A Comparative Study of Deductive Coding Methods for Enhancing Urban System Management with Large Language Models

Joshua Rodriguez[a], **Steven A. Conrad**[a,*] **and Om Sanan**[b]

[a]Department of Systems Engineering, Colorado State University, USA
[b]Scarsdale High School, Scarsdale, New York, USA
ORCID (Steven A. Conrad): https://orcid.org/0000-0002-2577-2926

Abstract. Urban systems are managed using complex textual documentation that need coding and analysis to set requirements and evaluate built environment performance. Yet, qualitative coding and assessment face challenges like resource limitations and bias, accuracy, and consistency between human evaluators. Errors in coding may result in omissions or false premises that carry into broader urban system design. This paper contributes to the study of applying large language models (LLM) to qualitative coding activities to reduce resource requirements while maintaining comparable reliability to humans. Here we report the application of LLMs to deductively code 10 case documents on the presence of 17 digital twin characteristics for the management of urban systems. We utilize two prompting methods to compare the semantic processing of LLMs with human coding efforts: whole text analysis and text chunk analysis using OpenAI's GPT-3.5 and 4 models. We found similar trends of internal variability between methods and results indicate that LLMs may perform on par with human coders when initialized with specific deductive coding contexts. Text chunking using GPT-4 resulted in similarity with human coders in terms of percent agreement, recall and accuracy. Adding GPT-4 chunked text results as an additional evaluator showed agreement levels of 89% among all reviewers, in contrast, analyzing entire texts showed lower agreement at 82%. Analyzing chunked text with GPT-4 also resulted in a 14% false negative (FN) rate and 6% false positive (FP) rate, while the same model with full text produced a 44% FN rate and 2% FP rate. These FP and FN rates resulted in a recall of 84%, and accuracy of 81% for chunked text and a lower recall of 49%, and reduced accuracy of 54% for whole text.

1 Introduction

The management of Urban Systems – the interconnection of built environment, the natural environment, and society – involves complex textual documentation that requires persistent review to derive requirements and performance standards. Architects and planners review codes and regulations to ensure construction projects align with current standards. Water managers review environmental protection rules and scientific reports to ensure water quality and policymakers review public consultation documents, policy proposals, and research studies to assess urban development projects. Each of these roles necessitates humans to read, digest, and semantically process hundreds of thousands of words to deduce relationships and presence of content for a specific theme or premise. Traditional methods for this qualitative deduction are time consuming, and the knowledge and control of the process varies by age, experience, and memory of the human coder [28, 29]. Human coders also present biases, variations in accuracy and consistency[9]. Missteps in the coding process could lead to omissions or inaccuracies, potentially skewing urban system designs and policy implementations. Moreover substantially large data sets may preclude analysis and remain unexamined [17].

In this paper we explore the application of Large Language Models (LLMs) to the address the resource and consistency challenges of humans coding complex scientific documents. We utilize the context of managing urban systems as a test case. We examine the premise that LLMs could reduce the burden of coding while maintaining reliability comparable to human coders. Specifically, we investigate the use of OpenAI's GPT-3.5 and GPT-4 models to deductively code digital twin characteristics from literature on urban water systems. We examine two semantic processing methods: whole text analysis and chunking and compare the performance to human evaluators.

The contribution of this paper is two fold:

1) We contribute to the ongoing discourse on automating analytical tasks in urban system management by integrating LLMs into the deductive coding process of complex textual documentation.

2) We assess and propose a method for prompting LLMs for deductive coding that achieves higher performance and similarity in outcomes to humans, presenting a viable option for including LLMs as an additional document evaluator.

2 Background

Pretrained LLMs show potential to automate or augment various Natural Language Processing workflows, one of the most promising being text classification and information retrieval from unstructured, text-based files [4, 22]. Despite their promise, LLMs carry risks associated with utilization in classification processes as they do not incorporate user goals, but rather focus on next-word prediction leading to propagated biases and incoherent texts [23, 1]. In addition, LLMs are subject to hallucinations or fabricating information due to false or inadequate training data and lacking knowledge recall processes [14]. To combat this, Ouyang et al. [20] utilized human input

* Corresponding Author. Email: steve.conrad@colostate.edu.

for fine-tuning LLM responses using reward modeling and reinforcement learning to better align LLM objectives with operator goals. Raczyński et al. [21] addressed issues of natural language model (NLM) coherence through the application of a transformer to better increase explainability of the outputs of language models. In addition, it has been shown that general language models perform significantly better at extracting information from text-based data that off-the-shelf language models when provided prior knowledge enhancement or a more technical training set [14, 16]. These studies indicate the need for adequate prompting before language analysis tasks.

2.1 Deductive Coding for Textual Data Classification

A frequently used approach for the analysis of multifaceted, qualitative characteristics from unstructured textual data is deductive coding [13]. Deductive coding begins with a set of pre-defined qualitative descriptors (e.g. codes based on theoretical foundations, hypothesises or themes) to define a codebook for labeling different datapoints within a text. Deductive coding is guided through the development of a list of definitions for each feature to be identified, which is typically created with the aid of expert knowledge [13]. Afterwards, raters (or coders) scan the text for the implicit or explicit discussion of each feature before validating results using statistical methods to assess interrater reliability [5]. The independent and qualitative performance of deductive coding is subject to the biases of researchers.

Interrater reliability measures are used to assess the quality of the classifications [18]. Interrater reliability is used to improve the trustworthiness of the deductive coding process and can be calculated using various statistics, the most common of which are Cohen's kappa and percent agreement among raters [19] [3]. Due to the prominence of rater bias, LLMs are being explored within the deductive coding process as both raters and validation metrics. Fleiss [7] expanded Cohen's kappa coefficient to represent interrater agreement amongst more than 2 raters, allowing for stronger assessment of chance agreement within groups of raters.

2.2 Utilization of LLMs for Deductive Coding

LLMs have the potential to act not only as an additional rater, but as many additional raters because of their stochastic response nature. Tai et al. [25] used LLM as a tool to identify 5 different characteristics from interview-based textual data and found that the stochastic nature of LLMs make them more effective as an increasing number of iterations are performed. Despite this, the variability also reduces consensus amongst prompt executions, such that Tai et al. suggest using LLMs in the context of a Human-AI team to validate human responses. In this regard, Chew et al. developed an LLM-assisted content analysis process using gpt-3.5-turbo which showed significant results in agreement with humans when assessing binary classification codes while greatly reducing the time needed for analysis. Gilardi et al. [11] [26] illustrated the capabilities of GPT-3.5 and GPT-4 in zero-shot natural language classification tasks, showing that it is more accurate than crowdsourced coder's even in novel applications.

3 Methods

In this section we describe our deductive coding method for analyzing case studies for evaluating LLM performance against human coding standards. We subsequently describe the application of LLMs using two prompting methods utilizing GPT-3.5 and GPT-4.

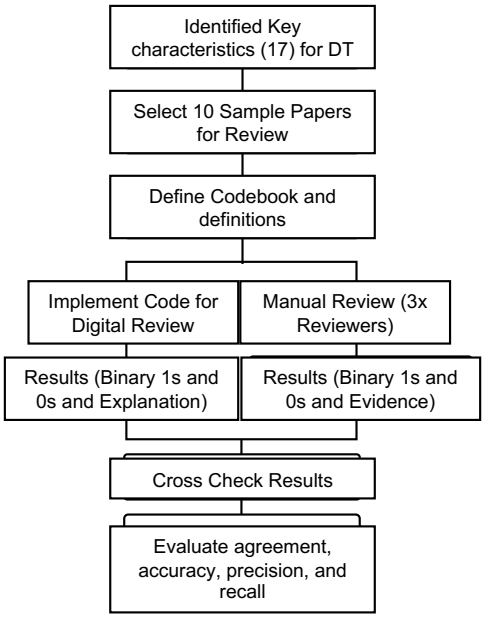

Figure 1. Methodology for evaluating the performance of deductive coding of Digital Twin literature by LLMs as compared to human coders

3.1 Deductive coding

Our deductive coding approach (Figure 1) was developed from ongoing work on developing digital twins at Colorado State University. To characterize digital twins the research team developed a codebook of 17 different digital twin (DT) characteristics adapted from Jones et al. [15] (Table 1). 10 peer-reviewed case studies discussing the development of digital twins were identified for coding. Deductive coding was performed manually using a mixed team of three expert and non-expert researchers according to the methods outlined in Elo and Kyngäs [5] to analyze if the papers discussed each of the DT characteristics. This human or *manual* coding was used as the gold standard for assessing the performance of the LLM. The code of the algorithm, list of documents analyzed, and supplemental analysis are publicly available[1].

The performance of the deductive coding approach was assessed using interrater agreement, accuracy, precision, and recall. Percent agreement was used to assess the overall reliability of the coding and was calculated as shown in McHugh [19]. Success rate or accuracy, precision, and recall of the LLM coding were used to evaluate intra LLM performance. Recall is the rate of relevant retrievals by the LLM to all relevant items while precision is the rate of relevant retrievals to all LLM retrievals. Accuracy is the total relevant retrievals to the entire body of mined data. Accuracy, precision, and recall were calculated as proposed by Witten et al. [27].

The percent agreement metric is limited as it does not account for the possibility of chance agreements [19]. In addition, percent agreement is less effective when employed in classification systems with non-binary or hierarchical levels as the difference in ratings may have variable magnitudes [12]. When using percent agreement as an interrater reliability metric, higher standards must be achieved to address the assumption that all agreements are not driven by chance, thus it is recommended by Stemler [24] that a target percent agreement of 90% should be achieved for the classifications to be considered strong, especially when used with adjacent categories. Fleiss' kappa

[1] https://github.com/anti-integral/ECAI-Paper-Supplementary-Info

coefficient is known to be subject to paradoxical behavior, where the kappa may be underestimated, even as percent agreement is high [6]. For the Fleiss kappa coefficient, Fleiss et al. [8] recommends a range of 0.40-0.75 for fair agreement beyond chance, while at scores below 0.40, agreement is primarily driven by chance agreements. Fleiss Kappa (K) values were calculated using the kappam.fleiss function from the irr v0.84.1 package for R 2023.12.1+402 [10].

Table 1. Digital Twin Characteristic used to test the performance of LLMs for deductive coding of complex urban systems documentation

Digital Twin Characteristic for Coding
Physical Entity and Processes
Virtual Entity
Virtual Processes
Physical Environment
Virtual Environment
Realization
State
Metrology
Fidelity
Parameters
Twinning Rate
Physical-to-Virtual (P2V) Connection
Virtual-to-Physical (V2P) Connection
Use Cases
Perceived Benefits
Data Ownership
Scope

3.2 LLM Utilization

Two LLM prompting approaches were used for comparison, along with two LLMs. Through the OpenAI API Text was extracted from Adobe Acrobat PDF formatted files of the publications using PyPDF2. Multiple verification processes and Optical Character Recognition technology were employed from Python's PyTesseract library, enabling more effective analysis of scanned documents and images compared to simply pulling the base text. Each text is scanned independently and for only one DT characteristic at a time. Through the openai Python API, the LLM instance is reset between each prompt execution. The models used are OpenAI's gpt-4-0125 and OpenAI's gpt-3.5-turbo-16k, both operated at a temperature of 0.7.

We address promoting specifically in our method as effective prompting is quintessential when leveraging LLMs for detailed text analysis, particularly with complex documentation. The quality of outcomes generated by LLMs is heavily dependent on the specificity and clarity of the prompts provided. Good prompting acts as precise instructions to the model, directing its attention to the specific elements of the text that are most relevant for analysis. This is crucial because LLMs, while highly capable, do not inherently understand the context or the importance of certain academic nuances without clear guidance.

In our methodology, we conducted extensive experimentation with different prompt structures to determine the most effective ways to engage with the LLMs. This experimentation involved:

1. Varying the Detail Level: We adjusted the complexity of the prompts when analyzing full texts versus segmented chunks to ensure that the model could maintain focus without being overwhelmed by information.

2. Prompting for Explanations and Binary Classifications: We specifically designed prompts that required the LLMs to explain their reasoning before classifying characteristics as present (1) or absent (0). This step was vital in validating the accuracy of the model's text interpretation against our analytical goals.

One of the most critical aspects of our prompting strategy was the continual refinement of characteristic definitions provided to the LLMs. We provided explicitly defined parameters to effectively parse and interpret the content. These definitions included: A) Explicit Instructions: Detailed descriptions of what constitutes a mention of a particular characteristic, including examples of explicit and implicit mentions. B) Contextual Clarity: Guidelines on the depth of analysis expected, specifying how the model should derive conclusions from the text, whether through direct mentions or inferred context.

Algorithms 1 & 2 show the methods we used for prompting the LLMs. Method 1 passes the LLM the entire text in the form of text, then for each characteristic, passes the body text along with the associated prompt. Method 2 separates each paper into 500-word chunks, passing each chunk to the LLM for deductive coding analysis. When processing the texts, whether in full or by segments, our script passed the codebook and request the LLMs to explain the presence or absence of each characteristic.

Algorithm 1 Whole text search takes as input StudySet, Codebook

```
for Text ∈ StudySet do
    for Dim ∈ Codebook do
        PASS Text to LLM
        PROMPT LLM to find Dim in Text
        if LLM = TRUE then
            Dim ← TRUE
        end if
        OutputTable(Text, Dim) ← Dim
    end for
RETURN OutputTable(Text)
end for
```

Algorithm 2 Chunk text search takes as input StudySet, Codebook

```
for Text ∈ StudySet do
    TextChunk ← list(Text(1 + 500i, 500 + 500i))
    for Item ∈ TextChunk do
        for Dim ∈ Codebook do
            PASS Text to LLM
            PROMPT LLM to find Dim in Item
            if LLM = TRUE then
                Dim ← TRUE
            else
                Dim ← FALSE
            end if
            OutputTable(Text, Dim) ← Dim
        end for
    end for
    RETURN OutputTable(Text)
end for
```

The following prompt was used for all models:

"Explain whether the parameter '{parameter}' is mentioned/directly talked about in the following text and provide evidence from the text. If it does, briefly explain how (3-5 sentences with 2̃ pieces of evidence); if it does not match, briefly explain why the paper does not focus on it (1 sentence). Note that '{parameter}' is defined as '{definition}'."

Return explanations were then converted into a Pandas DataFrame, encoding the presence or absence of characteristics with 1s and 0s respectively as illustrated in Table 2.

Each model and prompting approach was executed 15 times for each paper. The consensus result for each dimension was equivalent to the mode of the LLM responses across all iterations for the specific dimension and paper, including the manual approach. The itera-

tion approach treated each LLM execution as an individual rater. The number of raters (n) for the manual approach is equal to three; for the consensus and iteration approaches n = 4 (3 manual raters + 1 LLM consensus) and n = 18, respectively.

4 Results

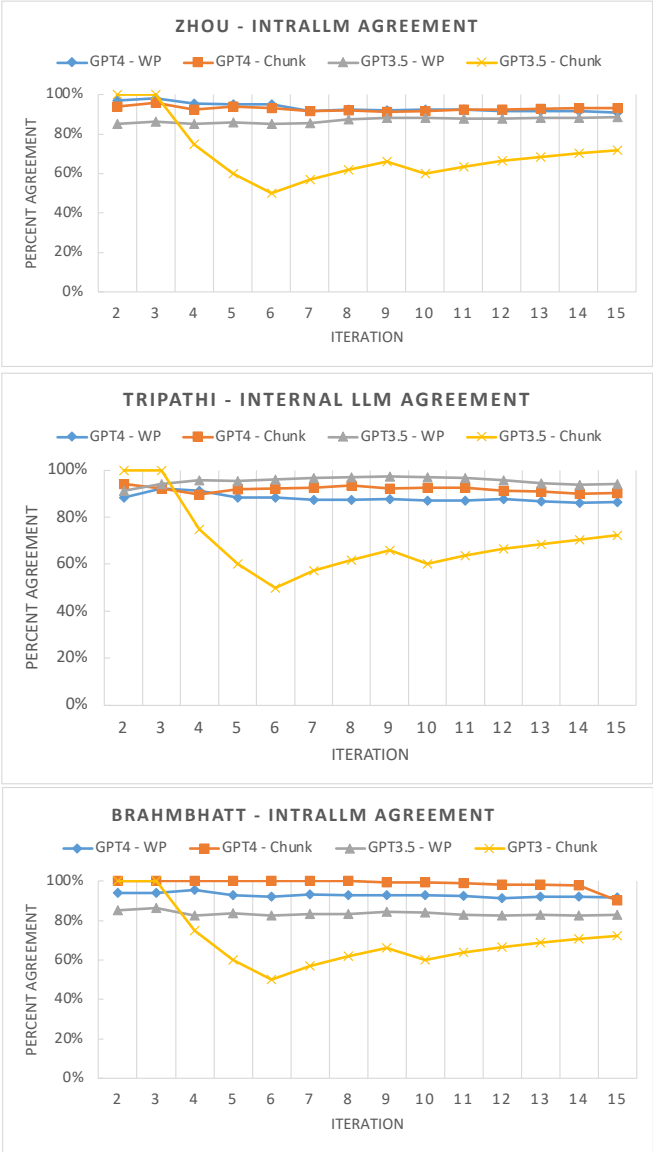

Figure 2. Intra-LLM Percent agreements across iterations of select analyzed papers by model (iteration n=2..15)

Each LLM was found to have strengths and limitations and the precise tuning of definitions was found to be critical; even minor ambiguities could lead to significant variances in the results. By meticulously adjusting the language used in our prompts, we managed to minimize these inconsistencies and enhance the reliability of our findings. Overall, the strategic use of prompted instructions and the careful definition of terms were instrumental in harnessing the full capabilities of LLMs for our research. This approach not only improved the accuracy of our text analysis but also ensured that the models could perform effectively within the specific context of Urban systems context related to digital twin technologies.

4.1 LLM Internal Variance

After 15 iterations and across all authors, the GPT-3.5 whole text model and both GPT-4 approaches showed similar levels of internal agreement, with the GPT-4 chunking model having the highest internal percent agreement range of 98% - 88%. Meanwhile, the GPT-3.5 chunking model had the least internal agreement with a percent agreement range of 71% - 73% across all papers, indicating high levels of variability as almost 30% of data generated was erroneous within itself, even when prompted the same. Figure 2 indicates a sample of the models' internal percent agreement for select papers across iterations.

4.2 Frequency of Classification

Across all dimensions and papers, the consensus of the manual coders marked 86% of classifications as True. When taking the consensus of the 15 iterations of each model, all models - barring the GPT-3.5 chunking approach - generated a lower positive classification rate. The consensus chunking approach utilizing GPT-3.5 classified all dimensions as True across all papers, with 71%-73% of all iterations classifying as True. Due to the frequency of positive values from manual raters, GPT-3.5 chunk had very high percent agreement with raters at 86%. Disagreement between the LLM and manual coders was especially prevalent across certain parameters, especially for the whole text models.

Figure 3 provides a representation of how frequently did each LLMs method and human rater consensus identify the characteristic of interest across all 10 papers across all 17 characteristics. We found difference in classifications between human coders and LLMs and between LLMs across all characteristics. The Physical-to-virtual connection (P2V) dimension was manually rated as True for 90% of papers by the consensus of human raters but the whole paper approach using GPT-4 did not find this dimension in any of the papers. Similarly, both whole text approaches struggled to find evidence of discussions on the twinning rate and metrology.

4.3 LLM and Manual Consensus Agreement, Accuracy, and Robustness

The highest levels of total accuracy were achieved by the text chunking methods using GPT-3.5 and GPT-4 with at 86% and 81%, respectively. Comparatively, the accuracy of the whole text models was 51% for GPT-3.5 and 54% for GPT-4, respectively. Figure 4 and Figure 5 presents the resulting performance of the LLM models as related to the consensus of the three human raters as the comparison benchmark across all dimension classifications. Analyzing chunked text with GPT-4 resulted in a 14% false negative (FN) rate and 6% false positive (FP) rate, while the same model with full text produced a 44% FN rate and 2% FP rate. These FP and FN rates resulted in a recall of 84%, and accuracy of 81% for chunked text and a lower recall of 49%, and reduced accuracy of 54% for whole text.

Adding GPT-4 chunked text results as an additional evaluator showed higher agreement levels of 89% among all reviewers, in contrast, analyzing entire texts with the same showed lower agreement at 83%. GPT-4 proved more effective at accurately identifying positives compared to GPT-3.5 when using the whole text approach, as shown by the higher recall rate when using human consensus ratings as a

Table 2. Example Pandas DataFrame showing the presence or absence of characteristics with 1s and 0s respectively

Characteristic	Response	Encoding
Physical Entity and Processes	The text directly talks about "Physical Entity and Processes" by discussing how digital twins and AI can optimize water utility operations, which include physical systems like water treatment plants and distribution systems. Evidence from the text includes ...	1
Virtual Entity	The paper directly talks about the concept of a "Virtual Entity" under the guise of "Digital Twin" technology, specifically in the context of optimizing energy savings in water supply systems through dynamic pump station scheduling. Here are pieces of evidence from the text that support this...	1
Virtual Processes	The parameter 'Virtual Processes' is not mentioned or directly talked about in the provided text. The focus of the paper is on the application of Digital Twin (DT) technology for dynamic pump station scheduling in water supply systems for energy optimization. Although the concept of replicating physical entities in a virtual environment (a key aspect of 'Virtual Processes') is inherent in the use of DT technology, the specific term or a detailed exploration of 'Virtual Processes' as defined is not provided.	0
Realization	The parameter 'Realization,' as defined, is not directly talked about in the provided text. The focus is primarily on the transformation of data into actionable insights, leveraging digital technologies like AI and digital twins, and improving operational efficiencies in water utilities through advanced technologies and data-driven decision-making. There is no explicit mention of the methods in which the delta or difference in the parameters of the physical and digital entities is determined and subsequently updated, modified, synchronized, or calibrated to ensure consistency or reconciliation.	0

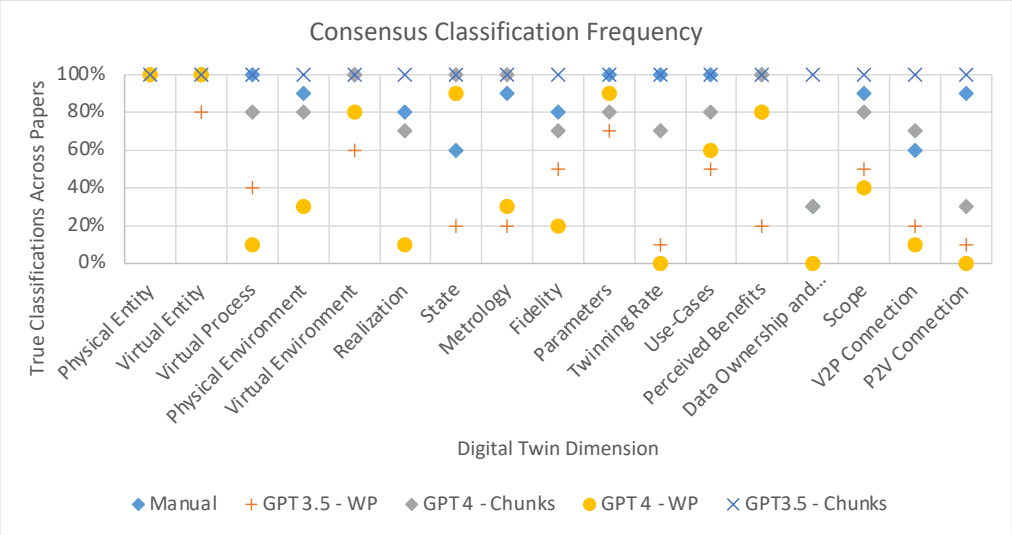

Figure 3. Percent Classification Frequency of each dimension by modeling

Table 3. Precision and Recall of Models across all classifications using consensus approach when compared to manual consensus results

Assessment	GPT3.5 - WP	GPT-3.5 - Chunk	GPT-4 - WP	GPT4 - Chunk
Precision	0.957	0.865	0.960	0.925
Recall	0.456	1	0.490	0.844

benchmark in Table 3. GPT-4 outperformed GPT-3.5 in the chunking approach as GPT-4 proved to be much more precise, indicating that it is has higher robustness as it can more effectively delineate relevant and irrelevant information. The chunking approach for GPT-4 and 3.5 proved to have a much lower false negative rate, albeit at a decrease to the precision of the model and an increase to the false positive rate when compared to both GPT-3.5 and GPT-4 whole text models.

The manual ratings achieved a Fleiss Kappa value of .397, indicating fair agreement amongst manual raters, a statistically significant difference from purely chance agreement. As shown in Table 4

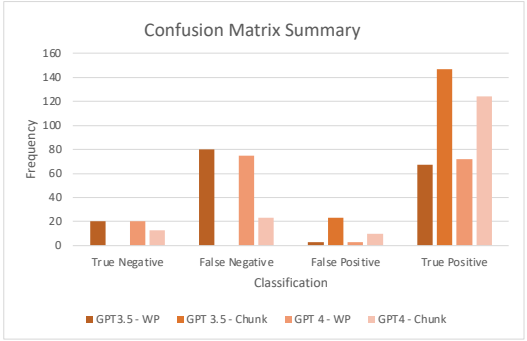

Figure 4. LLM model and prompting approach consensus classifications when compared to human rater consensus classifications

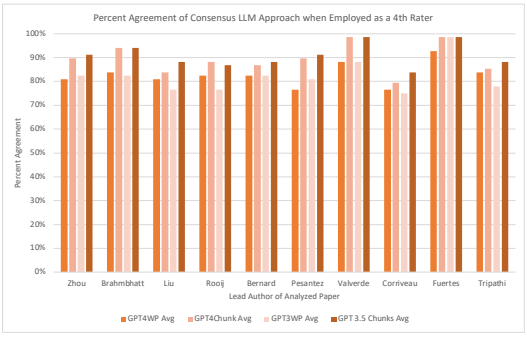

Figure 5. LLM-consensus Percent Agreement with Manual Raters

the Fleiss K values of the models when employed as a 4th rater in the consensus approach decreased from the manual baseline. On the other hand, using all 15 iterations as additional raters (thus n=18), increased the Fleiss K values in all models except the GPT-3.5 chunking approach to moderate agreement levels amongst raters. The negative Fleiss K value of GPT-3.5 Chunks indicates agreement driven almost entirely by chance, which is echoed by the fact that the model's consensus for all classifications was True.

Text chunking using GPT-3.5 resulted in greater relative accuracy, as compared with manual rating. Despite this, the GPT-3.5 chunking approach appeared to have a positive bias in that there were no true negatives were. Consequently, as prompted, it failed to adequately determine the absence of any DT dimension potentially hallucinating or inaccurately interrupting the definitions of the codebook. With this consideration, GPT-3.5 would likely have the highest variation in performance given the positive classification rate of the dataset.

Table 4. Fleiss K values of Models with manual ratings using consensus approach (n=4) and iteration approach (n=18)

Model	Consensus Approach	Iteration Approach
GPT-4 WP	0.195	0.544
GPT-4 Chunks	0.337	0.594
GPT-3.5 WP	0.157	0.442
GPT-3.5 Chunks	0.240	-0.044

5 Discussion

This study aimed to provide a comparative exploration of deductive coding methods utilizing LLMs to address the time and variability constraints of human coding of extensive textual documentation. We found that in the context of coding digital twin characteristics in scientific documents relating to urban systems that LLMs have demonstrated potential to address these challenges and provide a viable additional coder. Thereby providing an option for evaluating large datasets and adding additional context for managing complex urban environments.

Our methodology was designed to approximate the rigor of conventional complex documentation analysis with the augmented capacities of artificial intelligence. In deploying LLMs, we aimed to refine the coding process specifically within the domain of Urban systems applications. This application of LLMs is intended to augment the speed and analytical precision with which complex scientific papers are classified and examined and thereby aid in the sustainable management of urban system infrastructure. The findings suggest when using text chunking strategies, GPT-4 coding results closely align with human coding. These results suggest that LLMs can perform comparably to human coders when provided specific prompts and tasks structured within the deductive coding process. Utilizing LLMs as an additional coder could not only reduce the time and resource constraints associated with manual coding but provide a more reliable source of coding as once the LLMs is trained it produces compatible results. This suggests an application where AI-enabled coding could substantially augment the document review workflow.

Moreover, our research highlights the importance of prompt design and the need for analyzing the semantic processing of LLMs. We found that chunking may more closely resemble how humans process language. Our research also found that all models failed to effectively assess the presence or absence of a characteristic, leading to large discrepancies between human and LLM classifications within a subset of data. We however caution the readers on blanket application of the findings from this study. The performance demonstrated was specific to the codebook and prompting approach. Further research is needed to consider and address the range of consensus found and research would be beneficial to explore the specifics of how each GPT model explained its findings. With greater understanding into the semantic processing of LLMs, prompts and deductive coding codebooks can be engineered for better LLM performance and explainability. Whether these observed effects continue with later versions of the OpenAI or other developed LLMs is an area of potential research. Ultimately we found that given the high percentage of false negatives that certain contextual scanning could lead to anomalous investments. So while LLMs present a viable tool, their application should deployed with rigorous validation against human benchmarks.

Acknowledgements

This work was supported through related research funded by the National Alliance for Water Innovation and US Department of Energy (NAWI 3.25), Prof. Steven A Conrad is the corresponding author of the paper.

References

[1] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. ACM, 2021. ISBN 978-1-4503-8309-7. doi: 10.1145/3442188.3445922. URL https://dl.acm.org/doi/10.1145/3442188.3445922.

[2] R. Chew, J. Bollenbacher, M. Wenger, J. Speer, and A. Kim. LLM-assisted content analysis: Using large language models to support deductive coding, 2023. URL http://arxiv.org/abs/2306.14924.

[3] J. Cohen. A coefficient of agreement for nominal scales. 20(1):37–46. Publisher: Sage Publications Sage CA: Thousand Oaks, CA.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[5] S. Elo and H. Kyngäs. The qualitative content analysis process. 62 (1):107–115, 2008. ISSN 0309-2402, 1365-2648. doi: 10.1111/j.1365-2648.2007.04569.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2648.2007.04569.x.

[6] R. Falotico and P. Quatto. Fleiss' kappa statistic without paradoxes. 49(2):463–470, 2015. ISSN 0033-5177, 1573-7845. doi: 10.1007/s11135-014-0003-1. URL http://link.springer.com/10.1007/s11135-014-0003-1.

[7] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[8] J. L. Fleiss, B. Levin, and M. C. Paik. *Statistical Methods for Rates and Proportions*. Wiley Series in Probability and Statistics. Wiley, 1 edition, 2003. ISBN 978-0-471-52629-2 978-0-471-44542-5. doi: 10.1002/0471445428. URL https://onlinelibrary.wiley.com/doi/book/10.1002/0471445428.

[9] P. Fusch and L. Ness. Are we there yet? data saturation in qualitative research. 20(9), 2015. URL https://scholarworks.waldenu.edu/facpubs/455.

[10] M. Garmer, J. Lemon, I. Fellows, and S. Singh. Various coefficients of interrater reliability and agreement, 2014.

[11] F. Gilardi, M. Alizadeh, and M. Kubli. ChatGPT outperforms crowd-workers for text-annotation tasks. 120(30):e2305016120, 2023. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2305016120. URL http://arxiv.org/abs/2303.15056.

[12] M. Graham, A. Milanowski, and J. Westat. Measuring and promoting inter-rater agreement of teacher and principal performance ratings. 2014.

[13] H.-F. Hsieh and S. E. Shannon. Three approaches to qualitative content analysis. 15(9):1277–1288, 2005. ISSN 1049-7323, 1552-7557. doi: 10.1177/1049732305276687. URL http://journals.sagepub.com/doi/10.1177/1049732305276687.

[14] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. URL http://arxiv.org/abs/2311.05232.

[15] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52, 2020. ISSN 1755-5817. doi: https://doi.org/10.1016/j.cirpj.2020.02.002. URL https://www.sciencedirect.com/science/article/pii/S1755581720300110.

[16] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. 36(4):1234–1240, 2020. ISSN 1367-4803, 1367-4811. doi: 10.1093/bioinformatics/btz682. URL https://academic.oup.com/bioinformatics/article/36/4/1234/5566506.

[17] M. Marathe and K. Toyama. Semi-automated coding for qualitative research: A user-centered inquiry and initial prototypes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 1–12. Association for Computing Machinery, 2018. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3173922. URL https://dl.acm.org/doi/10.1145/3173574.3173922.

[18] J. Marques and C. McCall. The application of interrater reliability as a solidification instrument in a phenomenological study. 2015. ISSN 2160-3715, 1052-0147. doi: 10.46743/2160-3715/2005.1837. URL https://nsuworks.nova.edu/tqr/vol10/iss3/3/.

[19] M. L. McHugh. Interrater reliability: the kappa statistic. pages 276–282, 2012. ISSN 18467482. doi: 10.11613/BM.2012.031. URL http://www.biochemia-medica.com/en/journal/22/3/10.11613/BM.2012.031.

[20] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. 35:27730–27744, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[21] J. Raczyński, M. Lango, and J. Stefanowski. The problem of coherence in natural language explanations of recommendations. In K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, and R. Rădulescu, editors, *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2023. ISBN 978-1-64368-436-9 978-1-64368-437-6. doi: 10.3233/FAIA230482. URL https://ebooks.iospress.nl/doi/10.3233/FAIA230482.

[22] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[24] S. E. Stemler. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. 9(1), 2004. doi: 10.7275/96JP-XZ07. URL https://openpublishing.library.umass.edu/pare/article/id/1540/. Publisher: [object Object].

[25] R. H. Tai, L. R. Bentley, X. Xia, J. M. Sitt, S. C. Fankhauser, A. M. Chicas-Mosier, and B. G. Monteith. An examination of the use of large language models to aid analysis of textual data. 23: 16094069241231168, 2024. ISSN 1609-4069, 1609-4069. doi: 10.1177/16094069241231168. URL http://journals.sagepub.com/doi/10.1177/16094069241231168.

[26] P. Törnberg. ChatGPT-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning, 2023. URL http://arxiv.org/abs/2304.06588.

[27] I. H. Witten, E. Frank, and M. A. Hall. Credibility. In *Data Mining: Practical Machine Learning Tools and Techniques*, pages 147–187. Elsevier, 2011. ISBN 978-0-12-374856-0. doi: 10.1016/B978-0-12-374856-0.00005-5. URL https://linkinghub.elsevier.com/retrieve/pii/B9780123748560000055.

[28] A. M. Woollams, M. A. Lambon Ralph, G. Madrid, and K. E. Patterson. Do you read how i read? systematic individual differences in semantic reliance amongst normal readers. 7, 2016. ISSN 1664-1078. doi: 10.3389/fpsyg.2016.01757. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2016.01757/full. Publisher: Frontiers.

[29] W. Wu and P. Hoffman. Validated measures of semantic knowledge and semantic control: Normative data from young and older adults for more than 300 semantic judgements. *Royal Society Open Science*, 9(2): 211056, 2022.