

Integrating Computational Text Analysis into Risk and Crisis Communication Development

Madison H. Munro; Manuel Ruiz-Aravena; Elizabeth A. Shanahan; Savanna Washburn; Ann Marie Reinhold

Accessibility Disclaimer:

For a more accessible version of this document, please submit an accessibility request form through the Montana State University Library website.

Integrating Computational Text Analysis into Risk and Crisis Communication Development

Madison H. Munro^{¶*}, Manuel Ruiz-Aravena[†], Elizabeth A. Shanahan[‡], Savanna Washburn[‡], Ann Marie Reinhold^{¶,§}

[¶]*Gianforte School of Computing, Montana State University, Bozeman, MT, USA*

[†]*Department of Wildlife, Fisheries and Aquaculture, Mississippi State University, Starkville, MS, USA*

[‡]*Department of Political Science, Montana State University, Bozeman, MT, USA*

[§]*Pacific Northwest National Laboratory, Richland, WA, USA*

*Corresponding author

Email: madison.munro@student.montana.edu, mir41@msstate.edu, shanahan@montana.edu, savanna.washburn@student.montana.edu, reinhold@montana.edu

Abstract—Qualitative textual analysis is advancing with the integration of Natural Language Processing (NLP) and Large Language Models (LLMs). Although many multidisciplinary researchers adopt these analysis tools, Risk and Crisis Communication (RCC) researchers have not taken full advantage of such tools in content analysis tasks. We investigate computational tools' ability to replicate human analysis in RCC research. We integrate NLP and ChatGPT into content analysis steps performed on insider threat source text. The first content analysis step tasks ChatGPT with coding insider threat text as character types defined by the Narrative Policy Framework (NPF), and the second step leverages select NLP tools to derive meaning in the coded sentences. Content coding with ChatGPT varies in performance based on the amount of detail present in prompts. Select NLP tools have varying degrees of success when processing and analyzing coded text. Both ChatGPT and NLP require human intervention when performing content analysis, thus a mixed method approach is necessary for future RCC research.

Index Terms—Natural Language Processing, Large Language Model, Content Analysis, DARC Framework

I. INTRODUCTION

Advancements in computer science provide empirical techniques to process and analyze large sets of text. These techniques include Natural Language Processing (NLP), a set of machine learning tools that help with text classification and analysis [1], and Deep Learning models like Neural Networks and Large Language Models (LLMs) [2]. Both NLP and Deep Learning are widely adopted by researchers across multiple research domains— from STEM and medicine to the humanities and social sciences [3], [4].

Computational textual analysis enables researchers across various social science fields to empirically analyze qualitative text [1], [5]. One research domain where computational textual analysis is instrumental is Risk and Crisis Communication (RCC). A handful of RCC researchers have integrated NLP and LLMs to swiftly develop effective messaging across various hazard domains [1], [6], [7]. However, the use and integration of NLP and LLMs remain exploratory in RCC research.

Development of RCC messaging typically relies on manual content analysis [1], [8]. This content analysis is guided by a theoretical framework for identifying key terms and phrases, message structure, and beliefs expressed [8]. Solutions for guiding content analysis with computational tools have been proposed and tested [1], [8]–[10]. Moreover, messaging strategies are needed that bridge across hazard types and scientific disciplines and domains [11]. One solution proposed is the Domain Agnostic Risk Communication (DARC) Framework [12].

The DARC Framework integrates computational textual analysis in RCC message development to systematically improve content analysis and message construction efficacy [12]. Furthermore, the DARC framework reduces RCC development time through the integration of LLMs, thus enabling timely message delivery to impacted populations. We apply this framework to develop RCC for organizational insider threats. We focus specifically on steps using NLP and LLMs to perform content analysis, using the Narrative Policy Framework (NPF) [13] as the guiding framework.

The following research questions (RQs) drove our insider threat RCC content analysis:

- 1) Can LLMs perform content coding¹ as good or better than human coders using the NPF as a guiding framework?
- 2) Can select NLP techniques effectively identify and analyze NPF language to include in future RCC messaging?

II. METHODS

A. Gather Insider Threat Source Text

Insider threat source text came from two places: structured interviews and authoritative sources. For interviews, we recruited eleven participants from five organizations via snowball sampling (IRB Protocol #2024-1486-EXEMPT). Each participant varied in insider threat expertise, ranging from cybersecurity experts to individuals with minimal computational

¹The categorization of language in qualitative source texts (*sensu* [1])

skills. We interviewed participants on their experiences dealing with organizational insider threats; three interview questions targeted Problem Definitions, three targeted Risk Perceptions, and three targeted Solutions. Responses to each question were recorded and then transcribed.

Authoritative insider threat sources were pulled from three sources: the USA Cybersecurity & Infrastructure Security Agency's (CISA) 2020 Insider Threat Mitigation Guide², Gurukul^{3,4}, and Montana State University's Insider Threat Mitigation Guide. Each source was grouped into one of three hazard information sections—Problem Definition, Risk Perception, and Solutions—then underwent text preprocessing.

B. Data Preprocessing

Both sets of insider threat source text underwent manual and computational preprocessing. For manual preprocessing, we redacted sensitive interview information, extracted interview responses, and removed figures and tables within authoritative source text. For computational preprocessing, we converted the text to lowercase and removed numbers, URLs, and punctuation from the text (except for ending punctuation). All computational preprocessing was done in R [14] version 4.4.3 using the *tidytext* [15] package.

The preprocessed text was split into two copies before content coding steps. The first copy contained three subsets corresponding to hazard information sections defined in section II-A, and the second copy contained the original text separated into individual sentences. Both copies provided data for subsequent content coding steps with OpenAI's ChatGPT model⁵.

C. Content Coding with ChatGPT

Content coding on insider threat text required two sets of instructions⁶. Both instruction sets contextualized the insider threat text, provided NPF definitions for character types, and provided explicit instructions for response generation and formatting. The instruction sets guided two separate executions of ChatGPT (model version: *gpt-4o-mini*; model temperature: 0.3). These model instances ran within OpenAI's Python API⁷ to code for characters and character language on insider threat text.

The first ChatGPT model instance identified characters within the section-level text that matched the NPF definitions for Hero, Victim, or Villain (*sensu* [13]). For each section, we ran the model three times to code for the three character types. All characters identified were fed into the next model instance as a list of potential insider threat characters.

The second ChatGPT model instance classified sentence-level text as Hero, Victim, or Villain language. Language

groupings were based on which NPF-defined characters frequently appeared in a sentence and how these characters were framed in the sentence. Characters came from the list compiled in the first ChatGPT model instance.

The second model instance used three different prompts for the classification task. The first prompt classified sentences as (1) one or more NPF character types, (2) "NONE" if the sentence contained no characters, or (3) "UNCLEAR" if the character types were unclear. The option to label sentences as "NONE" was removed in the second prompt, and the third prompt removed the option to label sentences as "NONE" or "UNCLEAR". All prompt information was otherwise identical. These three different versions of the classification prompts were used to assess differences in inter-coder reliability between ChatGPT and humans.

The prompt engineering and human coding were aligned with best practices in social science research. Specifically, human coders and ChatGPT were allowed to assign multiple character labels to each sentence.

D. Content Coding Inter-coder Reliability

Content coding with ChatGPT underwent two rounds of inter-coder reliability to assess character type label agreement. The first round assessed the inter-coder reliability between a subset of ChatGPT-coded data and the same data subset coded by two trained human content coders. The data subset consisted of a 20% stratified random sample (SRS) of the section level-data and a 10% SRS of the sentence-level data. To code each SRS, the human coders followed the same instructions developed for ChatGPT content coding, and the coders adhered to the first prompt for sentence classification⁸. Inter-coder reliability between ChatGPT and the human coders was measured via Cohen's Kappa [16].

The second round assessed the inter-coder reliability between two copies of both the full section- and sentence-level text. For the sentence-level text, we used the classification prompt that exhibited the best inter-coder reliability between ChatGPT and the human coders. To assess inter-coder reliability between the two copies of sentence-level text, we ran the content coding pipeline described in Section II-C twice; inter-coder reliability between the two texts was measured via Cohen's Kappa for this round as well. After measuring Cohen's Kappa, we merged the two texts based on their assigned character label. Any disagreements in character label were resolved by assigning the sentence a multi-character label.

E. Natural Language Processing Tasks

The final coded sentence-level text was copied and grouped into two corpora sets for NLP analysis. One corpora set grouped sentences based on assigned character type, and one corpora set grouped sentences based on hazard information section. Both corpora sets provided text input for Frequency Analysis and Sentiment Analysis, and the original copy of the sentence-level text provided a dataset for Word Classification.

²<https://www.cisa.gov/resources-tools/resources/insider-threat-mitigation-guide>

³<https://gurukul.com/blog/what-is-an-insider-threat/>

⁴<https://gurukul.com/blog/best-insider-threat-tools-and-strategies-for-mitigating-risks/>

⁵<https://openai.com/chatgpt/overview/>

⁶Appendix A: <https://doi.org/10.5281/zenodo.15027281>

⁷<https://openai.com/api/>

⁸Appendix B: <https://doi.org/10.5281/zenodo.15027281>

a) *Word Classification*: We performed Word Classification in *Python* version 3.12.5.⁹ to predict character labels for insider threat sentences. Two classification models were fit to an 80/20 training and testing dataset of the coded sentence data. The models chosen were a Random Forest model and a Support Vector Machine (SVM). We trained both models with the training dataset encoded by the BERT [17] model “all-mpnet-base-v2.” After fitting both models and encoding the training dataset, we assessed the accuracy, precision, recall, and F1 scores of character classification on the testing dataset.

b) *Frequency Analysis*: We assessed the frequency of insider threat characters within hazard sections by calculating their Term Frequency-Inverse Document Frequency (TF-IDF)¹⁰. Before calculating TF-IDF, we removed stopwords from sentences using both a custom stopwords library and the *smart*¹¹ *R* stopwords library. We then calculated the TF-IDF for uni-, bi-, and trigrams within each sentence. For each ngram investigated, we also calculated the TF-IDF for their stemmed equivalents. The stemmed and nonstemmed ngrams provided a wider range of words from which meaning was derived.

c) *Sentiment Analysis*: We measured the sentiment of insider threat sentences in each corpora set. For each sentence in each corpora set, we calculated the overall sentence sentiment by summing the polarity scores calculated for each term in the sentence. Term polarity was determined using polarity scores provided by the *afinn*¹² library in *R*. After calculating the sentence sentiment for each corpora set, we performed One-Way Analysis of Variance (ANOVA) on the data, assessing any differences in mean sentiment across character and section corpora. We also performed Two-Way ANOVA to assess differences in mean sentiment across character types within sections.

III. RESULTS

A. Character Coding Inter-coder Reliability

The inter-coder reliability between ChatGPT and human coders varied depending on which sentence classification prompt was used. As the level of detail in each prompt decreased, label agreement between ChatGPT and the human coders increased (Table I). In contrast, the inter-coder reliability between the two ChatGPT coded data indicated very high agreement in character label assignment. Assessing label agreement yielded a Cohen’s Kappa of 0.907.

B. Word Classification

Both Word Classification models classified insider threat sentences as Hero, Victim, or Villain language with low fidelity (Table II). The Random Forest model averaged a 0.321 model performance accuracy after 10-fold cross-validation. These cross-validation scores ranged from 0.158 to 0.629 (Fig. 1).

The SVM performed better than the Random Forest model on average (Table II). The SVM averaged a 0.438 model

TABLE I
COHEN’S KAPPA CALCULATED FOR CHARACTER LABEL AGREEMENT
BETWEEN CHATGPT AND HUMAN CODERS

Sentence Classification Prompt Used by ChatGPT	Cohen’s Kappa
Classify sentence as one or more NPF character type, “NONE”, or “UNCLEAR”	0.309
Classify sentence as one or more NPF character type or “UNCLEAR”	0.393
Classify sentence as one or more NPF character type	0.474

TABLE II
PERFORMANCE METRICS OF THE RANDOM FOREST MODEL AND SVM
ON THE 80/20 DATASET SPLIT

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.393	0.388	0.393	0.356
SVM	0.398	0.392	0.398	0.377

performance accuracy after 10-fold cross-validation. These cross-validation scores varied less than scores for the Random Forest model, ranging from 0.333 to 0.703 (Fig. 1).

C. Frequency Analysis

Frequency Analysis on the section corpora unveiled potential key insider threat characters and terms. Nonstemmed bigrams revealed clearer characters and character language compared to other stemmed and nonstemmed ngrams investigated¹³.

Character types impacted which insider threat bigrams were important within hazard information sections (Table III). Many of the bigrams in the section corpora corresponded to explicit Hero, Victim, and Villain characters. Some of these bigrams show up as more than one character type (e.g., the bigram “compromised data” in the Risk Perception corpus). Other bigrams corresponded to character language as opposed to actual characters (e.g., the bigram “feel educated”).

D. Sentiment Analysis

Sentiment Analysis provided insight into how character types impact insider threat sentence structure (Fig. 2). Within the character corpora, mean sentiment differed significantly across each corpus (One-Way ANOVA, F-score: 6.592, p-value: 0.001, df: 2). Within the section corpora, there were no detectable differences in mean sentiment across sections (One-Way ANOVA, F-score: 0.016, p-value: 0.984, df: 2).

Mean sentiment did not differ significantly when only considering sections (Two-Way ANOVA, F-score: 0.038, p-value: 0.963, df: 2). However, mean sentiment differed significantly across each section when accounting for character types (Two-Way ANOVA, F-score: 6.694, p-value: < 0.001, df: 2).

IV. DISCUSSION & FUTURE WORK

A. Content Coding with Large Language Models

With respect to RQ #1, leveraging LLMs like ChatGPT can streamline qualitative content coding steps in RCC development. Our content coding results show that ChatGPT

⁹<https://www.python.org/>

¹⁰https://cran.r-project.org/web/packages/tidytext/vignettes/tf_idf.html

¹¹<https://cran.r-project.org/web/packages/stopwords/readme/README.html>

¹²https://afit-r.github.io/sentiment_analysis

¹³Appendix C: <https://doi.org/10.5281/zenodo.15027281>

TABLE III
TOP THREE CHARACTER TYPE BIGRAMS ACROSS THE SECTION CORPORA BASED ON TF-IDF

Character Type	Problem Definition		Risk Perception		Solutions	
	Bigram	TF-IDF	Bigram	TF-IDF	Bigram	TF-IDF
Hero	"indicator person"	0.251	"compromising data"	0.365	"quality code"	0.106
	"phishing emails"	0.147	"vigilant groups"	0.293	"resourcing people"	0.071
	"flagged organization"	0.139	"cyber threats"	0.221	"access information"	0.070
Victim	"negligent data"	0.120	"compromising data"	0.164	"job position"	0.180
	"organization logs"	0.120	"feel educated"	0.147	"resourcing people"	0.115
	"information leaving"	0.115	"cyber attacks"	0.139	"research lab"	0.092
Villain	"indicator person"	0.139	"compromising data"	0.453	"backing things"	0.264
	"indicators compromise"	0.122	"delete stuff"	0.377	"taking extra"	0.133
	"adept technology"	0.120	"high risk"	0.165	"drives backing"	0.132

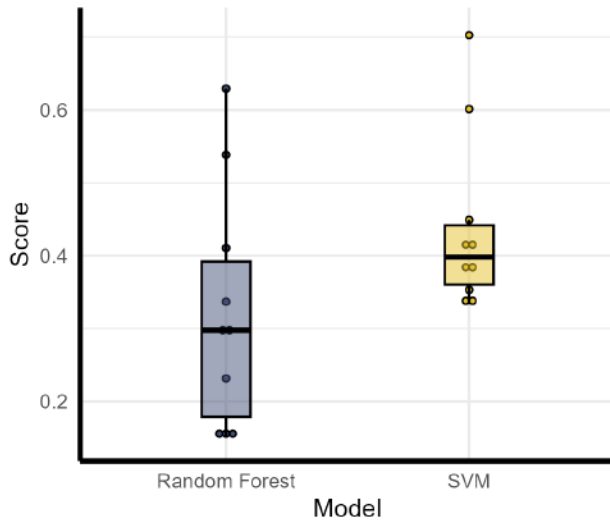


Fig. 1. Comparison of 10-fold cross-validation accuracy scores between the Random Forest and SVM models.

has near complete agreement in character label assignment (above 80%) between separate instances of itself. Achieving high label agreement indicates that at least 80% of the coded data is well-represented, thus conclusions drawn from the data are sound [16]. Coding data with ChatGPT suitably replicates content coding while ensuring coded data is well-represented.

Replicating the time-consuming and laborous task of content coding is achieved through robust prompt engineering. Creating robust prompts combines zero-shot or few-shot learning with well-structured response generation instructions [18], [19]. Our prompts for NPF content coding combine few-shot learning with structured prompt components such as study context, explicit instructions, and output format. Robust prompt engineering can lead to improved performance in qualitative text analysis [20]. However, qualitative text analysis done by a well-prompted LLM requires human involvement.

Content coding using an LLM needs human intervention to assist in validation and quality assurance. While LLMs can be refined to return high quality results and performance, prompt

engineers risk over-fitting the model [21]. Our inter-coder reliability results for ChatGPT and human coders indicate that including more detail in coding instructions returns lower quality responses (Table I). Additionally, LLMs cannot use deductive reasoning like humans, leading to less nuanced responses compared to humans [22]. Responses generated by LLMs can also be heavily influenced by biased or inaccurate information [8]. To ensure accurate, nuanced, and unbiased information is captured in LLM responses, future integration of LLMs in RCC research should include humans.

B. Content Analysis with Natural Language Processing

With respect to RQ #2, NLP techniques leverage quantitative analysis to derive meaning in qualitative source text. Our NLP analysis indicates that Sentiment Analysis and Frequency Analysis reveal NPF characters and character language that could have been missed by manual content analysis (Table III, Fig. 2). Both NPF characters and character language determine the narrative structure and hazard content needed for effective RCC messaging [1], [13]. However, the approach we used for Word Classification falters when determining narrative structure in hazard source text (Table II, Fig. 1). Specifically, allowing for multiple character labels for each sentence (Section II-C) negatively affects the Word Classification models' performance. We conducted additional preliminary analysis in an effort to improve performance, and found that allowing only one character label per sentence increases the performance of Word Classification. Nevertheless, determining message structure, content, and language requires researchers to provide extra contextual information and to remain involved in content analysis [1], [5].

Natural Language Processing cannot fully replace human involvement in content analysis steps. Content analysis with NLP is constrained by the robustness of NLP tools and techniques [5], [8]. All NLP techniques investigated in this study vary in efficacy depending how text is vectorized for classification tasks [23], how researchers tokenize, stem, and preprocess text [8] and what specific models are used to classify text [5] or determine text sentiment [24]. Furthermore, leveraging high-performing NLP techniques may only *augment* qualitative content analysis rather than automating it [1]. Thus, the union of qualitative human and quantitative NLP

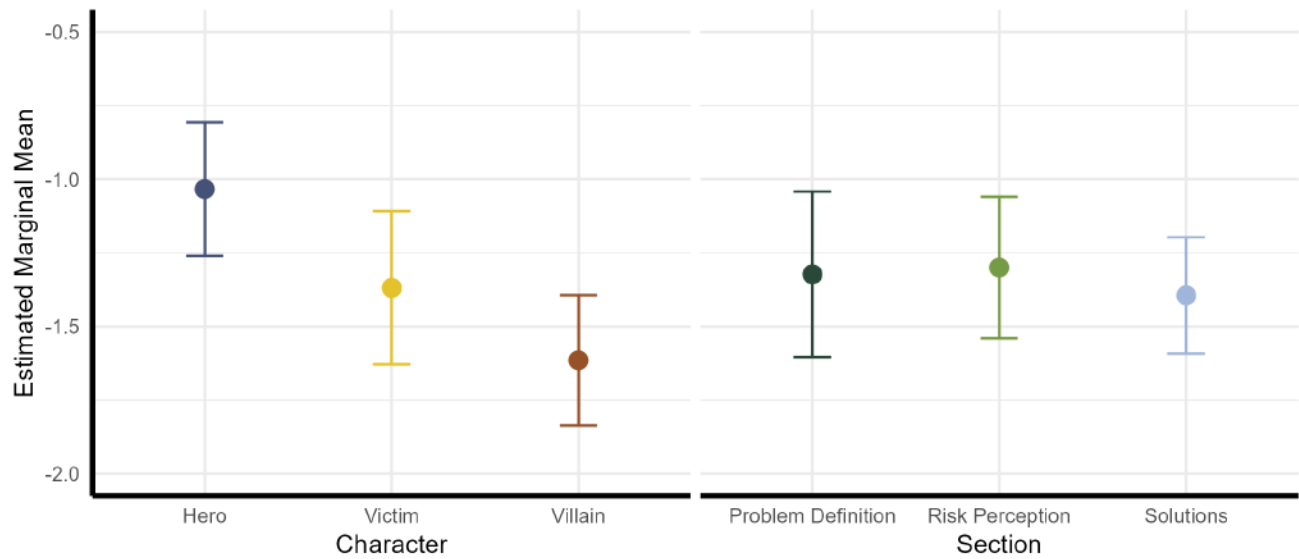


Fig. 2. Mean Sentence Sentiment Across Character Types (left) and Across Hazard Information Sections (right)

content analysis would lead to more effective RCC message development.

C. Future Directions for Computational Content Analysis

Integrating *quantitative* tools to analyze *qualitative* data is enhanced by the involvement of RCC researchers. To that end, RCC researchers benefit from adopting a mixed methods approach when developing RCC messaging [12]. Adopting a mixed methods approach like the DARC Framework enables RCC researchers to develop effective and efficient messaging [12].

Developing effective RCC messages efficiently is a core function of the DARC Framework [12]. Our current work covers applying the DARC Framework to content analysis steps—a critical pre-requisite for developing future RCCs—while also integrating ChatGPT in these steps. We find that ChatGPT is limited when integrated into content coding. For future content coding tasks, it would be beneficial to (1) investigate GPT models optimized to return more nuanced and unbiased responses, (2) simplify character label classification, or (3) investigate other LLMs such as Meta’s Llama model¹⁴, which allows researchers to specify the model’s training data to generate accurate responses. Using a different LLM may perform content coding as well—or better—than humans, thus improving other content analysis steps in RCC message development.

V. THREATS TO VALIDITY

A. Construct Validity

Construct validity refers to the extent our study assesses the investigated construct [1]. We identified some potential threats to construct validity with our preprocessing steps, namely

¹⁴<https://www.llama.com/docs/overview/>

determining hazard information sections and redacting text. Determining if text fell under the sections Problem Definition, Risk Perception, or Solutions was guided by NPF experts. Regarding the redacted text in the interview transcripts, redacting certain text was necessary given the sensitivity of the information relayed.

We identified a few threats to construct validity with our implementation of ChatGPT for content coding tasks. First, the human coders did not code for the whole data. Performing human coding on the full source text would have been time-consuming and expensive for our team, thus the coders coded on a randomly selected subset of data. The second threat identified is ChatGPT’s high variability in responses. We implemented several measures to limit as much response variability as possible. These measures include providing stringent, well-defined instructions in response generation prompts [18], lowering model temperature to reduce the likelihood of model hallucination [25], and using the most advanced model version available [26].

B. Internal Validity

Internal validity refers to the extent to which our cause-and-effect conclusions are sound [1]. We found no major threats to internal validity with our ANOVA on the data. Before performing ANOVA on the data, we assessed threats to statistical assumptions of equal variance of residuals, normality, significant outliers, and independence of observations. We found no significant violations of the above assumptions in our evaluation. Additionally, there was no statistically significant difference between fitting an interactive model for the Two-Way ANOVA versus an additive model (ANOVA model comparison, F-score: 1.198, p-value: 0.310, df: 4). Thus, we fit an additive model for the Two-Way ANOVA.

C. External Validity

External Validity refers to whether our study results can be generalized beyond the current research [8]. The use and integration of LLMs and NLP in RCC research is well documented [1], [5], [8] and can be applied to future RCC research. However, our current research uses the NPF as the guiding framework for content analysis and insider threats as the domain of interest, which could be too specific to generalize to most RCC research on its own. Yet, our application of the DARC Framework ensures that content analysis and subsequent message construction can be generalized to any hazard or theoretical framework of choice [12].

VI. CONCLUSION

Risk and Crisis Communication research benefits from a mixed methods approach for message development. Key components for effective and efficient RCC message development are the use and integration of NLP and LLMs in content analysis. However, each content analysis step continues to benefit from human validation and quality assurance. Human intervention in content analysis will likely be reduced as NLP tools and LLMs advance and refine. Continued and cogent integration of NLP and LLMs into RCC development will improve the efficacy and efficiency of future RCC messaging.

VII. ACKNOWLEDGMENT

This human subjects research was approved by the Montana State University IRB as protocol #2024-1486-EXEMPT. M. Ruiz-Aravena was supported by the U.S. Forest Service International Programs (Agreement 22-DG-11132762-347) and the work is a contribution of the Forest and Wildlife Research Center, Mississippi State University, supported by McIntire-Stennis funds. We thank Dr. Clemente Izurieta and Brittany Boles from the Montana State University (MSU) Software Engineering & Cybersecurity Lab (SECL) for providing expertise in NLP analysis steps and providing critical feedback on this manuscript's writing and structure, respectively.

REFERENCES

- [1] A. M. Reinhold, E. D. Raile, C. Izurieta, J. McEvoy, H. W. King, G. C. Poole, R. C. Ready, N. T. Bergmann, and E. A. Shanahan, "Persuasion with precision: Using natural language processing to improve instrument fidelity for risk communication experimental treatments," *Journal of Mixed Methods Research*, vol. 17, no. 4, pp. 373–395, 2023.
- [2] A. Kucharyv, *From Deep Neural Language Models to LLMs*, pp. 3–17. Cham: Springer Nature Switzerland, 2024.
- [3] O. Suissa, A. Elmalech, and M. Zhitomirsky-Geffet, "Text analysis using deep neural networks in digital humanities and information science," *Journal of the Association for Information Science and Technology*, vol. 73, no. 2, pp. 268–287, 2022.
- [4] J. Wang, H. Deng, B. Liu, A. Hu, J. Liang, L. Fan, X. Zheng, T. Wang, and J. Lei, "Systematic evaluation of research progress on natural language processing in medicine over the past 20 years: Bibliometric study on pubmed," *Journal of Medical Internet Research*, vol. 22, no. 1, 2020.
- [5] L. K. Nelson, D. Burk, M. Knudsen, and L. McCall, "The future of coding: A comparison of hand-coding and three types of computer-assisted text analysis methods," *Sociological Methods & Research*, vol. 50, no. 1, pp. 202–237, 2021.
- [6] E. Karinschak, S. X. Liu, J. S. Park, and J. T. Hancock, "Working with ai to persuade: Examining a large language model's ability to generate pro-vaccination messages," *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. CSCW1, p. Article 116, 2023.
- [7] S. Lim and R. Schmälzle, "Artificial intelligence for health message generation: an empirical study using a large language model (llm) and prompt engineering," *Frontiers in Communication*, vol. 8, 2023.
- [8] M. H. Munro, R. J. Gore, C. J. Lynch, Y. D. Hastings, and A. M. Reinhold, "Enhancing risk and crisis communication with computational methods: A systematic literature review," *Risk Analysis*, vol. n/a, no. n/a, 2024.
- [9] E. A. Shanahan, A. M. Reinhold, E. D. Raile, G. C. Poole, R. C. Ready, C. Izurieta, J. McEvoy, N. T. Bergmann, and H. King, "Characters matter: How narratives shape affective responses to risk communication," *PLOS ONE*, vol. 14, pp. 1–24, 12 2019.
- [10] E. D. Raile, E. A. Shanahan, R. C. Ready, J. McEvoy, C. Izurieta, A. M. Reinhold, G. C. Poole, N. T. Bergmann, and H. King, "Narrative risk communication as a lingua franca for environmental hazard preparation," *Environmental Communication*, vol. 16, no. 1, pp. 108–124, 2022.
- [11] A. M. Reinhold, R. J. Gore, B. Ezell, C. I. Izurieta, and E. A. Shanahan, "From cyclones to cybersecurity: A call for convergence in risk and crisis communications research," *Journal of Homeland Security and Emergency Management*, 2025.
- [12] A. M. Reinhold, M. H. Munro, E. A. Shanahan, R. J. Gore, B. C. Ezell, and C. I. Izurieta, "Embedding software engineering in mixed methods research: Computationally enhanced risk communication," *IJMRA*, vol. 15, no. 2, pp. 67–72, 2023.
- [13] E. A. Shanahan, M. D. Jones, M. K. McBeth, and C. M. Radaelli, "The narrative policy framework," in *Theories of the Policy Process* (C. M. Weible and P. A. Sabatier, eds.), ch. 5, New York: Routledge, 2018.
- [14] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2024.
- [15] J. Silge and D. Robinson, "tidytext: Text mining and analysis using tidy data principles in r," *JOSS*, vol. 1, no. 3, 2016.
- [16] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia Medica*, vol. 22, no. 3, pp. 276–282, 2012.
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the NAACL HLT*, pp. 4171–4186, Association for Computational Linguistics, 2019.
- [18] S. K. Mahjour, R. Soltanmohammadi, E. Heidaryan, and S. A. Faroughi, "Geosystems risk and uncertainty: The application of chatgpt with targeted prompting," *Geoenergy Science and Engineering*, vol. 238, p. 212889, 2024.
- [19] T. F. Heston and C. Khun, "Prompt engineering in medical education," *International Medical Education*, vol. 2, no. 3, pp. 198–205, 2023.
- [20] S. Ramanathan, L.-A. Lim, N. R. Mottaghi, and S. Buckingham Shum, "When the prompt becomes the codebook: Grounded prompt engineering (gropoe) and its application to belonging analytics," in *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK '25*, (New York, NY, USA), p. 713–725, Association for Computing Machinery, 2025.
- [21] B. Cao, D. Cai, Z. Zhang, Y. Zou, and W. Lam, "On the worst prompt performance of large language models," in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 69022–69042, Curran Associates, Inc., 2024.
- [22] M. Amirizani, E. Martin, M. Sivachenko, A. Mashhadi, and C. Shah, "Can llms reason like humans? assessing theory of mind reasoning in llms for open-ended questions," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, (New York, NY, USA), p. 34–44, Association for Computing Machinery, 2024.
- [23] S. Selva Birunda and R. Kanniga Devi, "A review on word embedding techniques for text classification," in *Innovative Data Communication Technologies and Application* (J. S. Raj, A. M. Iliyasu, R. Bestak, and Z. A. Baig, eds.), (Singapore), pp. 267–281, Springer Singapore, 2021.
- [24] Y. Mao, Q. Liu, and Y. Zhang, "Sentiment analysis methods, applications, and challenges: A systematic literature review," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 4, p. 102048, 2024.
- [25] L. Barkley and B. van der Merwe, "Investigating the role of prompting and external tools in hallucination rates of large language models," 2024.
- [26] M. Lewandowski, P. Łukowicz, D. Świątlik, and W. Barańska-Rybak, "Chatgpt-3.5 and chatgpt-4 dermatological knowledge level based on the specialty certificate examination in dermatology," *Clinical and Experimental Dermatology*, vol. 49, pp. 686–691, 08 2023.