Qualitative Coding with GPT-3 and -4

COGS 402 Report

Rishika Bose, Student number 27700475

Supervisor: Dr Peter Reiner

Abstract

Qualitative analysis of text data requires the analyst to find recurring themes and match each text to the themes (or 'codes') one-by-one. Automating this last step could save researchers significant resources. Large language models such as GPT-3 have allowed the automation of many tasks previously dependent on humans, and it may be worth investigating whether they could also be used to automate qualitative coding. In this paper, we carry out qualitative coding with GPT-3 and GPT-4 using a variety of prompts, and assess them against each other and against human experts. We find that with respect to human coding, both models perform with over 95% accuracy, and over 70% sensitivity (or recall) and precision when given 5 examples alongside detailed instructions. GPT-4 performs significantly better than GPT-3 with no or 1 example. These results imply that these models could serve as a useful tool for qualitative research.

Introduction

Qualitative analysis of text data, such as from interviews or surveys, requires the analyst to go through the texts, find recurring themes, and match each text to the themes (or 'codes') one-by-one [1]. Automating this last step could save researchers significant resources. Large language models such as GPT-3 have allowed the automation of many tasks previously dependent on humans, such as chatting with customers, building web applications, and summarising complex text [2-4]; it may be worth investigating whether they could also be used to automate qualitative coding.

In '*Social Science Research: Principles, Methods, and Practices*', Bhattacharjee notes that 'Unlike quantitative analysis...qualitative analysis is heavily dependent on the researcher's analytic and integrative skills and *personal knowledge of the social context* where the data is collected.' Machine learning methods such as regression or tree-based ensembles have been used in the past to carry out 'sentiment analysis', which typically involves classifying a comment into simple classes such as 'positive' or 'negative' [5]. These models are fairly successful at such tasks, but perform poorly when the categories grow more numerous or more complex [6]. Qualitative coding frameworks often consist of numerous subtle categories, and differentiating between them can require contextual knowledge that simple machine learning models do not have.

An advantage of LLMs such as GPT-3 and GPT-4 is that they are pre-trained on a variety of texts from different domains, allowing them to perform well in diverse tasks [7,8] . It is therefore possible that these models could be more successful in applying complex coding frameworks than models trained on data from one domain. Additionally, simpler models usually require large amounts of context-specific data to train on [5]. LLMs' 'contextual knowledge' could also help in classifying texts with only a few, or even no context-specific examples - referred to as 'few-shot' or 'zero-shot' classification [7]. This is

especially useful for research tasks where large amounts of training data (e.g. already-coded texts) are not available.

In this paper, we investigate if GPT-3 and -4 can carry out qualitative coding with performance comparable to expert human coders. Additionally, we compare the performance of these models to each other, and discuss the creation of a tool to automate this process.

Methods

Study design and participants

This was a secondary analysis of data collected by Cabrera et al. (2014) [9], which examined public attitudes towards pharmacological enhancement through the contrastive vignette technique [10]. Participants were presented with a survey containing a descriptive vignette of a person named John (e.g. '"John is a healthy 35-year-old man who has had modest challenges being cheerful ever since he was a young boy"), and then were asked to rate how comfortable they felt with John having taken a pill to enhance a given ability. This was followed by a free-response box that asked participants to provide, in their own words, their reasons for answering as they did.

Thematic analysis of the free-response comments was carried out by two coders. This involved going through each comment and extracting themes that appeared often (e.g. 'Not comfortable with the pill due to side effects', or 'Comfortable with the pill due to its positive benefits'). Each theme was assigned a unique numeric code, and then, depending upon its content, each comment was assigned one or more of these numeric codes by both coders. The final list of codes comprised those both coders had agreed on, after discussion. Further details can be found in Cabrera et al. (2014) [9]. A subset of this data (30 comments and their associated codes) was used for this analysis.

Procedures

1. Preprocessing

The coded dataset from the analysis by Cabrera et al. was processed so that each code was encoded by a single binary column, which would be equal to 1 if the code was assigned to that comment and 0 otherwise.

The coding framework was edited so that each code had a numeric label, going from 1 to 29 (Table 1).

2. Prompt engineering

To get useful and human-comparable responses from GPT-3 and -4, an appropriate prompt must be given to the model. Small differences in content and phrasing of prompts has been found to alter results significantly [11]. Past research in this field, often referred to as 'prompt engineering', has found a few different methods of prompting that can lead to higher-quality or more context-specific responses [11], including: 'role prompts' that tell the model what role to assume when responding; 'shot prompts' than list examples that the model can emulate; and instructions (referred to below as 'context prompts') that provide context and explain the task clearly [11, 12].

In order to systematically investigate what kind of prompt would lead to useful results, text for prompts were written for each of these categories (e.g. role prompt text: 'You are a social scientist'). Then these texts were combined iteratively to form a series of prompts with different attributes (e.g. role prompt + simple context prompt; role prompt + simple context prompt + examples). The full list is available in the supplementary materials.

3. Generating responses

The models were interacted with through the OpenAI API. For GPT-3, the Completion function was used, with the 'text-davinci-003' model (the most advanced from the GPT-3 series). The temperature, which controls the 'randomness' of the response, was set to 0 for maximum reproducibility. The maximum number of tokens allowed in the response was 450, as this would allow the model to list every code in the framework if it wanted to. 10 different prompt combinations were used with these settings, with the model being called to generate responses for a set of 30 comments for each combination.

For GPT-4, the ChatCompletion function was used. The only difference in method from the above was that the 'role prompt' text, instead of being part of the prompt itself, was passed as the 'system prompt'. The other prompt types were combined as before and passed as the 'user prompt'. 10 total combinations were used, as before, with responses being generated for the same set of 30 comments.

4. Evaluation

The models' responses were not always in the same format, especially when they had no examples to draw from. Some examples from two prompt types are shown below.

Prompt type	**Response samples**
Role prompt + simple context (no examples)	Summary: The text expresses discomfort with the idea of the individual taking the pill, citing reasons such as changing something fundamental (2), potential side effects (4), risks outweighing the benefits (8), social pressure to fit in (10), god or religious concerns (12), other fairness, social issues, or character concerns (13), and permanent changes (15).
	2, 5, 6, 7, 10, 11, 12, 13, 15, 17, 18, 19, 20, 21, 22, 24, 25, 27, 28
Story prompt + multiple examples	6 Not comfortable with him taking the pill - no need, nor medically necessary; 11 Not comfortable with him taking the pill - legal or prescribed

	6 Not comfortable with him taking the pill - no need, nor medically necessary; 18 Comfortable with him taking the pill - his choice

Table 1. Samples of model responses

Usually, the responses were either a list of numbers, or a list of numbered code descriptions. Therefore, we felt it was reasonable to assume that any number mentioned in the response was meant to represent the correspondingly numbered code, and extract the numbers from each response. These numbers were then transformed into a series of binary columns, with each column encoding a code, just as for the human-coded data. This allowed for easy comparison between the various codings.

To evaluate the match between the models' responses and the human coding, accuracy, precision, and sensitivity (also known as recall) were calculated for each set of responses [13] . Each time a code was assigned by both the human and model was counted as a true positive; each time a code was not assigned by both was a true negative; if the human assigned a code but the model didn't it was a false negative; vice versa was a false positive (as shown in Table 1). The metrics listed were calculated using these values (the equations are described in Table 2; see 'Precision and recall' in *Encyclopedia of Machine Learning*, pg 781, for further details).

	Human assigned	**Human not assigned**
Model assigned	$True\ positive$	$False\ positive$
Model not assigned	$False\ negative$	$True\ negative$

Table 2. Our definition of true positives, false positives, true negatives and false negatives

Metric	**Calculation**
Accuracy	$(TP + TN)/(TP + FP + TN + FN)$
Precision	$TP/(TP + FP)$
Sensitivity (or recall)	$TP/(TP + FN)$
$TP = True\ positive;\ TN = True\ negative;\ FP = False\ positive;\ FN = False\ negative$	

Table 3. Calculation of accuracy, precision, and recall

The code for the procedures described above can be found in the associated Github repository [13].

Objectives and Outcomes of Interest

The aim of this analysis was to compare the accuracy, precision, and sensitivity scores for each set of model coding responses against the human-coded dataset [14].

Accuracy evaluates the number of exact matches between the model and human. This can be misleadingly high, because it could be possible to get a fairly high accuracy just by not assigning anything (as most of the time, most codes wouldn't apply to a comment). Therefore it is more useful to focus on precision and sensitivity. Precision answers the question - 'of all the codes the model assigned, how many did the human also assign?'. Sensitivity (or recall) answers the question - 'of all the codes the human assigned, how many did the model also assign?'. A high precision and a high sensitivity would imply a model that was performing very similarly to the human coder.

Results

With only a role prompt and some simple context, GPT-3 achieved an accuracy of 0.62, with a precision of 0.12 and sensitivity of 0.67. Its performance improved with a more detailed context and the addition of an example (accuracy: 0.88, precision: 0.35, sensitivity: 0.67). There was a dramatic spike in precision with the addition of 4 more examples to the prompt (accuracy: 0.96, precision: 0.67, sensitivity: 0.73). The best overall performance was observed with the 'story prompt' with multiple examples, as well as an instruction to constrain the number of codes assigned to 'only the most appropriate' (accuracy: 0.96 , precision: 0.70, sensitivity: 0.77).

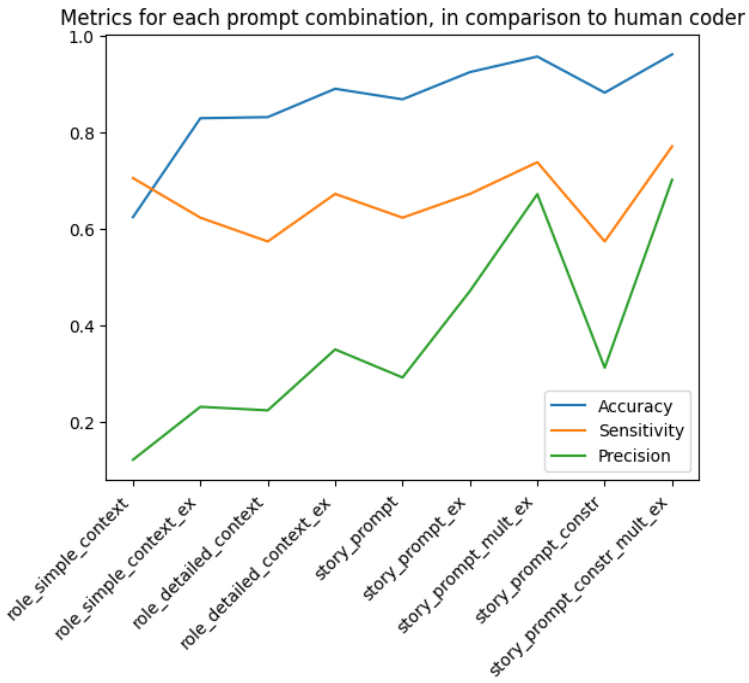

Figure 1. GPT-3 performance with different prompts (prompts roughly increase in complexity from left to right).

GPT-4 performed better than GPT-3 with the simple context + role prompt (accuracy: 0.87, precision: 0.33, sensitivity: 0.78). Precision improved with more detailed context as well as with an example

(accuracy: 0.95, precision: 0.64, sensitivity: 0.78). There was another jump in precision with the addition of more examples (accuracy: 0.97, precision: 0.75, sensitivity: 0.78), which was the best performance by GPT-4.

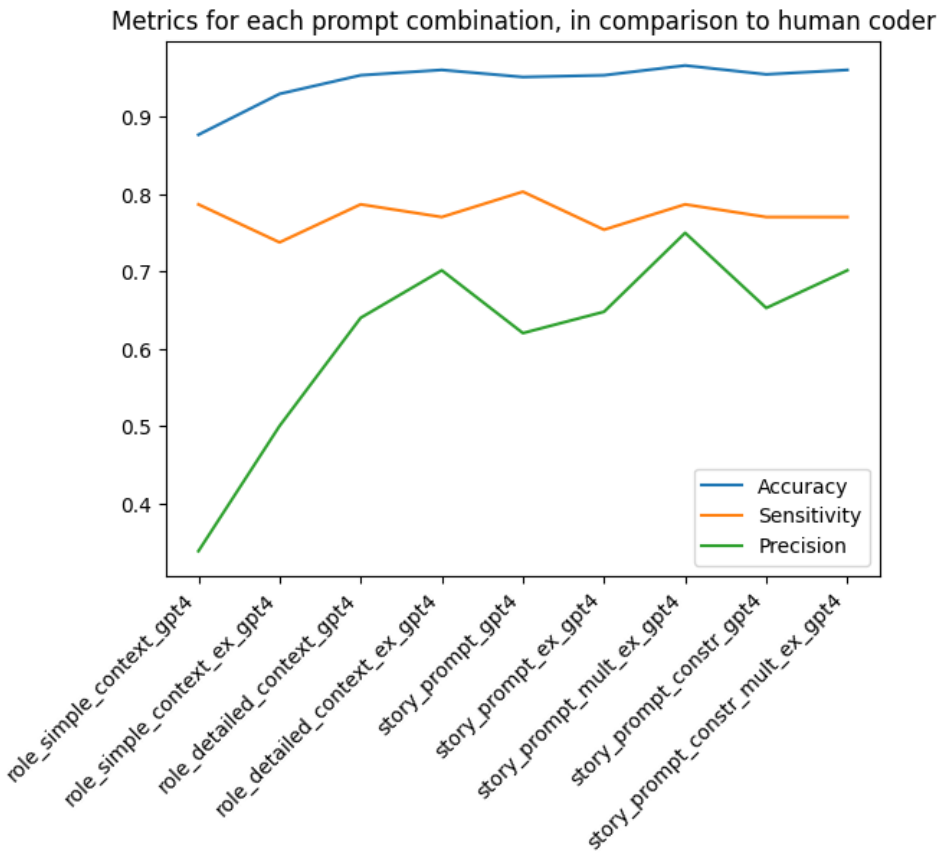

Figure 2. GPT-4 performance with different prompts (prompts roughly increase in complexity from left to right).

Overall, the highest scores from GPT-3 and GPT-4 were not very different from each other (accuracy and sensitivity within 0.1, precision within 0.5), but GPT-4 was able to produce relatively higher scores using simpler prompts and fewer examples.

Discussion

We found that, when considering their best performance across a variety of prompts, both GPT-3 and GPT-4 had an over 95% match with the human experts. If we focus only on matches of code assignment (rather than including cases where both did not assign a code), the numbers drop a little: out of the codes the models chose, around 70% (for GPT-3) or 75% (for GPT-4) were also chosen by the humans. Out of the codes the humans chose, 76% (for GPT-3) or 77% (for GPT-4) were also chosen by the models. Whether this is a high enough overlap to use these models for qualitative coding depends, at least in part, on the code assignments that did not match between the humans and models. Were these egregious errors on part of the models, or were they choices that could be justified by some slightly different chain of reasoning than that followed by the human coders? There is rarely perfect agreement between different human coders; minimum acceptable rates of agreement between coders (referred to as inter-rater reliability) is considered to be around 75% [15]. Considering this, next steps for this research may involve experts manually examining model responses and considering the percentage that appear to be justifiable. Another approach may involve creating prompts that ask the model to explain each code assignment, and then judging the explanations where the code assignments differ from the humans'.

While both GPT-3 and GPT-4 performed similarly given 5 examples and a detailed 'story' prompt, their performance differed significantly with simpler prompts. GPT-4 was able to achieve higher metrics for all the prompts, notably performing almost as well (within 10%) with only a single example as with 5. If examples are hard to generate for some task, then GPT-4 may be the better choice. However, if a few examples can be generated, it is important to note that (at time of writing) GPT-3 has a lower cost per token, and so could be the better choice for researchers wanting to preserve resources.

Limitations of this analysis include the assumption that all numbers mentioned in model responses correspond to a code; while this appeared to be a reasonable assumption after manually looking through the responses, responses might possibly contain numbers for other reasons (e.g. to refer to a sentence). Changing the prompts to explicitly specify a format for the response may be helpful in standardising responses and making such an assumption safer to make.

A second limitation is that, for speed of analysis, a dataset of only 30 comments was used. As the coding framework contains 29 codes, this means some codes may have appeared infrequently or not at all. Repeating the experiment with a larger dataset would help make the results more robust.

Further areas to explore include prompt tuning, an automated form of prompt optimisation that recent research has shown to lead to higher performances than manual optimisation for tasks such as classification [16]. Unfortunately, prompt tuning often requires large amounts of training data to optimise the prompt on [16]. A solution could lie in another recently expanding field, transfer learning, which can leverage training on one dataset to better performance on another [17]. Research on this may involve prompt-tuning a prompt with a previously-coded qualitative analysis dataset, and then trying to use that prompt to code a new dataset.

Overall, the results of this analysis suggest that GPT-3 and GPT-4 show promise for being used to automate qualitative coding tasks. Further research to standardise model responses and optimise the prompts used may be helpful in making these models a reliable tool for qualitative researchers.

References

1. Toleman, M., Rowling, S., Frederiks, A., Andersen, N., & Bhattacherjee, A. (2019, February 1). *Qualitative analysis*. Social Science Research Principles Methods and Practices Revised edition. Retrieved April 21, 2023, from https://usq.pressbooks.pub/socialscienceresearch/chapter/chapter-13-qualitative-analysis/
2. *Customer-led voice assistants*. PolyAI. (2023, April 20). Retrieved April 21, 2023, from https://poly.ai/
3. *Debuild: Code your web app in seconds.* Debuild. (n.d.). Retrieved April 21, 2023, from https://debuild.app/
4. *Powerful AI insights from your qualitative data*. Viable. (n.d.). Retrieved April 21, 2023, from https://www.askviable.com/
5. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011
6. Bouazizi, M., & Ohtsuki, T. (2019). Multi-class sentiment analysis on Twitter: Classification performance and challenges. *Big Data Mining and Analytics*, *2*(3), 181–194. https://doi.org/10.26599/bdma.2019.9020002
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., … Amodei, D. (2020). Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

8. OpenAI. (2023). GPT-4 Technical Report. *ArXiv [Cs.CL]*. Retrieved from http://arxiv.org/abs/2303.08774
9. Cabrera, L. Y., Fitz, N. S., & Reiner, P. B. (2014). Empirical support for the moral salience of the therapy-enhancement distinction in the debate over cognitive, affective and Social Enhancement. *Neuroethics*, *8*(3), 243–256. https://doi.org/10.1007/s12152-014-9223-2
10. Cabrera, L. Y., & Reiner, P. B. (2016). A novel sequential mixed-method technique for contrastive analysis of unscripted qualitative data. *Sociological Methods & Research*, *47*(3), 532–548. https://doi.org/10.1177/0049124116661575
11. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in Natural Language Processing. *ACM Computing Surveys*, *55*(9), 1–35. https://doi.org/10.1145/3560815
12. *Role prompting*. Learn Prompting. (n.d.). Retrieved April 21, 2023, from https://learnprompting.org/docs/basics/roles
13. Bose, R. (2023). GPT_qualitative_analysis (Version 1.0.0) [Computer software]. https://doi.org/10.5281/zenodo.1234
14. Zeugmann, T., Poupart, P., Kennedy, J., Jin, X., Han, J., Saitta, L., Sebag, M., Peters, J., Bagnell, J. A., Daelemans, W., Webb, G. I., Ting, K. M., Ting, K. M., Webb, G. I., Shirabad, J. S., Fürnkranz, J., Hüllermeier, E., Matwin, S., Sakakibara, Y., … Fürnkranz, J. (2011). Precision and recall. *Encyclopedia of Machine Learning*, 781–781. https://doi.org/10.1007/978-0-387-30164-8_652
15. Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: Assessment and reporting of Intercoder Reliability. *Human Communication Research*, *28*(4), 587–604. https://doi.org/10.1111/j.1468-2958.2002.tb00826.x

16. Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2021.emnlp-main.243
17. Su, Y., Wang, X., Qin, Y., Chan, C.-M., Lin, Y., Wang, H., Wen, K., Liu, Z., Li, P., Li, J., Hou, L., Sun, M., & Zhou, J. (2022). On transferability of prompt tuning for Natural Language Processing. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. https://doi.org/10.18653/v1/2022.naacl-main.290