset

The dataset was compiled from student and professor contributions across three social science faculties of different universities. It consists of 600 *code-quote* pairs (see example in Figure 2). As shown in Table 1, most of these studies were based on interviews covering various topics such as values, social expectations, interaction with technology, while one of them involved the analysis of online

[*] https://atlasti.com/
[†] https://www.dedoose.com/
[‡] https://www.maxqda.com/

Code	Quote
Mobility	Well since it's in a smartphone, it's usually always with me.
Helper	For me the most important thing is to have an assistant like Google assistant, meaning it doesn't talk to you itself, it only performs tasks.
Playing	For example I remember playing cities with her...

Figure 2: Dataset examples

reviews. The *Code* column in the dataset represents the *golden standard*, established by consensus among 3 to 5 coders. Coders initially labeled quotes independently, then discussed and agreed on the final golden standard label.

To enhance the dataset, an additional 400 *code-quote* pairs were incorporated from the SemEval-2014 dataset Task 4, which consists of online reviews (Pontiki et al., 2014). This data was manually coded by sociologists, who extracted the main idea of each review. Experiments demonstrated that models trained on the augmented dataset outperformed those trained on the original dataset (see Table 3). As a result, all subsequent experiments were conducted using the augmented dataset of 1,000 examples. The test set size was set to 100 examples (see Table 2). The dataset was split into training and testing sets without a separate validation set. Hyperparameters were selected based on the training results and evaluated on the test set.

N Quotes	Description
	Social Science Studies Data: 600 quotes
78	Study about interaction with self-tracking devices (interviews)
22	Study about life transitions and mobility (interviews)
82	Study about interaction with voice assistants (interviews)
28	Study about museums and cultural experiences (interviews)
25	Study on doctors' experiences with pregnant women (interviews)
110	Study on universal and national values (interviews)
24	Study on procrastination and budget planning (interviews)
56	Study on technology interactions and user feedback (reviews)
175	Study about social expectations (interviews)
	SemEval 2014; Task 4: 400 quotes
211	Restaurant reviews
189	Laptop reviews

Table 1: Summary of Data Sources with descriptions.

4 Automatic evaluation

As previously mentioned, this study focuses exclusively on the open coding phase, while categorizing and clustering codes into higher-order categories (axial coding) remains a separate task that requires distinct experimentation and evaluation, and will be covered in a future work. According to established guidelines, open coding does not necessitate prior knowledge of the research topic, whereas axial coding heavily relies on such knowledge (Miles and Huberman, 1994; Glaser and Strauss, 2017).

Statistic	Overall	Train	Test
Total Quotes	1000	900	100
Social Science Data	600	550	50
SemEval Data	400	350	50
Num of Data Sources	11	11	11
Unique Codes	680	624	94
Average Quote Length	$254.75_{274.28}$	$280.89_{280.89}$	$234.80_{201.61}$
Average Code Length	$19.95_{10.43}$	$20.04_{10.70}$	$19.27_{10.53}$

Table 2: Summary statistics of the dataset and train/test splits. Subscript refers to standard deviation where applicable.

In this study, we compared several open source models: Llama3 (Touvron et al., 2023), Falcon (Pineda et al., 2023), Mistral (Team, 2023), Vicuna (Li et al., 2023), Gemma (Team, 2024), and TinyLlama (Jiang et al., 2023) (see Appendix F), to evaluate their performance in the open coding task. We experimented with different approaches including zero-shot, few-shot (providing 1 to 5 examples), and parameter-efficient fine-tuning (Han et al., 2024) using Low-Rank Adaptation (Hu et al., 2021).

4.1 Metrics

To evaluate the performance of chosen models, two metrics were employed to capture both lexical and semantic similarity: ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019). BERTScore is a metric that computes the similarity between BERT token embeddings of two codes, which helps assess the meaning in the generated output compared to the reference. ROUGE is a lexical similarity measure that calculates the overlap of n-grams (1-unigram overlap, 2-bigram overlap, L-longest common subsequence) between the generated text and the reference text. ROUGE is particularly effective in summarization task (Fabbri et al., 2021), which is valuable when the exact wording of the output needs to match the reference.

4.2 Results

Finetuning Results show that Falcon and Mistral consistently performed better than other models across both the BERTScore and ROUGE metrics, particularly when fine-tuned on the augmented dataset. Falcon achieved the highest BERTSscore (0.7642) when trained on the full dataset, suggesting that it is better at capturing the nuances of sentence meaning compared to other models (see

Dataset Size	Parameters	BERTScore			ROUGE		
		P_{std}	R_{std}	$F1_{std}$	1	2	L
900 with augmentation							
Llama3 (instruct)	8B	$0.713_{0.060}$	$0.787_{0.084}$	$0.747_{0.062}$	0.142	0.033	0.136
Falcon (instruct)	7B	$\mathbf{0.744}_{0.100}$	$0.788_{0.100}$	$\mathbf{0.764}_{0.096}$	**0.204**	**0.089**	**0.202**
Mistral (instruct)	7B	$0.728_{0.076}$	$\mathbf{0.790}_{0.094}$	$0.756_{0.078}$	0.175	0.075	0.166
Vicuna (instruct)	7B	$0.726_{0.081}$	$0.788_{0.095}$	$0.754_{0.080}$	0.188	0.077	0.185
Gemma (instruct)	7B	$0.721_{0.083}$	$0.775_{0.092}$	$0.746_{0.081}$	0.165	0.059	0.159
Tinyllama (chat)	1.1B	$0.738_{0.091}$	$0.781_{0.095}$	$0.758_{0.088}$	0.185	0.073	0.179
500 without augmentation							
Llama3 (instruct)	8B	$0.714_{0.069}$	$0.763_{0.078}$	$0.737_{0.068}$	0.137	0.053	0.135
Falcon (instruct)	7B	$\mathbf{0.735}_{0.098}$	$0.757_{0.096}$	$0.745_{0.092}$	0.147	0.041	0.146
Mistral (instruct)	7B	$0.731_{0.095}$	$\mathbf{0.776}_{0.093}$	$\mathbf{0.751}_{0.088}$	0.180	**0.074**	0.173
Vicuna (instruct)	7B	$0.722_{0.078}$	$0.763_{0.080}$	$0.741_{0.074}$	0.141	0.039	0.137
Gemma (instruct)	7B	$0.702_{0.084}$	$0.769_{0.091}$	$0.733_{0.081}$	0.157	0.068	0.154
Tinyllama (chat)	1.1B	$0.726_{0.078}$	$0.773_{0.089}$	$0.748_{0.077}$	**0.187**	0.074	**0.178**

Table 3: Model Performance on open coding task with and without augmentation. The prompt used was 'Summarize the main idea of a sentence.'

Table 3). Mistral also demonstrated strong performance, especially in its consistency across different dataset sizes, showing a more stable performance with varying data availability.

Augmentation When comparing results between the augmented dataset (1000 examples) and the smaller dataset (600 examples), it is clear that increasing the training dataset size significantly improves model performance. For instance, Falcon's BERTScore increased from 0.7348 to 0.7642, and Mistral's BERTScore improved from 0.7308 to 0.7562. The results show that all models generally improved in performance as the dataset size increased, supporting that larger training datasets lead to better generalization. However, the most significant finding is that the performance, as measured by the BERTScore, plateaued after approximately 100 examples (demonstrated in the Figure 3). This suggests that while additional data beyond 100 examples can still contribute to slight improvements, the majority of performance gains can be achieved with a relatively small amount of data.

Zero-shot and Few-shot In this experiment, we evaluated various models across different settings: zero-shot, one-shot, three-shot, and five-shot scenarios. In the zero-shot setting, no examples were provided to the models, and they had to generate codes based solely on the initial prompt. In the one-shot, three-shot, and five-shot settings, the models were given one, three, and five examples, respectively, to help guide their coding (see Table 4).

The BERTScores across the different models varied depending on the number of examples provided. The performance generally improved when moving from zero-shot to one-shot scenarios, with most models achieving their highest scores with just one example. However, the models exhibited varying behaviors as more examples were provided. Notably, as depicted in Figure 4, Mistral demonstrated continuous improvement across all scenarios, achieving the highest BERTScore in the five-shot scenario. The best settings of models are demonstrated in Table 5.

Model	Zero-shot	1-shot	3-shot	5-shot
Llama3	0.6713	0.7488	0.7308	0.7473
Falcon	**0.7112**	0.7092	0.7195	0.7019
Mistral	0.6945	**0.7501**	**0.7536**	**0.7613**
Vicuna	0.6951	0.7496	0.6790	0.6893
Gemma	0.6951	0.7414	0.7227	0.7339
TinyLlama	0.6928	0.7444	0.7295	0.6893

Table 4: BERT F1 scores for Zero-shot and Few-shot performance across models

5 Human Expert Evaluation

The efficacy of LLMs in automating the open coding phase was evaluated through a comparison with human coders. Three expert qualitative researchers with social science educational background manually coded a selection of sentences, and their codes were compared with those produced by six LLMs in their best performance scenarios (see Table 5). The evaluation process was conducted in two stages.

5.1 Stage 1: Coding and Difficulty Rating

In the first stage, the human coders participated in an expert coding task, where they were presented with a set of 15 sentences (see Appendix D). According to the definition of a code by Saldana (2016) the coders were asked to generate an open code for each sentence that best encapsulated its core meaning. Each code had to be a word or short phrase summarizing the key idea of the sentence. This open coding process was conducted without any prior knowledge of the golden standard labels.

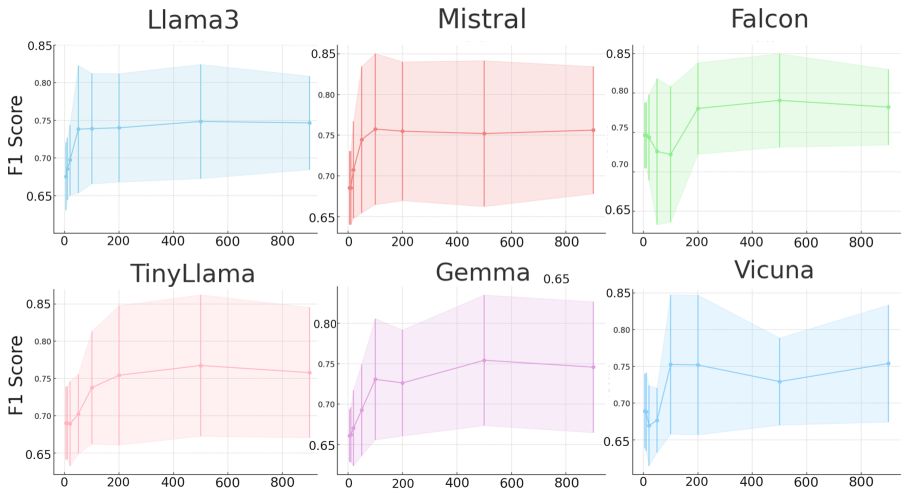


Figure 3: BERT F1 score with an increase of dataset size for all models. The shaded areas represent the standard deviation. The analysis shows how each model benefits from additional data, with some models like Mistral and Falcon displaying higher stability and faster performance gains compared to others. This figure illustrates that the few examples is enough for sufficient finetuning performance.

Additionally, the coders were asked to rate the subjective difficulty of coding each sentence. For each sentence, they chose one of three levels, based on the following criteria: *Easy (1)* - The sentence is straightforward and the code is obvious; *Medium (2)* - The sentence requires more thought, but a clear code can still be assigned; *Difficult (3)* - The sentence is complex or ambiguous, making it difficult to assign a suitable code. In the evaluation process, difficulty was initially assumed based on a 1-to-3 scale, and then compared to the results given by experts. After collecting the difficulty ratings from all coders, the average difficulty was computed across these values.

Upon analysis, we found that the averaged difficulty metric correlated strongly with the length of the sentences. Longer sentences tended to be perceived as more complex. In contrast, traditional lexicon-based readability metrics, such as Flesch Reading Ease (Kincaid, 1975) and the Coleman-Liau Index (Coleman and Liau, 1975)[§], were found to be uncorrelated with the difficulty ratings assigned by the coders. These readability scores, designed for general text comprehension, failed to capture the specific challenges associated with qualitative coding (see Appendix C). As a result, sentence length and the coders' averaged perceived difficulty were more reliable indicators of complexity in this open coding task.

[§]Ward, Alex. 2022. Textstat, https://pypi.org/project/textstat/

Figure 4: Mistral BERT F1 scores across different numbers of examples.

Stage 2: Rating Coder and Model Labels In the second stage, the coders were provided with the labels generated by the other coders, the labels generated by the best-performing LLM models from the first stage, as well as golden standard labels. The coders were asked to rate each label on a scale from 1 to 5, with 5 representing the most accurate and representative coding of the sentence's core idea. This stage aimed to evaluate both the quality of human-generated codes and the performance of the LLMs in comparison to them. The instructions for coders are attached in Appendix E. Two key metrics were used in the evaluation:

5.2 Metrics

Deviation from golden standard For each coder i and each sentence j, the deviation was calculated by comparing the average rating of the humans' and LLMs' codes with the golden standard label. The deviation for each code was computed using

Model	Parameters	Adaptation	Prompt	BERTScore			ROUGE		
				P	*R*	*F1*	*1*	*2*	*L*
Llama3 (instruct)	8B	Finetuning	Summarize the main idea of a sentence.	0.719	0.788	0.751	0.182	0.059	0.167
Falcon (instruct)	7B	Finetuning	From the perspective of a social scientist, summarize the following sentence as you would in thematic coding.	0.745	0.792	0.766	0.210	0.089	0.211
Mistral (instruct)	7B	Finetuning	Can you tell me what the main idea of this sentence is in just a few words?	0.742	0.795	0.766	0.246	0.106	0.235
Vicuna (instruct)	7B	Finetuning	Summarize the main idea of a sentence.	0.734	0.787	0.759	0.194	0.068	0.185
Gemma (instruct)	7B	Finetuning	If you were a social scientist doing thematic analysis, what code would you give to this citation?	0.724	0.784	0.751	0.170	0.066	0.168
TinyLlama (chat)	1.1B	Few-shot (5 examples)	Summarize the main idea of a sentence. Here are examples:	0.768	0.744	0.755	0.176	0.026	0.176

Table 5: Performance of various open-Source LLMs on open coding task across different adaptation methods and prompts. This table presents the BERTScore and ROUGE scores for each model, indicating precision (*P*), recall (*R*), and F1 scores for BERTScore, along with ROUGE scores (1, 2, L). Models were evaluated under different scenarios, including finetuning and few-shot approaches, with prompts designed to align with thematic analysis.

the formula:

$$\text{DGS}_{i,j} = \left(\frac{1}{N} \sum_{k=1}^{N} r_{k,j}^{(i)} \right) - \left(\frac{1}{N} \sum_{k=1}^{N} r_{k,j}^{(\text{GS})} \right)$$

where:
- i is the coder/model for whom the deviation is calculated.
- j is the specific sentence being evaluated.
- N represents the number of experts who rated both the coder/model i and the golden standard (GS) for sentence j.
- $r_{k,j}^{(i)}$ is the rating given by expert k to coder/model i for sentence j.
- $r_{k,j}^{(\text{GS})}$ is the rating given by expert k to the golden standard (GS) for sentence j.

Average DGS To compute an overall measure of deviation for each coder i across all sentences (M - total number of sentences), the average deviation was calculated as follows:

$$\text{Average DGS}_i = \frac{1}{M} \sum_{j=1}^{M} \text{DGS}_{i,j}$$

In this case, *positive deviation* occurs when an expert rates a code higher than the golden standard, resulting in a positive deviation from it. *Negative deviation*, on the other hand, is when a coder rates a sentence lower than the golden standard. Both types indicate a coder's divergence from the golden standard but in different directions, reflecting higher or lower evaluation by humans of a particular code.

Inter-Coder Reliability To assess the reliability and consistency of the codes generated by different coders, including both human coders and LLMs, Krippendorff's alpha (Krippendorff, 2018) was computed. Krippendorff's alpha is a widely used reliability coefficient that quantifies the level of agreement between coders on a set of coding tasks, while accounting for the possibility of chance agreement. It is particularly versatile, as it can handle various types of data, including nominal, ordinal, interval, and ratio-level data, making it well-suited for qualitative research where different coding schemes or scales are used.

Krippendorff's alpha is valuable because it accommodates situations where coders may not agree perfectly and where missing or incomplete data is present, unlike simpler agreement measures like percent agreement or Cohen's kappa (McHugh, 2012), which require complete data and assume equal distribution across categories. It can also handle any number of coders, not just pairs, making it ideal for our study, which involves multiple human coders and LLMs.

5.3 Results

The result shown in the Figure 5(a) indicate that human coders performed exceptionally well in coding difficult sentences, which often involved abstract concepts or nuanced language. However, their performance was less consistent with easier sentences, where LLMs tended to perform better. This discrepancy is likely due to the tendency of human coders to overcomplicate simple statements or overlook straightforward interpretations. One coder, in particular, commented during the evaluation phase that they tended to overinterpret data and make codes too abstract.

This tendency was shown when coding simple sentences, for instance *I can ask the voice assistant what the weather is like.* In this example, LLMs generated codes such as *weather forecast*

or *weather prediction* which are similar to Golden Standard label *weather*. However, humans provided more abstract code: *functional usage, device feature,* and *voice command.* While these additional layers of interpretation may add depth in some contexts, in this case, they introduced unnecessary complexity and deviated from the core meaning of the sentence. Nevertheless, this level of abstraction could be valuable for the next stage of axial coding where codes are organized into hierarchies.

This tendency of human coders to overcomplicate simple sentences was also reflected in the DGS evaluation results (see Figure 5(b)). For human coders (Coders A, B, and C), deviation from golden standard was particularly pronounced for easier sentences. Coder A, in particular, consistently showed positive deviation, being further from golden standard but rated higher by experts. In contrast, LLMs generally exhibited less deviation across sentence complexities. For instance, Llama3 demonstrated positive deviation for medium and difficult sentences, suggesting that it tended to overpredict or generate overly complex codes in certain cases that mimics human expert behavior. Models like Falcon and Mistral showed much lower deviation, particularly for easy and medium sentences, where their labels aligned more closely with the golden standard. Overall, LLMs demonstrated lower and more consistent deviation compared to human coders, particularly for easier sentences. This suggests that LLMs are more reliable in handling straightforward coding tasks. However, as the sentence complexity increased, some models, such as Llama3, exhibited positive deviation, meaning being evaluated higher than golden standard by experts, while other LLMs showed the opposite trend. In contrast, human coders, while showing higher deviation overall, were able to better handle the complexity of difficult sentences, albeit inconsistently.

Despite the effort to ensure consistency in the coding process, the inter-coder reliability, measured using Krippendorff's alpha, was low, with a value of 0.2. This low value indicates a significant lack of agreement between coders, which can be attributed to the subjective nature of the task and the inherent variability in how individuals interpret complex, abstract concepts (Hayes and Krippendorff, 2007). Additionally, the broad definition of a code provided by Saldana (2016) may have allowed for considerable variation in how coders applied and interpreted the codes, further contributing to the low reliability. In comparison, tasks like rating restaurant experiences or product reviews may be better suited to this evaluation metric because they involve more objective criteria (e.g., food quality, and service speed). The task of qualitative code evaluation, however, involves a higher degree of interpretation and abstraction (Galdas, 2017), making it less suitable for standard reliability metrics like Krippendorff's alpha.

6 Discussion

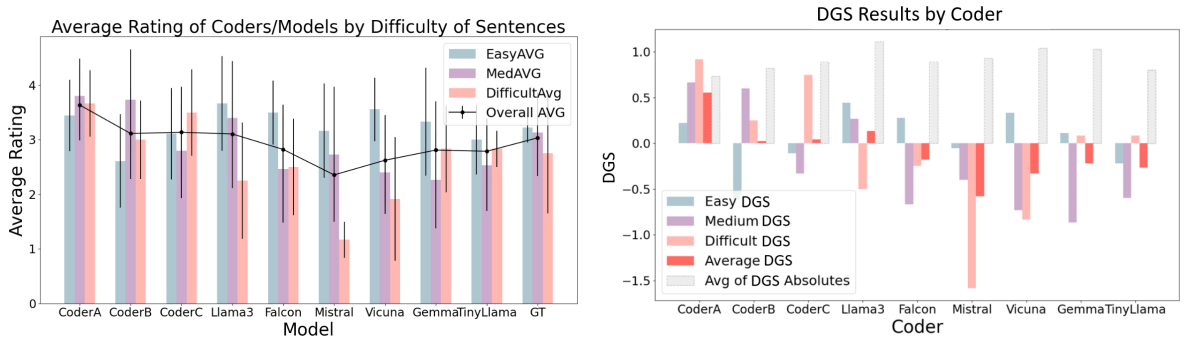
The results of this study revealed several surprising and, at times, counterintuitive findings. Notably, BERTScore performance plateaued after approximately 100 examples, suggesting that effective fine-tuning is achievable with a relatively small dataset. This has important implications for computational social science research, where data is often scarce and difficult to collect.

One of the most unexpected outcomes was that LLMs exhibited less deviation in coding, meaning their outputs were often closer to the golden standard compared to human coders. This finding challenges the common assumption that human coders, with their deep contextual understanding and expertise, would naturally generate more accurate and reliable codes.

Human coders had a tendency to overinterpret the data, adding unnecessary complexity to straightforward sentences. As it was highlighted by one of the experts, this is a pitfall in early qualitative analysis and may reflect an effort to capture nuances that aren't immediately relevant in the initial open coding phase. Interestingly, as research progresses into more advanced stages such as axial coding, these preliminary codes are often refined and simplified. Therefore, it would be fascinating to investigate how human coders' open codes might change if they were later exposed to the results of axial coding (higher order categories).

7 Conclusion

For the task of open coding, we tested six open-source LLMs and compared their performance in finetuning, zero-shot, and few-shot scenarios. Following this, we conducted a human expert evaluation to compare the codes produced by LLMs with those created by human coders. This comparative approach allowed us to assess the strengths and limitations of LLMs in automating qualitative coding



(a) Average rating of LLMs and Humans

(b) Deviation evaluation

Figure 5: Comparison of Average Ratings and Deviation from Golden Standard (DGS) for LLMs and human coders. Panel (a) shows the average ratings given to both human coders (CoderA, CoderB, CoderC) and various LLM models, segmented by sentence difficulty (Easy, Medium, Difficult). The graph highlights that LLMs generally receive higher ratings on easy sentences compared to human coders, while humans excel in coding more complex sentences. Panel (b) presents the DGS results for both human coders and LLMs across different sentence difficulties, with positive and negative deviations from the golden standard.

tasks.

The study revealed important insights into the potential of LLMs, as well as the challenges they face. While for short sentences LLMs tend to be closer to golden standard labels than human coders, they lack the interpretative depth necessary for complex qualitative analysis. Human coders, despite their expertise, often introduce unnecessary complexity into their codes, reflecting a tendency to overinterpret data during the open coding phase. The results demonstrated that LLMs hold great promise in automating the open coding process, especially in domains where the data is straightforward and repetitive, such as customer feedback analysis or social media monitoring. However, for more nuanced tasks, particularly in academic social science research, human coders remain more reliable due to their ability to interpret longer and more difficult sentences that LLMs struggle to handle.

Future work will extend this study into the axial coding stage, with the goal of developing a complete thematic analysis pipeline and evaluating its performance against human expert results. This next phase will assess how effectively LLMs can contribute to the full qualitative coding process and determine whether they are suitable for full automation or better suited as an assistive tool.

Ethics Statement

The use of LLMs in qualitative research introduces new ethical considerations, particularly concerning bias and the potential for automated systems to replicate or amplify human biases. In this study, we took steps to identify and mitigate biases in the models and human coders. However, we acknowledge that the use of LLMs in sensitive research areas requires the development of guidelines to ensure that these tools are used responsibly. The human participants involved in the expert evaluation were fully informed about the study's objectives and provided their consent to participate. Their expertise was crucial in evaluating the performance of the LLMs, and their input was treated with the utmost respect and consideration.

Limitations

Firstly, it is important to highlight that our focus was limited to open coding; we did not explore the full qualitative analysis process, particularly axial coding, which organizes open codes into higher-order categories. Future research could extend this work by investigating whether LLMs can assist in axial coding, potentially offering a complete automation of thematic analysis.

The dataset used in this study focused primarily on social science research, supplemented by online reviews. However, the scope of QDA extends beyond this, encompassing domains such as social media posts, medical texts, media content, and field notes. It means that the current models may not generalize well across all potential domains, especially those with specialized terminology and professional knowledge requirements.

Another limitation is that while we used established metrics such as BERTScore and ROUGE, these may not fully capture the quality and inter-

pretative nature of qualitative coding. Developing more nuanced evaluation metrics that align better with the goals of qualitative research would be an important step forward.

Lastly, this study compared LLM performance to human coders based on alignment with a golden standard, which may not be the ideal measure of codes' quality. In real-world coding scenarios, human coders often reach a consensus after discussion, whereas LLMs do not undergo such collaborative process. This raises the question of whether LLM-generated codes could eventually reach consensus between models or if their role is better suited as an assistive tool for human coders. Further investigation into a hybrid approach—where LLMs handle initial coding and humans provide further refinement and interpretation, especially for complex or ambiguous data—would be a valuable direction for future research.

References

H. Alshenqeeti. 2014. Interviewing as a data collection method: A critical review. *English Linguistics Research*, 3:39–45.

E.G. Avjyan. 2005. Asynchronous on-line focus group: technology and procedures of conducting. *Southern Russian Journal of Social Sciences*, (1):116–129.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bonnie S Brennen. 2021. *Qualitative research methods for media studies*. routledge.

Ştefania Bumbuc. 2016. About subjectivity in qualitative data interpretation. In *International Conference Knowledge-Based Organization*, volume 22, pages 419–424.

Kathy Charmaz. 2014. *Constructing Grounded Theory*. SAGE Publications.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

John W Creswell. 2016. *30 Essential Skills for the Qualitative Researcher*. SAGE Publications.

Norman K Denzin and Yvonna S Lincoln. 2005. *The Sage Handbook of Qualitative Research*. SAGE Publications.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation.

Tim Fischer and Chris Biemann. 2024. Exploring large language models for qualitative data analysis. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 423–437, Miami, USA. Association for Computational Linguistics.

Uwe Flick. 2018. *The SAGE Handbook of Qualitative Data Collection*. SAGE Publications.

Paul Galdas. 2017. Revisiting bias in qualitative research: Reflections on its relationship with funding and impact.

Barney Glaser and Anselm Strauss. 2017. *Discovery of grounded theory: Strategies for qualitative research*. Routledge.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

T.C. Guetterman, T. Chang, M. Dejonckheere, T. Basu, E. Scruggs, and V.G.V. Vydishwaran. 2018. Augmenting qualitative text analysis with natural language processing: Methodological study. *J. Med. Internet Res.*, 20.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

Monique M Hennink, Bonnie N Kaiser, and Vincent C Marconi. 2020. Code saturation versus meaning saturation: How many interviews are enough? *Qualitative Health Research*, 29(11):1487–1496.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Zhe Jiang et al. 2023. Tinyllama: Distilling large language models for efficiency. *arXiv preprint arXiv:2310.05637*.

JP Kincaid. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Chief of Naval Technical Training*.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

David MJ Lazer, Alex Pentland, Duncan J Watts, Sinan Aral, Susan Athey, Noshir Contractor, Deen Freelon, Sandra Gonzalez-Bailon, Gary King, Helen Margetts, et al. 2020. Computational social science: Obstacles and opportunities. *Science*, 369(6507):1060–1062.

W. Leeson, A. Resnick, D. Alexander, and J. Rovers. 2019. Natural language processing (nlp) in qualitative public health research: a proof of concept study. *International Journal of Qualitative Methods*, 18.

Chenghao Li, Fangchen Xie, et al. 2023. Vicuna: An open-source chatbot. *FastChat: Open Assistant*. Available at https://vicuna.lmsys.org/.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Kathleen M MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. 2008. *Codebook Development for Team-Based Qualitative Analysis*. Cultural Anthropology Methods.

Daniel Matter, Miriam Schirmer, Nir Grinberg, and Jürgen Pfeffer. 2024. Close to human-level agreement: Tracing journeys of violent speech in incel posts with gpt-4-enhanced annotations.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Matthew B Miles and A Michael Huberman. 1994. *Qualitative data analysis: An expanded sourcebook*. sage.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Haradhan Kumar Mohajan et al. 2018. Qualitative research methodology in social sciences and related subjects. *Journal of economic development, environment and people*, 7(1):23–48.

Janice M Morse. 2015. *Critical Issues in Qualitative Research Methods*. SAGE Publications.

Angelina Parfenova. 2024. Automating the information extraction from semi-structured interview transcripts. In *Companion Proceedings of the ACM on Web Conference 2024*, WWW '24. ACM.

Michael Quinn Patton. 2014. *Qualitative Research & Evaluation Methods: Integrating Theory and Practice*. SAGE Publications.

Felipe Pineda, Raphael Milliere, Andreas Vlachos, Myle Ott, Richard Yates, Amelia Glaese, et al. 2023. The falcon series of language models. *arXiv preprint arXiv:2306.01116*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Gery W Ryan and H Russell Bernard. 2003. *Techniques to Identify Themes*, volume 15. SAGE Publications.

J. Saldana. 2016. *The Coding Manual for Qualitative Researchers*, 3rd edition. Sage, Los Angeles, CA.

Cesare Spinoso-Di Piano, Samira Rahimi, and Jackie Cheung. 2023. Qualitative code suggestion: A human-centric approach to qualitative coding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14887–14909, Singapore. Association for Computational Linguistics.

Gemma AI Research Team. 2024. Gemma: An instructable, open-source large language model.

Mistral AI Team. 2023. Mistral: Efficient pretraining of transformer language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, and Xianyu Bao. 2015. A semantic approach for text clustering using wordnet and lexical chains. *Expert Syst. Appl.*, 42(4):2264–2275.

Ziang Xiao, Xingdi Yuan, Q. Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *28th International Conference on Intelligent User Interfaces*, IUI '23. ACM.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

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

A Prompts used for finetuning

In the conducted experiments, several prompts were designed to guide the models in performing open coding tasks. These prompts varied in their level of explicitness, the perspective they asked the model to adopt, and the amount of detail they requested. Additionally, the experiments compared the effect of using a line break versus a period (dot) at the end of each prompt to assess how subtle changes in prompt formatting might influence the model's performance (see Appendix B). Below is a brief description of each prompt:

- **Explicit Instruction (Prompt 1):** *Summarize the main idea of a sentence.* This prompt provides a direct and clear instruction to the

model, asking it to summarize the core idea of a given sentence. The expectation is for the model to extract the primary message or theme conveyed in the sentence with no additional context or framing. This prompt is designed to test the model's ability to perform a straightforward task without needing implicit knowledge.

- **Informal Request (Prompt 2):** *Can you tell me what the main idea of this sentence is in just a few words?* This prompt is phrased as a casual, conversational question, asking the model to summarize the sentence in "just a few words." The informal tone encourages a more concise and simplified response, aiming to capture how well the model can extract the essence of the sentence in a more natural, everyday context.

- **Expert Angle (Prompt 3):** *From the perspective of a social scientist, summarize the following sentence as you would in thematic coding.* This prompt takes a more specialized approach, asking the model to assume the perspective of a social scientist performing thematic coding. The expectation here is for the model to not only summarize the sentence but to apply a more analytical and structured lens, possibly introducing higher-level categorizations that would be typical in qualitative data analysis.

- **Impersonalization (Prompt 4):** *If you were a social scientist doing thematic analysis, what code would you give to this citation?* In this prompt, the model is asked to act as a social scientist and assign a code, which is a brief label representing the central idea of the sentence. It emphasizes the objectivity of thematic analysis, expecting the model to depersonalize the task and focus on generating an appropriate label that accurately reflects the content.

- **Detailed Explanation (Prompt 5):** *Explain in a couple of words the primary thought expressed in the following text.* This prompt asks the model to provide a more detailed, thorough explanation of the primary thought behind the text. It is designed to encourage the model to go beyond a simple summary and delve into the deeper meaning or implications of the sentence.

- **Simplified Task (Prompt 6):** *What is the gist of this sentence?* This prompt simplifies the task by asking for the "gist" of the sentence. It challenges the model to provide a very brief and straightforward summary, focusing on distilling the essential meaning of the sentence.

B Detailed fine-tuning results

These results (see Table 6) demonstrate the performance of various models when fine-tuned on the task of open coding using different prompts. BERTScore and ROUGE are reported.

C Sentence difficulty and lexicon-based metrics

In this section we present the Table 7 with the assessed difficulty levels of sentences using a range of lexicon-based metrics. This includes readability scores like Flesch Reading Ease and Coleman-Liau Index, along with indicators of linguistic complexity such as sentence length and syllable count. The table provides a color-coded overview of sentence difficulty based on calculated average perceived difficulty of a sentence.

D Sentences used for expert coding

This section lists the specific sentences that were selected from the test set for the expert coding phase (see Table 8). These sentences span a broad spectrum of themes and linguistic features, from simple descriptive statements to complex and long sentences.

E Instructions given to coders in the second stage

This section describes the instructions provided to coders during the second stage of evaluation, illustrated through an interface screenshot (see Figure 6).

F Links to Models on Hugging Face

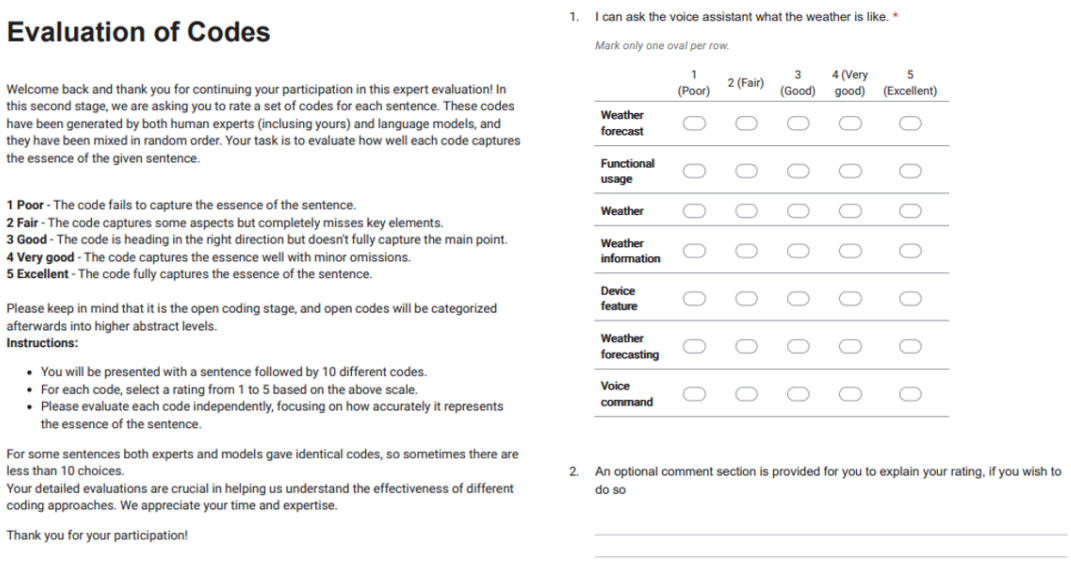
- **Llama3**: https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
- **Falcon**: https://huggingface.co/tiiuae/falcon-7b-instruct
- **Mistral**: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Model	BERTScore			ROUGE		
	P_{std}	R_{std}	$F1_{std}$	1	2	L
Summarize the main idea of a sentence\n						
Llama3	$0.713_{0.060}$	$0.758_{0.040}$	$0.734_{0.062}$	0.141	0.033	0.153
Falcon	$0.746_{0.073}$	$0.782_{0.097}$	$0.764_{0.095}$	0.176	0.047	0.189
Mistral	$0.729_{0.076}$	$0.787_{0.093}$	$0.756_{0.078}$	0.178	0.047	0.195
Vicuna	$0.731_{0.063}$	$0.777_{0.095}$	$0.753_{0.079}$	0.163	0.028	0.182
Gemma	$0.712_{0.084}$	$0.738_{0.078}$	$0.745_{0.080}$	0.163	0.030	0.168
TinyLlama	$0.718_{0.072}$	$0.775_{0.090}$	$0.757_{0.087}$	0.164	0.052	0.158
Summarize the main idea of a sentence.						
Llama3	$0.718_{0.072}$	$0.788_{0.089}$	$0.750_{0.073}$	0.181	0.059	0.166
Falcon	$0.738_{0.099}$	$0.787_{0.103}$	$0.761_{0.096}$	0.213	0.077	0.210
Mistral	$0.719_{0.072}$	$0.768_{0.086}$	$0.742_{0.075}$	0.157	0.055	0.148
Vicuna	$0.733_{0.079}$	$0.787_{0.095}$	$0.758_{0.081}$	0.193	0.068	0.185
Gemma	$0.719_{0.071}$	$0.779_{0.089}$	$0.746_{0.072}$	0.172	0.049	0.166
TinyLlama	$0.736_{0.083}$	$0.788_{0.092}$	$0.760_{0.081}$	0.207	0.074	0.199
Can you tell me what the main idea of this sentence is in just a few words?						
Llama3	$0.688_{0.055}$	$0.778_{0.084}$	$0.729_{0.061}$	0.116	0.034	0.110
Falcon	$0.753_{0.105}$	$0.787_{0.108}$	$0.768_{0.102}$	0.236	0.104	0.239
Mistral	$0.742_{0.106}$	$0.795_{0.106}$	$0.766_{0.101}$	0.246	0.106	0.235
Vicuna	$0.691_{0.060}$	$0.783_{0.087}$	$0.732_{0.063}$	0.168	0.047	0.164
Gemma	$0.711_{0.075}$	$0.786_{0.093}$	$0.746_{0.079}$	0.171	0.057	0.168
TinyLlama	$0.725_{0.083}$	$0.789_{0.090}$	$0.754_{0.079}$	0.178	0.067	0.177
From the perspective of a social scientist, summarize the following sentence as you would in thematic coding\n						
Llama3	$0.698_{0.059}$	$0.784_{0.083}$	$0.738_{0.062}$	0.130	0.033	0.119
Falcon	$0.745_{0.109}$	$0.792_{0.105}$	$0.766_{0.102}$	0.210	0.089	0.211
Mistral	$0.688_{0.060}$	$0.785_{0.086}$	$0.732_{0.064}$	0.139	0.041	0.131
Vicuna	$0.713_{0.080}$	$0.778_{0.094}$	$0.743_{0.080}$	0.169	0.061	0.166
Gemma	$0.721_{0.085}$	$0.784_{0.093}$	$0.749_{0.082}$	0.180	0.070	0.177
Tinyllama	$0.718_{0.073}$	$0.776_{0.083}$	$0.745_{0.072}$	0.165	0.053	0.158
From the perspective of a social scientist, summarize the following sentence as you would in thematic coding.						
Llama3	$0.685_{0.082}$	$0.781_{0.064}$	$0.733_{0.081}$	0.136	0.025	0.154
Falcon	$0.754_{0.066}$	$0.778_{0.091}$	$0.759_{0.088}$	0.181	0.048	0.190
Mistral	$0.740_{0.067}$	$0.780_{0.088}$	$0.756_{0.071}$	0.172	0.045	0.187
Vicuna	$0.718_{0.071}$	$0.780_{0.094}$	$0.753_{0.073}$	0.165	0.039	0.185
Gemma	$0.700_{0.072}$	$0.780_{0.085}$	$0.746_{0.080}$	0.180	0.046	0.187
TinyLlama	$0.729_{0.076}$	$0.778_{0.089}$	$0.754_{0.080}$	0.169	0.046	0.183
If you were a social scientist doing thematic analysis, what code would you give to this citation?						
Llama3	$0.692_{0.060}$	$0.785_{0.083}$	$0.735_{0.064}$	0.064	0.043	0.126
Falcon	$0.736_{0.093}$	$0.785_{0.101}$	$0.759_{0.092}$	0.206	0.076	0.200
Mistral	$0.686_{0.057}$	$0.785_{0.082}$	$0.731_{0.061}$	0.132	0.044	0.123
Vicuna	$0.719_{0.070}$	$0.789_{0.091}$	$0.751_{0.073}$	0.183	0.063	0.169
Gemma	$0.724_{0.085}$	$0.784_{0.091}$	$0.751_{0.082}$	0.170	0.066	0.168
Tinyllama	$0.720_{0.071}$	$0.778_{0.083}$	$0.747_{0.072}$	0.186	0.053	0.182
What is the gist of this sentence?						
Llama3	$0.680_{0.064}$	$0.780_{0.086}$	$0.725_{0.066}$	0.129	0.042	0.121
Falcon	$0.731_{0.091}$	$0.780_{0.098}$	$0.754_{0.089}$	0.182	0.080	0.179
Mistral	$0.726_{0.079}$	$0.785_{0.095}$	$0.753_{0.079}$	0.165	0.057	0.160
Vicuna	$0.720_{0.070}$	$0.781_{0.089}$	$0.748_{0.072}$	0.172	0.055	0.162
Gemma	$0.707_{0.077}$	$0.773_{0.091}$	$0.737_{0.076}$	0.152	0.059	0.146
Tinyllama	$0.713_{0.057}$	$0.773_{0.079}$	$0.741_{0.061}$	0.143	0.032	0.139
Explain in a couple of words the primary thought expressed in the following text\n						
Llama3	$0.691_{0.062}$	$0.783_{0.085}$	$0.733_{0.066}$	0.120	0.038	0.110
Falcon	$0.734_{0.078}$	$0.778_{0.090}$	$0.754_{0.078}$	0.171	0.049	0.165
Mistral	$0.698_{0.067}$	$0.780_{0.088}$	$0.735_{0.070}$	0.141	0.038	0.131
Vicuna	$0.703_{0.072}$	$0.780_{0.088}$	$0.738_{0.072}$	0.155	0.048	0.148
Gemma	$0.706_{0.064}$	$0.786_{0.086}$	$0.742_{0.066}$	0.177	0.053	0.170
Tinyllama	$0.720_{0.077}$	$0.784_{0.091}$	$0.750_{0.078}$	0.168	0.071	0.163
Explain in a couple of words the primary thought expressed in the following text.						
Llama3	$0.700_{0.068}$	$0.784_{0.055}$	$0.747_{0.063}$	0.142	0.025	0.152
Falcon	$0.752_{0.088}$	$0.779_{0.061}$	$0.760_{0.086}$	0.183	0.042	0.193
Mistral	$0.738_{0.070}$	$0.790_{0.090}$	$0.759_{0.073}$	0.173	0.047	0.183
Vicuna	$0.717_{0.066}$	$0.780_{0.094}$	$0.752_{0.099}$	0.161	0.025	0.182
Gemma	$0.708_{0.068}$	$0.778_{0.079}$	$0.746_{0.098}$	0.172	0.039	0.186
TinyLlama	$0.728_{0.073}$	$0.778_{0.091}$	$0.755_{0.089}$	0.168	0.053	0.168

Table 6: Detailed Fine-tuning Results. The following table presents the detailed results from fine-tuning experiments, including precision (P), recall (R), F1 score, and ROUGE across different models and prompts.

Sentence	Length	Difficulty assumed	Difficulty avg	Flesch Reading Ease	Coleman-Liau	Automated Index	Difficult Words	Syllable count
Quote1	51.00	1.00	1.50	46.44	10.80	8.50	1.00	14.00
Quote2	55.00	1.00	1.50	85.69	4.74	3.30	1.00	14.00
Quote3	75.00	1.00	1.00	42.38	12.28	9.70	4.00	21.00
Quote4	101.00	2.00	1.50	103.12	3.05	2.20	2.00	23.00
Quote5	105.00	1.00	1.50	87.72	8.03	6.30	3.00	24.00
Quote6	155.00	1.00	1.50	67.42	7.50	13.70	3.00	39.00
Quote7	179.00	3.00	2.25	88.74	4.83	4.00	4.00	42.00
Quote8	194.00	2.00	1.75	40.01	13.41	19.00	6.00	51.00
Quote9	289.00	2.00	2.00	91.82	5.79	6.10	8.00	65.00
Quote10	291.00	2.00	1.75	85.89	4.39	4.60	5.00	70.00
Quote11	309.00	3.00	2.25	80.17	4.54	3.50	7.00	77.00
Quote12	350.00	3.00	2.75	67.79	9.05	11.10	9.00	86.00
Quote13	379.00	2.00	2.75	72.19	6.22	10.60	6.00	96.00
Quote14	889.00	3.00	2.75	75.20	7.53	7.20	21.00	218.00
Quote15	1117.00	3.00	2.75	54.09	8.43	16.70	22.00	282.00

Table 7: Sentence Difficulty and Lexicon-based Metrics. Full quotes are in Appendix E.



Evaluation of Codes

Welcome back and thank you for continuing your participation in this expert evaluation! In this second stage, we are asking you to rate a set of codes for each sentence. These codes have been generated by both human experts (inclusing yours) and language models, and they have been mixed in random order. Your task is to evaluate how well each code captures the essence of the given sentence.

1 Poor - The code fails to capture the essence of the sentence.
2 Fair - The code captures some aspects but completely misses key elements.
3 Good - The code is heading in the right direction but doesn't fully capture the main point.
4 Very good - The code captures the essence well with minor omissions.
5 Excellent - The code fully captures the essence of the sentence.

Please keep in mind that it is the open coding stage, and open codes will be categorized afterwards into higher abstract levels.
Instructions:

- You will be presented with a sentence followed by 10 different codes.
- For each code, select a rating from 1 to 5 based on the above scale.
- Please evaluate each code independently, focusing on how accurately it represents the essence of the sentence.

For some sentences both experts and models gave identical codes, so sometimes there are less than 10 choices.
Your detailed evaluations are crucial in helping us understand the effectiveness of different coding approaches. We appreciate your time and expertise.

Thank you for your participation!

1. I can ask the voice assistant what the weather is like. *

Mark only one oval per row.

	1 (Poor)	2 (Fair)	3 (Good)	4 (Very good)	5 (Excellent)
Weather forecast					
Functional usage					
Weather					
Weather information					
Device feature					
Weather forecasting					
Voice command					

2. An optional comment section is provided for you to explain your rating, if you wish to do so

Figure 6: Screenshot of a user interface for the second stage of expert evaluation.

Num	Sentence
Quote1	She doesn't always understand correctly what I say.
Quote2	I can ask the voice assistant what the weather is like.
Quote3	The food was delicious and the waiter was incredibly helpful and attentive.
Quote4	I was so worried. The thoughts kept spinning in my head and I'd lay there with my eyes open for hours.
Quote5	Had a great experience at this restaurant... staff was pleasant; food was tasty and large in portion size.
Quote6	I wasn't really as concerned about portability (it's a very large laptop) but it's not hard to move around or take on a trip which was a pleasant surprise.
Quote7	Well, often people eat up guilt. They start to, I don't know, do bad things. They start drinking, for example. I don't know, they start to ruin themselves in every possible way.
Quote8	I think that everything is possible in this world, that everyone will definitely understand that there is happiness for them, so it gives them positive emotions and develops their values within.
Quote9	...there are a lot of memes about Duolingo, even if you look on the Internet. About this green owl, which, if you haven't started speaking Mexican, Spanish, comes to you and kills your whole family with a shotgun. It's very active there, yes. As if you should, you promised us and so on.
Quote10	Interviewer: Okay, can procrastination be funny? Informant: Not funny, THIS IS NOT FUNNY. I don't know why do I do this... you know... it's like... feeling like you don't want to learn. It's not funny because it requires rest, you can't get it out of your head that you don't want to do it.
Quote11	Well, there is a probability. But I would say that each person has his own head on his shoulders. That is, he defines his own barriers, as they say. Every person determines his own barriers. I mean, if he's, I don't know. If he wants to take risks, let him take risks. It's everybody's business in principle.
Quote12	For me, fear is a constant companion. I think, because shame and fear, well, shame is probably an emotion that you realize that you are justifiably ashamed, probably, that is certain moral norms that you have violated, so you are ashamed. Fear, on the other hand, is an emotion that can occur regardless of whether you've done something wrong or not.
Quote13	I don't sleep much, only 5 hours a day. But I know sleep is important, because I am often tired and, due to lack of sleep, I cannot listen. It's probably important to go to classes, rest and then start studying again, although I have a lot of problems with this, because if I rest, I don't start studying again, so for me it's better to continue studying during the day to finish.
Quote14	House in some small town, or at least I live in Korolev, in this area. Pets and everything connected with it. I would like to see this come true. Some kind of stability, a well-paid job. Positive feelings... in some kind of future, not quite distant, but not exactly near, which is exactly tomorrow. Can plan for a couple of months (it all depends on how my goals in life change, it's all very changeable, it seems to me). Ten years later - still a husband, perhaps children, but not a fact. Stable work is quite possible. Animals are possible, maybe not in the house, but in the apartment. If you earn enough money, you can have a separate apartment. The only obstacle that can become is myself, some complexes, self-doubt, perhaps psychological problems, all that. If, for example, you don't have enough qualifications for a job, you can finish your studies and gain some more knowledge.
Quote15	I understand that there is an expression in the Bible that everything works out for the good of the loving God, and in my life, analyzing global situations from above and having lived some certain things in my life, milestones, I saw that having made this or that choice, having destroyed something, having left somewhere, for example, having changed even my job, initially I didn't want to lose it, I didn't want to leave this situation. I was comfortable there, it was good for me, and suddenly it all broke, destroyed, I think ah how bad, ah how sad, but having experienced all this and looking back, I realize that I gained much more. That is, I have morally matured, I have experienced some things, I realized that I will know in the next situation what to do and how to act, how to perceive. And globally I gained more, i.e. I, for example, found myself in another job, which now suits me better or at that time suited me better than what I had before. Even though initially it was just the collapse of my whole life, everything, everything is very bad there, I mean I realize that everything ends well anyway.

Table 8: Sentences used for expert coding.

- **Vicuna**: https://huggingface.co/lmsys/vicuna-7b-v1.5
- **Gemma**: https://huggingface.co/google/gemma-7b-it
- **TinyLlama**: https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0

G LoRA Configuration

This section provides the LoRA (Low-Rank Adaptation) configuration used for fine-tuning the models in this study. The configuration includes the rank r, the scaling factor α, target modules, dropout rate, bias handling, and task type. Below is the code snippet used for configuring LoRA:

```
config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["gate_proj", "up_proj", "down_proj"],
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

model = get_peft_model(model, config)
print_trainable_parameters(model)

generation_config = model.generation_config
generation_config.max_new_tokens = 15
generation_config.temperature = 0.7
generation_config.top_p = 0.7
generation_config.num_return_sequences = 1
generation_config.pad_token_id = tokenizer.eos_token_id
generation_config.eos_token_id = tokenizer.eos_token_id
```