

From nCoder to ChatGPT: From Automated Coding to Refining Human Coding

Andres Felipe Zambrano, Xiner Liu, Amanda Barany, Ryan S. Baker, Juhan Kim and Nidhi Nasiar

University of Pennsylvania
azamb13@upenn.edu

Abstract. This paper investigates the potential of utilizing ChatGPT (GPT-4) as a tool for supporting coding processes for Quantitative Ethnography research. We compare the use of ChatGPT and nCoder, the most widely used automated coding tool in the QE community, on a dataset of press releases and public addresses delivered by governmental leaders from seven countries from late February to late March 2020. The study assesses the accuracy of the automated coding procedures between the two tools, and the role that ChatGPT's explanations of its coding decisions can play in improving the consistency and construct validity of human-generated codes. Results suggest that both ChatGPT and nCoder have advantages and disadvantages depending on the context, nature of the data, and researchers' goals. While nCoder is useful for straightforward coding schemes represented through regular expressions, ChatGPT can better capture a variety of language structures. ChatGPT's ability to provide explanations for its decisions can also help enhance construct validity, identify ambiguity in code definitions, and assist human coders in achieving high interrater reliability. Although we identify limitations of ChatGPT in coding constructs open to human interpretations and encompassing multiple concepts, we highlight opportunities and potential benefits provided by ChatGPT as a tool to support human researchers in their coding process.

Keywords: Automated Coding, ChatGPT, nCoder, Coding Process, Reliability.

1 Introduction

One of the key components of qualitative analysis of textual data is the process of defining themes and patterns and identifying where they appear in the data, to derive meaningful insights. This is especially true in quantitative ethnographic (QE) research, where theoretical and practical attention has been paid to the concept of *fairness* to ensure interpretations are consistently agreed upon by relevant stakeholders and that different manifestations of the same concept are comparable [1, 2]. This process of assigning valid codes in a reliable manner often proves both tedious and complex. Establishing inter-coder agreement can reduce the time and effort spent by allowing researchers to code sections of the dataset independently but remain time-consuming. While manual coding techniques remain popular in both QE and broader research [3], such approaches remain unsuitable for larger datasets, and the potential for human error remains.

Recent efforts to offload the burden of coding while maintaining or maximizing fairness have drawn on the affordances of computer science and natural language processing technologies. The development of automated coding tools such as nCoder [4], for example, has assisted researchers in coding data that would otherwise necessitate an extensive and time-consuming manual process. This freely accessible online tool (<https://app.n-coder.org/>), enables users to define codes and automate the coding process based on regular expressions provided by the researchers [5]. nCoder also allows for the assessment of coding agreement between and across humans and machines, and supports reviewing any disagreements to support a human analyst in refining the regular expressions used in these classifiers. Through this iterative process, the classifiers become more capable and proficient at discerning language structures that signify the presence of each code on each line. Once researchers deem the agreement between themselves and the detector sufficiently high, based on indicators such as Kappa, precision, recall, and Shaffer's rho [1, 6, 7], researchers can then generalize these classifiers to code the entire dataset. While nCoder has proved useful in QE research [3], there are still concerns that nCoder may be unable to capture all cases where a human would identify a code (recall), as regular expressions may not match human coders' capacity to consider contextual features or semantic nuance.

A similar iterative process could be conducted with other algorithms as well. There is an extensive history of machine learning methods being used to code textual data [8-10], with neural networks emerging as a particularly effective algorithm for this application in the last few years [11-13]. Very recent work has suggested that the use of neural networks may make possible to enhance the process of qualitative coding, through suggesting cases to code [14]. Other very recent work has demonstrated the potential of Large Language Models (LLM) such as ChatGPT [15] to perform automated coding of textual data [16]. LLMs can discern semantic relationships among words and concepts, capturing complex linguistic patterns that are difficult for human beings to identify rationally through regular expressions. Furthermore, ChatGPT's ability to interact with human beings enables a human analyst to query it to better understand the conceptual reasons for discrepancies between human and machine coding, potentially furnishing valuable insights that may enhance human coding practices. For this reason, LLM chatbots such as ChatGPT hold potential utility as a coding instrument that can facilitate higher-fidelity automated coding with less effort than the methods currently most-widely used in quantitative ethnography.

Given these possibilities, this work is an exploratory look at the potential of ChatGPT for supporting data coding processes in Quantitative Ethnography. We evaluate the accuracy and reliability of ChatGPT within the context of a dataset consisting of press releases and public addresses delivered by governmental leaders from seven countries during the period of late February to late March 2020 [17]. We investigate the consistency and construct validity of both ChatGPT and nCoder's automated coding procedures as compared to human-generated codes, and examine the role that ChatGPT's explanations of its coding decisions in supporting humans in revising a coding scheme.

2 Methods

2.1 Study Contexts

In this study, we examine the potential advantages of utilizing ChatGPT (GPT-4 model; see [15]) in the coding process. We first focus on improving the accuracy of the automated coding procedure, using nCoder as a reference for comparison. Then, we assess the consistency and construct validity among human-generated codes using ChatGPT's explanations of its coding decisions.

For this purpose, we employ a dataset that comprises transcripts of press releases and public addresses delivered by governmental leaders from seven countries from late February to late March 2020 [17]. The authors used a codebook incorporating seven categories: *Medical Positive*, *Medical Negative*, *Economic Positive*, *Economic Negative*, *Social Positive*, *Social Negative*, and *Political Positive*. One of the authors that originally used this dataset contributed to this analysis to reduce any potential risk of misinterpretation of the original codebook. The average inter-rater agreement, measured by Cohen's Kappa, was 0.94. For a more comprehensive discussion of the codebook, refer to [17].

We selected this dataset because the categorization of the content of leaders' discourse in the seven constructs defined by the authors is complex. In this dataset, similar information can be conveyed using a wide variety of vocabulary, and subtle differences between some constructs, such as social and political, or between positive and negative sentiment within the same construct, may prove difficult to distinguish reliably using automated classifiers. Therefore, we hypothesize that automated coding based on semantic similarity could be more effective than relying on exact wording for this dataset, making this a case where ChatGPT might demonstrate advantages relative to nCoder.

2.2 Coding Process with nCoder and ChatGPT

For both nCoder and ChatGPT, we randomly selected a training set of 100 lines from the original dataset. We tested both nCoder and ChatGPT's performance using an additional set of 100 unobserved lines from the dataset. We used the same training set and test set for both nCoder and ChatGPT.

Following the process outlined by Cai et al. [5], we used an iterative approach to craft regular expressions in nCoder that maximized agreement between the classifier and human coders for each code category. We first established a set of regular expressions based on the construct definitions provided by [17] and applied them to code the training set. We then analyzed any instances where nCoder and human coders disagreed and evaluated the effect of adding or deleting regular expressions on the overall kappa value. Next, we fine-tuned the regular expressions to enhance the agreement between nCoder and humans for the training set. Finally, we tested the optimized regular expressions on an additional set of 100 unobserved lines for each construct.

We adopted a similar approach for ChatGPT. For each construct, we provided ChatGPT with the construct name and original definition that was used by the authors [17]. We then addressed each disagreement between ChatGPT and human coders by requesting explanations from ChatGPT regarding its decisions. We incorporated these explanations to enhance the code definitions used in the prompts for ChatGPT, refining,

clarifying, or supplementing the definitions where needed, and including clarifying statements and examples of appropriate and inappropriate phrases to help define the constructs. It is important to note that the definitions given to ChatGPT were revised, but we did not subsequently have the human coders re-code the data based using this modified codebook (we discuss this possibility below). We also asked ChatGPT for suggestions for updating the definitions to gain insights about how we could reframe the definitions of each construct. Any definitions or supplements suggested by ChatGPT were reviewed and fine-tuned by human researchers. If a revised code definition negatively impacted the agreement between ChatGPT and human coders, we disregarded that new definition and proceeded to the next disagreement until we had reviewed the entire training set once. The flowchart of this process is presented in Figure 1.

An example of this process is shown in the results section. To avoid confusing ChatGPT due to information overload, we limited each definition to a maximum of five sentences. The coding process with ChatGPT was conducted in subsets of 25 lines, repeating the process four times, due to the maximum prompt length currently allowed by ChatGPT. After completing all coding, review, and refinement, we compared ChatGPT's performance to nCoder, using 100 additional unobserved lines from the dataset. This test set was the same for both coding strategies. Kappa, precision, recall, and Shaffer's rho [6, 7] were then calculated for the comparison between nCoder and ChatGPT using [18].

3 Results

3.1 Prompt creation

In this section, we illustrate an example of the ChatGPT prompt creation process using the code category labeled as *Social Positive*. According to [17], this construct is associated with concepts such as empathy, community, unity, and public education. Based on this original definition, we tasked ChatGPT with coding the training set of 100 randomly selected lines (in sets of 25 lines, as mentioned above). The first disagreement observed between ChatGPT and human coders occurs in Line 280, which states: "*Let us never despair.*" ChatGPT justified its choice of coding this line under the construct of *Social Positive* by explaining that:

This line conveys a message of hope and unity, encouraging people to remain hopeful and resilient in the face of challenges. It promotes a positive attitude and urges community members to support each other, which aligns with the concept of Social Positive.

Human coders, on the other hand, determined that this line should be coded as *Social Negative*, as they viewed it as a warning rather than a message of hope. After providing

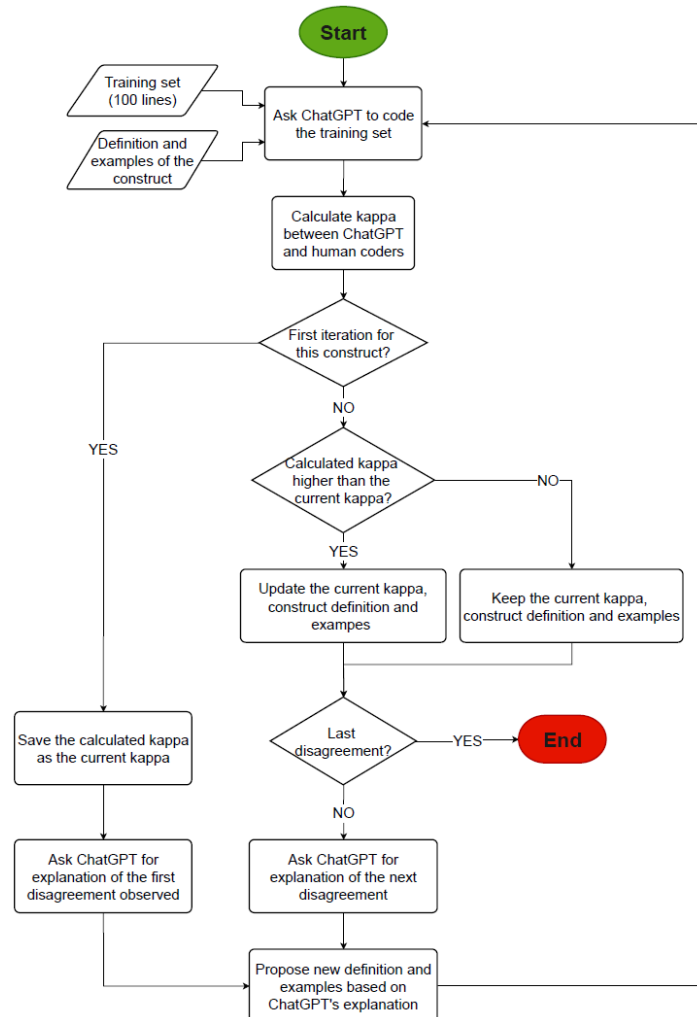


Fig. 1. Flowchart of the proposed process for developing the coding prompt of one construct using ChatGPT.

this clarification and requesting a revised definition of the code, ChatGPT responded:

I apologize for the misunderstanding. To modify the original definition of Social Positive, you may want to emphasize the explicit demonstration of beneficial impact, and provide clearer examples. Here's an updated definition:

*The concept of "social positive" refers to **actions, attitudes, or behaviors that explicitly demonstrate a beneficial impact on society** or positive perceptions of the community, through promoting activities or initiatives that directly support the community. Some examples of this concept are: volunteering, community-building events, promoting unity and collaboration, educational programs, or clear expressions of empathy and support for fellow community members.*

Based on this new definition, we ask chatGPT to code the training set again and continue to the next disagreement, which occurred in Line 71, which states: "*My dear fellow citizens.*" According to ChatGPT:

This line is a simple salutation or greeting addressed to the citizens. Although it demonstrates a level of empathy and connection, it does not explicitly promote activities or initiatives directly supporting the community, unity, collaboration, or educational programs as described in the revised definition of Social Positive.

Based on this explanation, we clarified to ChatGPT that greetings that show empathy and camaraderie should also be considered under the category *Social Positive*. After offering ChatGPT this clarification, ChatGPT developed the following definition of the code:

*The concept of "social positive" refers to actions, attitudes, or behaviors that explicitly demonstrate a beneficial impact on society or positive perceptions of the community, through promoting activities or initiatives that directly support the community **or by expressing empathy and connection**. Some examples of this concept are: volunteering, community-building events, promoting unity and collaboration, educational programs, clear expressions of empathy and support for fellow community members, **or heartfelt greetings that display a sense of camaraderie and connection**.*

This process was replicated for all disagreements in the training set of each code category. In nCoder, code and coding refinement requires that researchers examine inconsistencies, diagnose validity or consistency issues independently from the tool, and alter regular expressions accordingly. Through the use of ChatGPT, the process becomes more interactive and embedded in the study context (code and data features), serving as a scaffolded support for identifying and addressing issues related to construct validity (code definitions) or consistency (code applications).

3.2 Agreement metrics

Table 1 shows the degree of agreement between machine and human codes. During training, nCoder shows higher agreement with human coders (average Kappa = 0.77) than ChatGPT (average Kappa = 0.54), which is likely due to the use of regular expressions that explicitly define the words and language structures considered by humans when categorizing the examples. For the test data, nCoder also shows a higher level of agreement with human coders (Kappa = 0.53) compared to ChatGPT (Kappa = 0.46). In terms of precision, which assesses the machine's ability to accurately identify the presence of a construct in each line while minimizing false positives, nCoder (Precision = 0.79) outperforms ChatGPT (Precision = 0.52). However, for recall, which evaluates the proportion of true positive categorizations made by the machine over all the actual positives coded by humans, ChatGPT (Recall = 0.8) surpasses nCoder (Recall = 0.6).

In looking at these differences, one of the key factors is whether specific words are sufficient to recognize a category. Take, for example, the *Medical Positive* code, which exhibits a precision of 0.88 but a recall of 0.54 for nCoder. For this code, the presence of words such as "*vaccines*" or "*hospitals*" allows nCoder to easily recognize this construct. Consequently, each time a new unobserved line contains either of these

words, if there is no language structure indicating a reduction or absence, the nCoder classifier can categorize it as *Medical Positive* with a low risk of error. However, nCoder may struggle to generalize to all the names of medical institutions or devices that were not observed during training. In other words, although nCoder’s precision can be relatively high because it received specific examples for each code category, it may face low recall when generalized to broader data, since regular expressions require exact words. As Cai et al. [5] note, it is impossible to provide nCoder with all possible examples.

Table 1. Training and testing agreement between human and machine coding.

Code	Machine Coder	Train Kappa	Kappa	Test		
				Precision	Recall	Shaffer’s rho
Medical/ Positive	nCoder	0.85	0.5	0.88	0.54	1
	ChatGPT	0.59	0.47	0.59	0.69	1
Medical/ Negative	nCoder	0.73	0.39	0.65	0.5	1
	ChatGPT	0.54	0.50	0.60	0.69	1
Economic/ Positive	nCoder	0.83	0.64	0.76	0.7	0.5
	ChatGPT	0.87	0.83	0.90	0.87	0.04
Economic/ Negative	nCoder	0.82	0.47	1	0.43	1
	ChatGPT	0.56	0.51	0.42	0.89	1
Social/ Positive	nCoder	0.78	0.66	0.83	0.8	0.47
	ChatGPT	0.46	0.38	0.5	0.81	1
Social/ Negative	nCoder	0.86	0.67	0.82	0.69	0.46
	ChatGPT	0.31	0.2	0.27	0.83	1
Political/ Positive	nCoder	0.55	0.41	0.58	0.54	1
	ChatGPT	0.46	0.32	0.36	0.84	1
Average	nCoder	0.77	0.53	0.79	0.60	0.78
	ChatGPT	0.54	0.46	0.52	0.80	0.86

As ChatGPT is a pre-trained LLM, recall issues can be mitigated by the vocabulary and semantic structures that ChatGPT has already learned. For example, for the *Medical Positive* code, ChatGPT has previously learned a broad range of medical vocabulary, under the semantic field of medicine or health, which can enable it to recognize organizations, devices, or other concepts that are not observed during the training process but that would be related to the code category. This capability of ChatGPT is also evident in the examples provided when defining the prompt of the category *Social Positive* in Section 3.1. ChatGPT identified a greeting even without being given an example or the explicit language structure of a greeting in the prompt.

In general, ChatGPT can offer high performance in code categories where the range of possible interpretations is limited, and the semantic field is concrete. For instance, for the construct Economic Positive, which the authors originally defined as any concept related to commerce, trade, and marketplace, ChatGPT achieves precision and recall greater than 0.85 and a Shaffer's rho of 0.04, indicating it can accurately code the entire dataset for this construct according to the standards suggested by [1, 6]. However, for constructs that are more open to interpretation or less thematically discrete, ChatGPT's pre-established semantic connections may result in overgeneralization, leading to the incorrect categorization of some lines as belonging to those constructs. This issue is apparent in the lower precision shown by ChatGPT for code categories about social or political actions compared to nCoder. For example, Line 1 of Emmanuel Macron's speech states:

“Thursday evening, I spoke to you to discuss the health crisis that our country is going through. Until then, the epidemic may have been a distant idea for some, it has become an immediate, pressing reality.”

This line was categorized as *Medical Negative* by human coders, who identified the concept of a health crisis in this line exclusively as a medical issue, recognizing that its social and economic consequences are not mentioned within the line. However, ChatGPT interpreted this line as belonging to both *Medical* and *Social Negative* categories, providing the following rationale for the *Social Negative* construct:

This line is coded as Social Negative because it refers to the negative consequences of the ongoing crisis, which likely includes social issues such as close contact restrictions, lockdowns, or other related social issues. The statement implies that the implemented measures are a response to the crisis, and the deepening of the crisis suggests that the population is facing challenges and difficulties.

Even after instructing ChatGPT not to categorize lines that exclusively refer to the health issues of the pandemic as Social or Economic Negative, its semantic structures continue to classify such lines as Social Negative. This overgeneralization is the main reason for the relatively low precision ChatGPT achieves.

3.3 ChatGPT explanations for consistency checks and construct validity

When we consider nCoder and ChatGPT's properties for replicating human coding on this task, it is worth noting that human coding can also be fallible. A low precision or recall can indicate that ChatGPT or nCoder perform poorly in the coding task, but it also can represent fuzziness in the construct being coded by humans. Disagreements between human coders and ChatGPT or nCoder can arise from inconsistencies or idiosyncrasies in how a construct is conceived or coded. Therefore, although the semantic structures developed by ChatGPT as described in the previous sections may be imperfect at capturing the codes, the tool's capacity to offer explanations grounded in those previous structures can be valuable for detecting possible inconsistencies and idiosyncrasies in human understanding. This provides an additional perspective for some decisions in the coding process, supporting the humans who make decisions about codes in a critical examination of their interpretations and assumptions.

ChatGPT's grounded and detailed explanations offer an opportunity for human coders to identify and critically reexamine any inconsistencies that may exist between any code they apply and their working understandings of the code categories. ChatGPT can support human coders to think through their own rationales (and any underlying assumptions) while coding, especially when a code definition is more subjective or leaves room for interpretation. It can offer peer-like support (a thought-partner) to help human coders be aware of their own decision process in a more systematic fashion. An exploration of repetitive inconsistencies may also further highlight issues in construct validity, when the coding scheme is not clearly defined or fails to capture the underlying construct [19-21].

These types of inconsistencies can occur in almost any coding scheme attempting to capture complex, rich constructs. Though we explore them in [17] work, it is important to note that these inconsistencies only manifest in a small proportion of the data codes. We highlight them with the purpose of showing the potential benefits of using ChatGPT as an additional tool for revising the inconsistencies that can be present in the human coding process -- even for coding schemes and codes that are generally high-quality -- and for enhancing construct validity and interrater reliability.

We can illustrate this process by studying the construct with the lowest agreement between humans and ChatGPT, *Social Negative*. This code category was originally defined to include "nationalism, xenophobia, racism, lockdown, and closed schools" [17], a broad range of topics. However, on two occasions, human coders agreed not to categorize lines related to school closures and lockdowns under the *Social Negative* construct. For example, Line 136 from the speech by Yoshihide Suga states:

"We ask private companies to take into consideration the situation of employees with children. In particular, the closure of schools will be a very significant burden for micro-, small- and medium-sized business operators, who already routinely face a shortage of workers, as well as medical professionals and nursing and childcare workers."

Although human coders did not consider this to be within *Social Negative* category, ChatGPT categorized this line as *Social Negative* arguing that:

It [Line 136] addresses the negative consequences faced by employees with children due to the closure of schools, which is a social issue. The line highlights the challenges and difficulties that micro-, small-, and medium-sized business operators, medical professionals, and childcare workers experience as a result of this situation.

In this example, ChatGPT clearly associated the school closure mentioned in this line with the original definition of the code category *Social Negative*. Similarly, for Line 196 in the speech by Cyril Ramaphosa, which mentioned:

"As a consequence, the National Coronavirus Command Council has decided to enforce a nation-wide lockdown for 21 days with effect from midnight on Thursday 26 March,"

ChatGPT coded this as *Social Negative* due to the inclusion of the concept of lockdown, disagreeing with the human coders, who coded this as *Political Positive*.

Although human coders included lockdown as one of the examples of the *Social Negative* code, they agreed not to code Line 196 as *Social Negative*. The explanation provided by ChatGPT was:

This line is associated with the concept of "Social Negative" because it discusses a nation-wide lockdown, which is a direct response to a social issue (the pandemic). Lockdowns lead to negative consequences and difficulties faced by the population, such as restrictions on movement, closures of businesses, and potential economic repercussions.

This explanation reveals again that ChatGPT was not fully successful in differentiating between medical and social issues, which aligns with COVID-19 emerging in 2020 as a global medical concern with a high level of social impact. As mentioned earlier, this limitation is a primary cause of the low precision (but high recall) exhibited by ChatGPT when identifying the *Social Positive* and *Social Negative* constructs. Despite this limitation, ChatGPT proved helpful, detecting that human coders were not classifying this line adhering to the predetermined definition of *Social Negative*.

Furthermore, ChatGPT also provided the opportunity to continue consideration of these disagreements. For instance, after reviewing its explanation for the coding of Line 196, we clarified to ChatGPT that, under the original coding scheme, *the pandemic is considered a medical issue rather than a social issue*. ChatGPT's response to this explanation was:

You're right that the pandemic is primarily a medical issue. However, the consequences of the pandemic, such as lockdowns and restrictions, have significant social impacts as well. These social impacts include disruptions to daily life, economic challenges, and mental health issues, among others. In this context, line 196 touches upon a "social negative" aspect since it refers to the implementation of a lockdown, which affects the social life and well-being of the population.

This response serves as evidence that ChatGPT can present valid justifications for its coding decisions and maintain consistency in those decisions, providing similar argumentation to the one previously observed in the example shown in Section 3.2 for Line 1. While it may not necessarily persuade human researchers to modify their own coding schemes, it can prompt them to engage in more critical reflection, in this case on the discreteness of coding categories.

In addition to discrepancies between construct definitions and coding, ChatGPT can also assist in uncovering inconsistencies between two lines that have similar semantic meanings but were coded differently by humans. For instance, human coders agreed to categorize the following lines in the speech by Cyril Ramaphosa as *Social Positive* but not as *Medical Positive*:

- *Wash hands frequently with hand sanitisers or soap and water for at least 20 seconds;*
- *Cover our nose and mouth when coughing and sneezing with tissue or flexed elbow;*
- *Avoid close contact with anyone with cold or flu-like symptoms.*

This categorization suggests that, for these human coders, all instructions aimed at citizens for reducing contagion risk, without any negative connotation in the message, should be coded exclusively as *Social Positive*. However, Line 174 in the speech by Cyril Ramaphosa, which states, "*We reiterate that the most effective way to prevent infection is through basic changes in individual behavior and hygiene,*" was coded as both *Medical* and *Social* by human coders. ChatGPT therefore argues that this set of lines should be categorized consistently. It categorizes all the abovementioned lines as *Medical Positive* and provided the following rationale for each of them regarding the *Social Negative* construct:

This line provides a recommendation for a hygienic practice to help prevent the spread of illness. It does not explicitly discuss any of the social issues mentioned in the definition of "social negative" (nationalism, xenophobia, racism, stockpiling, basic necessities unsatisfied, lockdowns, and closed schools) nor does it highlight the negative aspect of these issues. Therefore, it was coded as 0, meaning it is not associated with the concept of Social Negative.

ChatGPT's primary contribution is not to correct human coding on these lines, but to emphasize that all of these lines should be categorized consistently. Consequently, ChatGPT may also be employed as an additional means of reviewing the application of codes across the dataset, reducing potential inconsistencies even after agreement has been reached between the two human coders.

However, ChatGPT's reasoning is still imperfect in many cases. For instance, Line 163 in Cyril Ramaphosa's speech - "*The response of the South African people to this crisis has been remarkable. Millions of our people have understood the gravity of the situation*" - was coded by ChatGPT as *Social Negative* rather than *Social Positive*, with the following rationale:

It [Line 163] refers to the crisis and the gravity of the situation being faced by the South African people, highlighting the challenges and difficulties they are experiencing due to the crisis. While it doesn't mention a specific social issue, it implies that there are negative consequences faced by the population in a broader context.

This example illustrates that ChatGPT can sometimes overlook valuable nuances within lines, such as the word "*remarkable*", which gives this line a positive connotation that caused humans to categorize it under the *Social Positive* construct instead of *Social Negative*. Therefore, although ChatGPT's explanations can be valuable for detecting possible inconsistencies in human coding, it can also neglect relevant information, focusing on other words or concepts that it considers more relevant, leading to errors in ChatGPT's coding. Despite these limitations, the examples discussed in this section demonstrate how ChatGPT has the potential to support human coders by providing explanations and warnings about possible issues in their codes and/or coding schemes. These explanations can assist human coders in becoming more aware of their own decision-making process and becoming more systematic. Repetitive inconsistencies may also highlight issues with construct validity, alerting human researchers to aspects of the coding scheme that are not clearly defined or fail to capture the intended construct.

4 Discussion & Conclusions

In this paper, we have discussed how ChatGPT can be a useful tool for improving the coding process. We first explored the use of ChatGPT as a tool for automated coding. We compared ChatGPT with nCoder, the current most common tool for automated coding within the quantitative ethnography community, in terms of each approach's performance for unseen data. We observed that the tools had different affordances and constraints, suggesting that the decision to use one or the other depends on the context and the nature of the data, as well as the researcher's goals. The use of the regular expressions in nCoder led to higher precision but lower recall, as it could capture exactly what the human coder intended but often could not capture ways to phrase the construct that were not seen in the training data. By contrast, ChatGPT can achieve better recall because it can capture the same semantic meaning represented with different words and phrases, but in some cases, it can miss nuances or misinterpret information, leading to lower precision for some constructs.

Based on these results, nCoder might be more helpful than ChatGPT in contexts and codebooks in which a code can be represented through a small set of regular expressions. However, even in those situations, ChatGPT's prompts can be fine-tuned to recognize specific language structures. Although we did not explore the use of regular expressions in ChatGPT's prompts, if the definitions on those prompts specifically include the set of regular expressions that ChatGPT should use during the coding process, it might be able to emulate nCoder functioning. Nevertheless, for this type of more concrete constructs, the semantic structures previously learned by ChatGPT can also be enough for reaching a higher agreement. For example, the agreement between ChatGPT and human coders was the highest for the construct *Economic Positive*, outperforming nCoder without needing to include any regular expression in the prompt.

On the other hand, ChatGPT's agreement with human coders diminished for constructs with less concrete definitions and constructs open to human interpretation. Consider, for example, the *Social Positive* and *Social Negative* constructs, for which ChatGPT had the lowest agreement. Firstly, these two codes are prone to subjective interpretation regarding what constitutes positivity or negativity; this also holds true for social, political, and economic aspects to a lesser extent. In addition, the *Social* constructs emerged in inductive coding as a broader, catch-all code that covers a wider spectrum of concepts from more diverse semantic domains, such as healthcare guidelines, education, empathy, lockdowns, and xenophobia, which may not fall under a single discrete thematic construct in many cases. In fact, ChatGPT was more likely to relate health care instructions with medical concepts (*Medical Positive*) than with community empathy or unity (*Social Positive*). Therefore, we hypothesize that ChatGPT may be most successful with constructs that are mutually exclusive and collectively exhaustive [19]. However, this does not indicate that ChatGPT does not provide valuable insights about constructs that are less thematically discrete.

As previously mentioned, the indicators presented in this study evaluate the (dis-)agreement between human and machine coding, but they do not necessarily establish which one is accurate or erroneous. Consequently, a reduced Kappa, precision, or recall for ChatGPT (or nCoder) only indicates that the automated method has a different understanding of the code definitions provided by the human researchers. Therefore, even with constructs where the agreement between ChatGPT and human beings is low, the main benefit offered by ChatGPT, as we demonstrate in

section 3, is that ChatGPT can explain its reasoning. These explanations help the human using ChatGPT to gain more insight into these disagreements and review whether any inconsistencies in human coding have gone undetected during the earlier interrater reliability checking with another human. Ultimately, even if ChatGPT's insight into its own decisions is not fully correct, human coders may find the explanations it provides valuable for supporting reflexivity around the coding process.

Moreover, ChatGPT's ability to provide explanations can also be valuable for enhancing construct validity by identifying potential ambiguity in the definition of a coding category and illustrating when and how a code may not appropriately capture the phenomenon/construct. Human-developed code definitions, even after substantial attempts to reach clarity, still often leave room for interpretation. ChatGPT can help to tackle this issue by providing elaborate explanations for the reasons for each of its codes, enabling humans to develop richer, more contextualized, and more concrete code definitions. This may make it easier to get high inter-rater reliability amongst human coders, and eventually to achieve better human-computer agreement. This approach may also prove useful for checking and improving consistency across longer human-coded datasets, as coders often refine construct definitions and code applications over time (drift, see [20]). If we train ChatGPT with definitions and examples of constructs at the end of the coding process, it could help coders detect and resolve inconsistencies in earlier coding stages without needing us to check or recode the entire dataset. Future work might also examine the potential for ChatGPT, and LLMs in general, to support thematic analysis of qualitative data as part of the inductive development of codes and coding schemes. Each of these directions for future research (construct validity, drift, and inductive code development) could yield valuable insights into the potential of ChatGPT as a responsive resource for qualitative data analysis beyond its benefits for automation of the coding process.

This paper has explored the potential of ChatGPT as a tool for supporting the coding process. ChatGPT has specific affordances and limitations for coding certain constructs in specific contexts as compared to other automated tools such as nCoder. Beyond this, ChatGPT can also provide justifications for coding decisions that can help researchers to develop richer and more concrete or complete construct definitions. This feature also offers an extra layer of verification of the accuracy and consistency of codes and constructs, ultimately improving the current approach of assessing validity through assessing inter-rater reliability. The application of language models such as ChatGPT in qualitative research is quite new. Further exploration of their potential is necessary, particularly in terms of their ability to enhance fairness and reliability in coding and analysis. Therefore, we encourage our colleagues to continue exploring the possibilities and constraints of large language models to facilitate their effective utilization in research. Our hope is that this work opens new avenues in quantitative ethnography to explore the potential of GPT and other language models to refine the coding process and enhance our understanding of complex data.

Acknowledgments. We extend our sincere gratitude to the members of the 2020 and 2021 ICQE data challenges who collected, processed, and coded the dataset used in this study. Special thanks go to Michael Phillips, Anthony J. Taiki Kawakubo, and Jun Oshima for their vital contributions to coding, project conceptualization, and research dissemination. We would also like to acknowledge the assistance provided by ChatGPT in improving the grammar and writing clarity of our initial drafts. All content was thoroughly reviewed and edited by human authors prior to submission. Andres Felipe Zambrano thanks the Ministerio de Ciencia, Tecnología e Innovación

and the Fulbright-Colombia commission for supporting his doctoral studies through the Fulbright-MinCiencias 2022 scholarship.

References

1. Shaffer, D. W., & Ruis, A. R. How we code. In *Advances in Quantitative Ethnography: Second International Conference, ICQE 2020, Malibu, CA, USA, February 1-3, 2021, Proceedings 2*. pp. 62-77. Springer International Publishing (2021).
2. Kim, Y. J., Choi, J.: Expanding fairness in game-based assessment with quantitative ethnography. In: *Damşa, C., Barany, A. (eds.) Fourth International Conference on Quantitative Ethnography: Conference Proceedings Supplement*, pp. 49-54. The International Society for Quantitative Ethnography (ICQE) (2023).
3. Zörgő, S., Peters, G.J.Y., Porter, C., Moraes, M., Donegan, S., Eagan, B. Methodology in the Mirror: a Living, Systematic Review of Works in Quantitative Ethnography. In: *Advances in Quantitative Ethnography. Communications in Computer and Information Science, Vol 1522*. Eds. Wasson B and Zörgő S., pp. 144-159. Switzerland: Springer Nature (2022).
4. Marquart, C.L., Swiecki, Z., Eagan, B., Shaffer, D.W.: *nCodeR (Version 0.1.2)* (2018).
5. Cai, Z., Siebert-Evenstone, A., Eagan, B., Shaffer, D. W., Hu, X., & Graesser, A. C. *nCoder+*: a semantic tool for improving recall of nCoder coding. In *Advances in Quantitative Ethnography: First International Conference, ICQE 2019, Madison, WI, USA, October 20–22, 2019, Proceedings 1*. pp. 41-54. Springer International Publishing (2019).
6. Shaffer, D. W. (2017). *Quantitative ethnography*. Cathcart Press (2019).
7. Eagan, B., Brohinsky, J., Wang, J., & Shaffer, D. W. Testing the reliability of inter-rater reliability. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. pp. 454-461 (2020).
8. Grimmer, J., Roberts, M. E., & Stewart, B. M. *Text as data: A new framework for machine learning and the social sciences*. Princeton University Press (2022).
9. Chang, T., DeJonckheere, M., Vydiswaran, V. V., Li, J., Buis, L. R., & Guetterman, T. C. Accelerating mixed methods research with natural language processing of big text data. *Journal of Mixed Methods Research*, 15(3), 398-412 (2021).
10. González Canché, M. S. Latent Code Identification (LACOID): A Machine Learning-Based Integrative Framework [and Open-Source Software] to Classify Big Textual Data, Rebuild Contextualized/Unaltered Meanings, and Avoid Aggregation Bias. *International Journal of Qualitative Methods* 22 (2023).
11. Yao, L., Mao, C., & Luo, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* 33(1). pp. 7370-7377 (2019).
12. Wang, J., Wang, Z., Zhang, D., & Yan, J. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (2017).
13. Sagha, H., Cummins, N., & Schuller, B. Stacked denoising autoencoders for sentiment analysis: a review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7(5). e1212 (2017).
14. Choi, J., Ruis, A. R., Cai, Z., Eagan, B., & Shaffer, D. W. Does Active Learning Reduce Human Coding?: A Systematic Comparison of Neural Network with nCoder. In *Advances in Quantitative Ethnography: 4th International Conference, ICQE 2022, Copenhagen, Denmark, October 15–19, 2022, Proceedings*. pp. 30-42. Springer Nature Switzerland (2023).
15. OpenAI. *GPT-4 Technical Report* (2023). ArXiv, abs/2303.08774.
16. Gilardi, F., Alizadeh, M., & Kubli, M. Chatgpt outperforms crowd-workers for text-annotation tasks. (2023). arXiv preprint arXiv:2303.15056.
17. Barany, A., Philips, M., Kawakubo, A. J. T., & Oshima, J. Choosing units of analysis in temporal discourse. In *International Conference on Quantitative Ethnography*. pp. 80-94. Springer International Publishing (2021).

18. Eagan, B., Rogers, B., Pozen, R., Marquart, C., & Shaffer, D.W. rhoR: Rho for inter rater reliability (2016). Retrieved from <https://app.calrho.org/>
19. Saldaña, J. The coding manual for qualitative researchers. The coding manual for qualitative researchers, 1-440 (2021).
20. Miles, M. B., & Huberman, A. M. Qualitative data analysis: An expanded sourcebook. Sage (1994).
21. Charmaz, K. Constructing grounded theory: A practical guide through qualitative analysis. Sage (2006).