Reflections on the Coding Ability of LLMs for Analyzing Market Research Surveys

Shi Zong
University of Waterloo, Waterloo Canada
s4zong@uwaterloo.ca

Santosh Kolagati
Nexxt Intelligence, Toronto, Canada
santosh@nexxt.in

Amit Chaudhary
Nexxt Intelligence, Toronto, Canada
amit@nexxt.in

Josh Seltzer
Nexxt Intelligence, Toronto, Canada
josh@nexxt.in

Jimmy Lin
University of Waterloo, Waterloo Canada
jimmylin@uwaterloo.ca

ABSTRACT

The remarkable success of large language models (LLMs) has drawn people's great interest in their deployment in specific domains and downstream applications. In this paper, we present the first systematic study of applying large language models (in our case, GPT-3.5 and GPT-4) for the automatic coding (multi-class classification) problem in market research. Our experimental results show that large language models could achieve a macro F1 score of over 0.5 for all our collected real-world market research datasets in a zero-shot setting. We also provide in-depth analyses of the errors made by the large language models. We hope this study sheds light on the lessons we learn and the open challenges large language models have when adapting to a specific market research domain.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**.

KEYWORDS

Coding, Market research surveys, Large language models

ACM Reference Format:
Shi Zong, Santosh Kolagati, Amit Chaudhary, Josh Seltzer, and Jimmy Lin. 2024. Reflections on the Coding Ability of LLMs for Analyzing Market Research Surveys. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3661362

1 INTRODUCTION

Large language models have achieved great success in recent years. With their great capability of analyzing a large amount data with few and even no annotated data provided (few-shot and zero-shot setting; [2]), and their generalization ability to different domains, large language models are becoming increasingly popular in many domains including finance [9], medical and healthcare [3], social sciences and psychology [4], and marketing research [1].

In this paper, we conduct the first systematic study that utilizes GPTs for the coding problem in market research. In market research, a coding task is to group the various responses collected from customers (also called *verbatims*) during the fieldwork process into a set of several key themes (or issues) – each issue then becomes a code. Coding is most often performed on open-ended questions to provide quantitative insights on users' feedback. A typical coding workflow involves the following steps: (1) based on the collected responses and defined research objectives, market researchers design a *codeframe* (also known as a *codebook*), and (2) then assign one or multiple codes to each response.

We note there have been some prior studies that try to apply large language models for a coding task [7, 10, 11]. For example, Xiao et al. [10] explore combining expert-drafted codebooks with GPT-3 for deductive coding. To the best of our knowledge, few studies have been done in a market research domain. Furthermore, our study is unique, as all the experiments are conducted on real-world private datasets from clients that we believe are not used in the training corpus of GPTs. As there is a debate on whether language models just memorize the training data, instead of making real inferences [5], our study could provide to some extent an objective evaluation to better understand the GPTs' effectiveness.

In this work, we also provide an in-depth analysis of the error cases that models make and discuss some open challenges. We hope our study will be a useful read for industry practitioners when applying large language models to their own domains.

2 CODING TASK IN MARKET RESEARCH

2.1 Problem Formulation

We have briefly introduced the coding task in Section 1: given a set of verbatims and an associated codeframe, our coding problem is to assign one or multiple codes from the codeframe to each verbatim. From this perspective, the coding problem can be treated as a multi-class classification task.

We now highlight the following two differences between our coding problem and the classic multi-class classification problem.

The codeframe is not unique. For the same set of users' responses and research goals, market researchers might develop different codeframes that are equally good, and it is thus hard to choose

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3661362

Dataset Name	# Codes	# Inputs	Avg. Len	Area	Anno. Method	Difficulty	Survey Question
Video Ad.	10	398	12.84	Ad. Test	Human	Normal	Can you explain why this ad speaks to you more?
Message Ad.	41	1200	11.41	Ad. Test	Human	Difficult	[*An ad. has been shown*] Thinking about what you just saw, what main idea do you think the concept was trying to express, convey or get across to you?
Video Service	25	403	18.68	Brand Study	AI w/ human review	Normal	Now, tell me anything that comes to your mind when you think of [*Brand Name*] Video Service! Please give us your honest opinions.
Sports Bet	32	441	12.74	Customer Exp.	Human	Normal	Why did you rate [*Sports Bet Name*]? What made you feel this way?
Chat Bot	13	953	9.45	User Experience	Human	Easy	Could you tell me why you felt that way when interacting with [*Chat Bot Name*]?
Water Filtration Feeling	21	342	14.73	Usage & Attitude	AI w/ human review	Difficult	[*images of water filtration pitcher have been shown*] What makes you think that this image best represents your ideal Water Filtration Pitcher & Filter System? Why?
Hair Care	7	173	15.14	Usage & Attitude	Human	Easy	Can you tell me what has changed regarding [*Hair Care*] products before and after COVID-19? That could include how/where/what/when/how often you buy, etc.

Table 1: An overview of our collected market research survey datasets.

which one is better. Inevitably, the selection of a codeframe would affect the accuracy of the downstream coding task. Specific to this study, we will skip discussions of the development of codeframes and only consider the situation where the model is given a reliable codeframe that has been reviewed by market researchers.

Allowance of flexibility (or variance) for the assigned codes to each verbatim. Even with a fixed codeframe, based on the market researchers' own interpretations and personal preferences, there might be different versions of the assigned codes that could all be treated as the ground truth. In other words, the assigned codes are arguable and inter-annotator agreement between different market researchers is not necessary or even desirable. We will look further into how this factor would affect the performance evaluation by using more fine-grained annotations in Section 3.3.

2.2 Datasets

In this work, we develop our own benchmark for evaluating the model coding quality. The benchmark consists of 7 datasets, collected from real survey questions conducted by the clients. These datasets are carefully chosen to make sure they cover a variety of fields, with verbatims of various lengths, and difficulties for coding.

Table 1 presents a summary of these datasets. In total, we collect 3,910 real-world users' responses, covering a variety of areas in market research, including advertisement testing, brand study, customer experience, user experience, and usage and attitude. Each dataset consists of a codeframe along with annotated codes, and the survey questions used for dataset collection.[1] We also ask the market researchers to rate the difficulty level to code the dataset as *easy*, *normal*, and *difficult*. Some samples from `Chat Bot` dataset are provided in Appendix A.1.

2.3 Experimental Settings

We experiment with GPT-3.5 and GPT-4 models.[2] OpenAI's GPT models are chosen as these are commonly used large language models with reasonable performance on a variety of tasks. As we only have a limited number of annotated verbatims for each code, and we reserve them all for evaluation, we do not fine-tune a large language model, nor perform any in-context learning (although it is an open challenge discussed in Section 4). Our experiments are conducted under a zero-shot learning setting. The temperature is set to 0.5 and the used prompt is in Appendix A.2.

As discussed in Section 2.1, although the assigned codes from the market researchers might not necessarily be the golden answers, we will still use them as ground truth labels for performance evaluation purposes. All the assigned codes in our benchmark datasets are either (1) directly provided by our clients or market researchers, or (2) first generated by using computational models and then reviewed by human. The methods for getting these codes are provided in Table 1. In Section 3.3, we further ask two of our market researchers to perform another round of annotations to explicitly distinguish between codes that must be and could be predicted by the model.

2.4 Experimental Results

2.4.1 Main Results. We calculate the macro F1 and micro F1 scores for all our datasets and report our main results in Table 2. All the generated labels are treated as predictions, regardless of their associated probabilities. We draw the following observations.

Perhaps unsurprisingly, both GPT-3.5 and GPT-4 do a decent job of classifying the input verbatims, even with no annotated data provided in the prompt. We observe a majority of datasets have a macro F1 ranging from 0.4 to 0.6 for GPT-3.5, and over 0.5 for GPT-4. We also observe for most of the datasets (except for `Hair Care` dataset), there is not a huge gap between macro F1 and micro F1 scores. It indicates that both GPT-3.5 and GPT-4 in general have rather balanced performance over most codes (in Table 3 we provide a breakdown of F1 scores for all the codes in `Chat Bot` dataset).

2.4.2 Results When Applying Thresholds on Confidence Scores. We also evaluate the models' F1 scores when varying the thresholds of the corresponding confidence scores generated for each label. For the generated tokens, we compute the joint probability as the sum of the log probabilities assigned to the gold answer(s). Specifically, if a predicted code c is composed of a sequence of tokens $t_1, ..., t_n$ with the corresponding probabilities $p_1, ..., p_n$, then the confidence score for this code c is $P(c) = \prod_{i=1}^{n} p_i$.

We set the following thresholds [0.1, 0.3, 0.5, 0.7, 0.9] and then calculate the macro F1 scores. Our experimental results are shown in Figure 1. We observe that setting a higher cutoff for the confidence scores does not always lead to an improvement in the macro F1 scores for classification. It seems to suggest that applying cutoffs on the confidence scores from GPTs might not be necessarily beneficial (see reported macro F1 scores with a threshold of 0.1).

[1] Due to privacy issues, the actual client names in the datasets are marked. A verbatim belongs to the `Uncategorized` category if there are no appropriate codes.
[2] Our experiments are done with `gpt-3.5-turbo-0125` and `gpt-4`.

Dataset Name	GPT-3.5							GPT-4							GPT-4 Re-Anno				
	Ma F1	Mi F1	Corr	Miss	More	Mix	Wrong	Ma F1	Mi F1	Corr	Miss	More	Mix	Wrong	Ma F1	Mi F1	Corr	Wrong	# Re-A
Video Ad.	0.57	0.66	0.46	0.02	0.22	0.03	0.27	0.60	0.67	0.48	0.02	0.29	0.01	0.21	0.69	0.73	0.57	0.14	167
Message Ad.	0.40	0.44	0.17	0.03	0.15	0.16	0.49	0.58	0.56	0.25	0.02	0.39	0.12	0.22	0.58	0.56	0.26	0.22	10
Video Service	0.48	0.54	0.13	0.13	0.15	0.21	0.37	0.61	0.62	0.15	0.06	0.43	0.23	0.13	0.61	0.60	0.15	0.16	290
Sports Bet	0.60	0.66	0.32	0.22	0.06	0.14	0.26	0.73	0.77	0.45	0.14	0.18	0.13	0.10	0.74	0.80	0.53	0.07	174
Chat Bot	0.31	0.32	0.20	0.03	0.03	0.02	0.72	0.57	0.58	0.35	0.04	0.28	0.05	0.27	–	–	–	–	–
Water Filt Feel	0.61	0.64	0.23	0.31	0.07	0.22	0.17	0.68	0.73	0.28	0.11	0.31	0.22	0.07	0.69	0.72	0.37	0.05	207
Hair Care	0.43	0.60	0.32	0.07	0.16	0.02	0.42	0.55	0.73	0.50	0.10	0.14	0.02	0.23	–	–	–	–	–

Table 2: Model performance in our benchmark. We report macro F1, micro F1 scores, and breakdown percentages (%) of the error types (details in Section 3.1.1). *GPT-4 Re-Anno* column reports results in Section 3.3, where the original golden codes are updated with *essential codes* defined in Section 3.3. *# Re-A* column refers to the number of verbatims re-annotated.

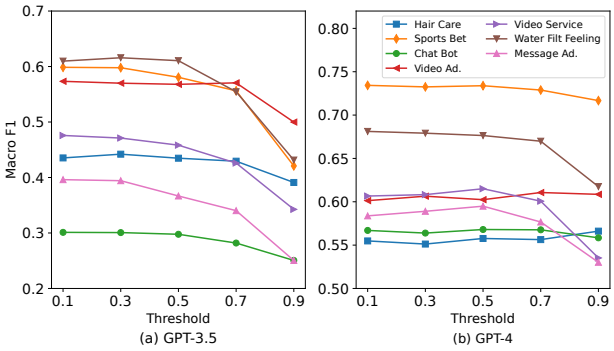

Figure 1: Macro F1 scores when varying the threshold for the confidence scores of predictions.

3 ANALYSIS

3.1 Types of Verbatims GPTs Fail to Classify

3.1.1 Methods. We conduct an error analysis by classifying the predictions of each verbatim into the following categories, based on the comparisons between the golden set and the prediction set: (1) The predictions match the ground truth exactly (`Correct`); (2) The predictions are correct but are missing some codes that are in the ground truth (`Miss`); (3) The predictions cover all the ground truth but have some incorrect predictions (`More`); (4) The predictions are a mixture of missing and incorrect codes (`Mixture`); and (5) The predictions are totally incorrect (`Wrong`).

3.1.2 Observations. We observe the following common issues where *both* GPT-3.5 and GPT-4 have difficulty across all datasets.

Short Responses. We observe that GPT models do not perform well on very short verbatims, including words with a clear meaning (short responses could also lead to the problem of lacking context, which we will discuss next). For example, some positive words such as "*good*", "*is the best*", and "*great*" are mapped to `Uncategorized` category. Among 760 verbatims that are in `Wrong` category, 58.9% of them have a length of less than or equal to 5 tokens.

Lack of Context. Another category that GPTs fail to make correct predictions is due to the lack of context. Specifically, we observe two situations: (1) the verbatim itself is not sufficient to make predictions, and (2) the same verbatim can be interpreted with different meanings under different contexts.

For (1), we notice that for some verbatims, especially with limited length, the models could not make valid inferences but rely on direct string matching. For example, for "*straightforward*", the prediction is `Simple/Straightforward Questions`, while the true label is `Simple/Easy Experience`. As another example, we have "*Everything is about Canada and interconnectedness*", the prediction is `Canada/Canadian`, while the ground truth is `Other`.

For (2), the same input may have specific meanings under a specific context and models could not distinguish them. For example, "*no*", "*na*", "*no reason*" and "*Don't know never used it*" could be coded as `Don't Know`, `Nothing`, `No Change` or `I Don't Use/Need It`, depending on the survey questions asked. Other examples include "*It was fine*", "*It is OK*", and "*Offer more booster*".

Readers might wonder whether such a problem could be solved by just incorporating corresponding survey question wording into the prompt, so as to provide context for GPTs to make predictions. In our experiments, we do not observe a significant gain from doing this. It might suggest that a more systematic way of helping GPT-based models perform inference under certain domains would be needed, such as introducing domain-specific tuning or incorporating a domain knowledge base.

Other Issues. In some datasets, we use SmartProbe [8] to ask a follow-up question to get further information about the original survey question from users. In practice, we notice that directly feeding all above as a whole confuses GPTs. We experiment with the method of first chunking the original sentences into several pieces, and this method only works for a portion of the verbatims.

We also observe that GPTs fail to distinguish between similar codes that have subtle differences. For example, the model codes "*It is a fun site*" as `Fun`, while the true code is `Generic Positive`.

3.2 Codes that GPTs are Bad at Predicting

To compare GPT-4's performance across different codes, we calculate Precision, Recall, and F1 scores for each code. Results are presented in Table 3. We observe that GPTs are not good at identifying generic themes (`generic positive` has the lowest Precision of 0.21 and lowest F1 score of 0.32). Other datasets also exhibit the same behavior (such as `generic negative` and `nothing`). We do not observe a clear pattern across datasets for the codes with high F1 scores for coding.

Although it is outside the scope of this paper, our finding that models fail to assign generic codes shall not be mixed with the issue that models also fail to generate specific codes. In pilot studies, we

No.	Code	P	R	F1	# Supp
0	simple/easy experience	0.58	0.73	0.65	190
1	like the format/chat/style	0.73	0.84	0.78	151
2	simple/straightforward questions	0.59	0.65	0.62	133
3	fast/short survey	0.47	0.62	0.54	124
4	fun/engaging/interesting	0.45	0.64	0.53	106
5	standard/same survey	0.50	0.61	0.55	106
6	good questions/express my feelings	0.39	0.62	0.48	95
7	difficult question/too many OEs	0.37	0.53	0.43	40
8	slow/long survey	0.65	0.72	0.68	36
9	generic positive	0.21	0.71	0.32	38
10	nothing	0.32	0.38	0.35	26
11	don't know	0.58	0.82	0.68	17
12	boring/not exciting	0.60	1.00	0.75	6

Table 3: Precision (P), Recall (R) and F1 scores (F1) for all the codes in `Chat Bot` dataset (using GPT-4). # Supp denotes the number of true samples for each class. Note that one verbatim can be mapped to more than one code.

Category	# Re-Anno	Relax Corr	# Codes Pr	Essent	Extra	Wrong
Miss	89	0.53	204	0.82	0.05	0.13
More	361	0.19	1116	0.45	0.13	0.41
Mixture	222	0.15	780	0.50	0.11	0.38
Wrong	176	0.39	248	0.19	0.20	0.61

Table 4: Breakdown counts of re-annotated verbatims for different error types (from GPT-4), with the percentage (%) among these verbatims that meet the relaxed correct requirement (Relax Corr). We also report the percentage (%) for the types of codes over the total predicted codes (# Codes Pr).

observe GPTs tend to use broad terms when asked to generate codeframes based on the given verbatims, which lack specificity.

3.3 Data Re-Annotation and Evaluation

3.3.1 Methods. We discussed in Section 2.1 that the assigned codes from market researchers should be treated as references rather than golden labels that cannot be revised. We now further examine the effect of this issue.

To do this, we ask two in-house market researchers to provide more fine-grained annotations for each input verbatim.[3] We divide the codes into two categories: (1) We say a code is *essential* if this code has to be predicted by a model. If the code is not picked up by the model, then a penalty should be applied. (2) *Extra* codes are any codes that would be nice to be coded but not a necessity, for example, codes that might be difficult for the model to pick up without the contextual information that a researcher would have.

As the re-annotation requires huge human effort, in collaboration with market researchers, we identify 848 verbatims that GPT-4 fails to predict exactly correct from the datasets with difficulty level *normal* or *difficult* (in Table 1) for re-annotation. Table 4 provides a breakdown of the number of verbatims that we re-annotate for different error types. Market researchers also suggest the following when merging two annotations: essential codes are those that at least one researcher marks as essential. The same applies to extra codes (excluding those that meet the essential codes criterion).

[3]We acknowledge the efforts from Jiahua Pan and Danica Deyto from the market research team of Nexxt Intelligence for this data re-annotation.

3.3.2 Results. We observe that two market researchers provide the same essential codes for 62.2% of the re-annotated data, while there are more divergences for the extra codes, mainly because of the varying degrees of additional interpretations researchers make. We then use essential codes as the new golden labels (verbatims that are not re-annotated remain unchanged) and recalculate GPT-4's classification performance in Table 1. Results show that in most cases macro F1 and micro F1 scores improve. Further analysis in Table 4 confirms that our prior evaluations in Section 2.4.1 are actually stricter. We note 33.1% of verbatims can meet the *relaxed correct* criterion: the models make no wrong predictions and no essential codes missing (thus can be treated as correct). We also report the distribution for the types of predicted codes in Table 4.

4 DISCUSSIONS AND OPEN QUESTIONS

In this work, we provide the first comprehensive study of GPTs for the zero-shot coding task in real market research survey data. We now discuss some open questions for future research.

(1) Combination of different trials. In real practice, we normally need to merge the results from different random trials to get more stable outputs, for example, only use the common assigned codes across runs. This method could effectively reduce the overcoding issue, while at the same time increasing the cases for the missing codes. A probabilistic or confidence-score based method is thus a promising direction to improve the final coding accuracy.

(2) In-context learning. As codeframes are different between datasets and we only have a very limited number of annotated verbatims for each code from the benchmark, we do not include any annotated samples in the prompt. In pilot studies, we tried to generate more examples for each code using GPTs and then include them in the prompt. However, we did not get promising results. As prior studies (for example, [6]) have shown that in-context learning would improve classification accuracy, it is worth exploring this direction under an annotation-lacking scenario.

A APPENDIX

A.1 Sample Verbatims from `Chat Bot` Dataset

1. "*A lot of surveys are repetative with the questions. I understand they have to make sure that the person is answering consistantly the same but it starts to frustrate me. Also they want your opinion on something that you say you know nothing about. I don't feel that I am giving honest opinions really.*"
2. "*Too many open ended questions, rather be able to tick boxes.*"
3. "*It took me a few moments to get the feel of scrolling. It also made me want to ask if you're AI is turned beyond the questions.*"
4. "*I don't like this kind of format. I have to wait longer for the next question to show up. Some pictures/photos were not able to show on the same page, so I had to scroll up and down to see it.*"

A.2 Used Prompt for Generating Coding Results

"*You are a professional market researcher who studies consumer and social trends. You classify survey responses into topics. Suppose you are given a list of topics with the sentiment associated with those topics. Classify the given survey response into one or multiple topics by listing out the topic names without the sentiment. If the response cannot be classified, return OUTLIER.*"

B MAIN PRESENTER'S BIO

Dr. Jimmy Lin is a Professor and the David R. Cheriton Chair in the David R. Cheriton School of Computer Science at the University of Waterloo. His area of research lies at the intersection between natural language processing and information retrieval.

REFERENCES

[1] James Brand, Ayelet Israeli, and Donald Ngwe. 2023. Using GPT for market research. *Harvard Business School Marketing Unit Working Paper No. 23-062* (Mar 2023).

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

[3] Tirth Dave, Sai Anirudh Athaluri, and Satyam Singh. 2023. ChatGPT in medicine: an overview of its applications, advantages, limitations, future prospects, and ethical considerations. *Frontiers in Artificial Intelligence* (2023).

[4] Dorottya Demszky, Diyi Yang, David S. Yeager, Christopher J. Bryan, Margarett Clapper, Susannah Chandhok, Johannes C. Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, Michaela Jones, Danielle Krettek-Cobb, Leslie Lai, Nirel JonesMitchell, Desmond C. Ong, Carol S. Dweck, James J. Gross, and James W. Pennebaker. 2023. Using large language models in psychology. *Nature Reviews Psychology* (2023).

[5] Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Investigating data contamination for pre-training language models. arXiv:2401.06059

[6] Youri Peskine, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso. 2023. Definitions matter: guiding GPT for multi-label classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

[7] Tim Rietz and Alexander Maedche. 2021. Cody: an AI-based system to semi-automate coding for qualitative research. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

[8] Josh Seltzer, Jiahua Pan, Kathy Cheng, Yuxiao Sun, Santosh Kolagati, Jimmy Lin, and Shi Zong. 2023. SmartProbe: a virtual moderator for market research surveys. arXiv:2305.08271

[9] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: a Large language model for finance. arXiv:2303.17564

[10] 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 *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*.

[11] He Zhang, Chuhao Wu, Jingyi Xie, ChanMin Kim, and John M. Carroll. 2023. QualiGPT: GPT as an easy-to-use tool for qualitative coding. arXiv:2310.07061