2023 Summer Conference on Applied Data Science

Executive Summary

Laboratory for Analytic Sciences

NC STATE UNIVERSITY

PUBLISHED
November 2023

1. Introduction

The annual Summer Conference on Applied Data Science (SCADS) is an 8-week event hosted by the Laboratory for Analytic Sciences (LAS) at North Carolina State University (NC State). SCADS launched in 2022 with an overarching multi-year grand challenge to generate tailored daily reports (TLDRs) for knowledge workers. These TLDRs are intended to be fairly short reports that contain a summary of information that is specifically relevant to an individual knowledge worker's unique set of interests and objectives. High-level documents from the National Security Commission on Artificial Intelligence (NSCAI) [1] and the Center for Strategic and International Studies (CSIS) [2] helped to inspire the grand challenge through their message of the need to transform national intelligence by adopting AI-enabled capabilities.

SCADS 2023 brought together 50 participants from government, academia, and industry to advance the state of research around the SCADS grand challenge. Additional collaborators included LAS staff with extensive experience with the workflow and environment in which TLDRs might eventually be deployed and visiting researchers who shared their expertise and perspectives during multi-day engagements. While the TLDRs will eventually be generated in a classified environment, the conference was conducted at an unclassified level.

During the inaugural SCADS 2022, the participants quickly defined a high-level workflow of a TLDR system that provided context for their projects. [5] The workflow consisted of the following steps:

1. Multiple information sources exist that contain data in a variety of formats and modalities
2. Some process or processes identify data within those sources that are relevant to a user's interests and information needs
3. Some process or processes condense the identified data into a quickly-consumable format
4. Condensed information is presented to a user via a TLDR
5. User interacts with the TLDR and interaction details are captured

This workflow aligned with several disciplines and technologies identified as necessary components in a TLDR system, and which served as four focus areas around which working groups and projects were formed: automatic summarization, human-computer interaction, knowledge graphs, and recommender systems. SCADS 2023 was organized around these same focus areas, plus a fifth focus area for 2023: data set creation and augmentation.

The SCADS 2023 problem book (Appendix A), developed by conference organizers and a set of experts in the five focus areas, introduced participants to the focus areas and provided background on progress made during the previous SCADS. The problem book also described a number of critical challenges and research questions in each focus area that, if solved, would advance the state of the art in understanding and technology required to create TLDRs. The problem book served as a guide for participants when developing their research projects. A number of problems provided clear starting points from which participants could quickly initiate a project, while other problems had a broader scope and served as inspiration for participant-defined projects.

In this report, we present the outcomes of work conducted during SCADS 2023. This report is organized into 2 sections. The first section presents a high-level overview of work done during SCADS 2023, including top takeaways and a discussion of selected work. The second section is organized around the different SCADS focus areas, plus an additional section on building TLDR prototypes. Each portion introduces the respective focus area and presents the technical reports from projects related to that focus area, which participants were required to submit to serve as documentation and were not peer-reviewed.

2. SCADS 2023 - Top Takeaways

SCADS researchers can use prototype TLDR systems to discover the path forward.
Necessary components of a TLDR concept exist today and can produce functional, end-to-end prototype TLDR systems, as demonstrated by SCADS 2023 researchers working on prototype development efforts. These prototype systems are helpful in communicating the concept of a TLDR system, establishing baseline performance, and defining general components to any future TLDR systems. However, it was widely agreed that the overall system behavior and performance of these prototype TLDR systems would not be acceptable to operational users.
Recommendation: Use TLDR prototypes to deepen our understanding of TLDR user behavior, interaction between underlying components, and opportunities for further development. Develop a metric to quantify end-to-end system performance and provide insight into the impact of system modifications.

We developed a framework to understand how a TLDR system would meet analysts' needs.
The value provided by any TLDR system will be determined in large part by how well the system is meeting an analyst's needs. A key result from SCADS 2023 was the development of an analyst's hierarchy of needs, modeled after Maslow's hierarchy of needs, which provides a scale of necessity to the various characteristics of a hypothetical TLDR system. Developing metrics that can measure how well a TLDR demonstrates these characteristics should be a top priority for future SCADS participants. These metrics should take into account that while some characteristics such as efficiency or functionality could be measured objectively, others such as usability, trustworthiness, and relevance may be more subjective.
Recommendation: Apply the analysts' hierarchy of needs to provide context for TLDR capabilities and associated metrics. This would assist with identifying coverage and/or gaps in areas of TLDR support to analysts, and enable measurement of the impact of any changes to the system.

Incorporating large language models (LLMs) in the creation of TLDRs helps with summarization, but not recommendations.
SCADS 2023 researchers made a significant effort to investigate the potential of applying large-language models to the grand challenge. Overall, LLMs performed best at tasks related to summarization and worst when serving as a recommender system. LLMs demonstrated some remarkable capabilities for generating summaries and content for a TLDR, and LLMs in isolation or in an ensemble were shown to be effective at refining and evaluating summaries. The best results occurred when the LLM was provided with relevant context for the request and clear instructions on what to produce. Considerations that surfaced around how to effectively employ LLMs in operational contexts and environments included incorporating guardrails to constrain the content being generated by LLMs and identifying compliant and scalable LLMs.
Recommendation: Carefully consider the task to which an LLM will be applied. The task should be appropriate for an LLM and should be defined in a clear and comprehensive manner that imposes constraints or guardrails on LLM behavior.

The existing TLDR prototypes need to be larger in scale.
The prototype TLDR systems created during SCADS 2023 demonstrate ways in which a TLDR system could provide tailored, relevant information to an individual user. These prototypes also fall short of meeting the grand challenge. Specifically, a TLDR system that fully addresses the grand challenge would need to be larger in scale along at least three different dimensions:

1. Types and Sources of Information: Potential users who interacted with the prototypes often requested the incorporation of additional types, sources, and modes of information. Future instances of a TLDR system must consider ways in which a larger variety of information can be rapidly and effectively integrated into the knowledge base.
2. Number of Users: The current prototype systems were not specifically designed to accommodate large numbers of concurrent users. To achieve the goal of enabling tailored daily reports for large numbers of analysts, SCADS researchers should consider how to

enable individualized models, concurrent users, and compartmentalized information.

3. Variety of Information Needs: As the number of users increases, so will the variety of information and tasks required by the user base. Methods for addressing these needs will need to be developed, but care must be taken to avoid deteriorating performance on existing tasks and information needs.

Recommendation: Deepen our understanding of the trade-offs between system characteristics such as performance, effectiveness, and customizability, as the TLDR increases in scale. If increasing scalability reduces system behavior in some other capacity, initial TLDR deployments might benefit from focusing on a specific subset of possible users to enable scaling to a multi-user base while retaining desired system behavior.

TLDR user feedback is essential to further development. TLDR research and prototype development efforts have benefited greatly from engagement with analysts throughout SCADS. This engagement has primarily consisted of interviews, focus groups, and other conversations with potential users. However, user testing is needed to refine the prototype systems and prioritize features and capabilities for development. Such interaction would enable research on the influence of design decisions and underlying capabilities on a TLDR's usability, trustworthiness, and relevance. *Recommendation*: Future work should obtain feedback from potential TLDR users by capturing information about interactions with TLDR prototypes. This feedback can provide insight into possible TLDR updates, such as additional interface features or opportunities for user interaction in the system workflow, and can also be incorporated into underlying models.

3. 2023 Overview

Here we present a high-level overview of projects conducted during SCADS 2023. Projects that implemented end-to-end TLDR prototypes are grouped in the *Building a TLDR* section. Other projects are presented according to one of the five corresponding focus areas.

3.1. Building a Tailored Daily Report, or TLDR

We built multiple TLDR prototypes.

Several projects explored the application of and interaction between TLDR components that spanned multiple focus areas. As part of these efforts, we developed two end-to-end TLDR systems that implemented the full workflow from ingesting data to producing a tailored report for that data. We also explored aspects of computational performance, such as latency and compression, that will be important considerations when operationalizing TLDR components.

When considering the TLDR system, we identified three concepts that are assumed throughout the TLDR workflow: active data, reference knowledge, and information requests. (See 9.1) Active data refers to the stream of transient data that could be incorporated into a TLDR; reference knowledge refers to background information or corporate knowledge that can provide context for incoming information; and information requests refer to the persistent queries that TLDR users have that influence the relevance of the active data.

In one project, we recognized semantic similarity, embeddings, and knowledge graphs as possible mechanisms for implementing these concepts, and developed Python scripts accordingly. The Python scripts executed processes that, together, completed the following tasks to constitute an end-to-end TLDR workflow:

1. Load reference knowledge into a knowledge graph.
2. Ingest new articles into a knowledge graph.
3. Create information request via a natural language query.
4. Infer reference edges by linking entities in the article knowledge graph to objects in the reference knowledge graph.
5. Infer and hypothesize new objects by identifying and attempting to fill gaps in reference knowledge based on article knowledge.
6. Recommend articles by identifying those most relevant to the information request.
7. Summarize the recommended articles.

In another end-to-end system implementation, we ingested active data via an RSS feed of news articles, then provided a graphical user interface that allowed users to select articles relevant to their interests and information needs. (See 9.2) Through user interaction with the recommendations, the system iteratively refined content recommendations for that user. The system then applied summarization functions to highlight the most relevant portions of a recommended article to draw a user's attention to the sections of most interest to them.

While no ground truth data set or established accuracy metric for TLDRs currently exists with which we could compute evaluation metrics, we designed a performance metric intended to enable quantifying performance of the end-to-end system. The metric assesses whether certain key facts from the source articles were included in the summary report, and could serve as a mechanism with which to assess a TLDR system as a black box or to drill down to see which component(s) passed or failed in moving a fact through the system workflow.

We measured other operational performance aspects of TLDR components that will be important to consider when moving from prototype to deployable systems. We conducted multiple tests that represented tasks in a TLDR workflow, such as storing and querying information in a knowledge graph and summarizing content. For each task, we compared

the time to complete different tasks in a TLDR workflow across multiple implementation options for that task. (See 9.4) When querying different types of graph implementations, we observed negligible differences in query completion time across the graph representations that would be undetectable by a human user. Compression ratio measurements of different graph representations varied with compression type, and seemed to be influenced by graph complexity rather than the size in MB of the graphs. In the summarization tasks, we observed that the BART and DistilBART models were the slowest of the models tested in generating summaries, and also required the mode memory to load and run. These types of measurements will be valuable when understanding possible tradeoffs between system efficiency and accuracy when developing operational TLDRs.

3.2. Human-Computer Interaction

People tend to accept AI-generated output, though trust in technology is highly subjective and outweighed by efficiency and functionality as most necessary TLDR user requirements. In the HCI focus area, participants built off of work in 2022 that explored how information is passed through an analytic workflow. For 2023, an overarching theme of the HCI projects was how to align TLDRs with users. We developed an analyst hierarchy of needs that provides a framework for designing and developing TLDRs that prioritize analysts' needs. (See 4.2) The analysts' hierarchy of needs, modeled after Maslow's hierarchy of needs [6], describes characteristics a TLDR needs to display on a scale of necessary to aspirational. These characteristics and necessity scale were based on information elicited through focused discovery group exercises with people experienced with various analytic workflows, and are listed below from most foundational to a TLDR to least:

1. Efficient and functional - Rated as the most necessary characteristic for a TLDR, this moves the mission forward in a productive, lawful way that does not place further burden on the analysts
2. Trustworthy and reliable - Enables the analyst to work with tangible, transparent, immediately available sources they can verify
3. Context-aware and relevant - Incorporates awareness of analytic workflow and tasks in a flexible and responsive manner
4. Tailored - Accounts for the analyst's role, mission, or organization
5. Customizable - Presents content dynamically based on analyst preference

We might apply this hierarchy of needs as a framework for quantifying various aspects of the TLDR. We have ways to measure straightforward interpretations of efficiency and functionality, such as the time it takes a system to complete a process (efficiency) or the accuracy with which a model makes predictions or recommendations (functionality). Other characteristics, however, are less straightforward to articulate and accordingly more difficult to measure. In applying this hierarchy, we could also validate its utility for a broader set of analysts and other types of knowledge workers and refine the hierarchy as needed.

Additional efforts in human-computer interaction investigated aspects of user-TLDR dynamics during different tasks in an analytic workflow. In one study that elicited information about user trust in TLDR components, we found that the study participants tended to trust AI-generated summaries more than human-generated summaries across all dimensions of trust measured in the study. (See 4.3) Another effort focused on understanding the effect of AI-assisted data exploration and hypothesis generation when interpreting a variety of visualizations. We found that AI-assisted data discovery contributed slightly to users seeing more relevant data points than without AI assistance, but did not result in better user decision-making. We also observed high acceptance of AI suggestions by users. (See 4.4) These studies demonstrated that users are accepting of AI output, though trust in that output is subjective. Existing definitions of general dimensions of trust can be used to categorize the variety of reasons an individual might trust a bit of content or the characteristics that contribute to information's perceived trustworthiness. Measurements of

user trust in TLDRs along those dimensions could, in turn, provide insight into trade-offs between tailoring TLDRs to improve trustworthiness to the user and attempts to mitigate potential bias influenced by the AI output.

Another area of effort extended the investigation around subjectivity through a case study that examined individuals' cognitive demands when completing tasks in an analytic workflow, as measured using an fNIRS device. (See 4.5) In this work, individuals read summaries of transcribed conversations from the Nixon tapes that were generated either by a manual or an automated process, then reported which of the summaries they preferred. The individuals did not know by which process each summary was generated. This feasibility study showed that there were detectable differences in the levels of cognitive demand between reading human-generated and computer-generated summaries, and those differences seem to align with the individuals' reported preferred text. Furthermore, a majority of individuals reported a preference for the automatically-generated summaries in this case. These outcomes demonstrate the feasibility of using cognitive demand to understand users' preferences related to different TLDR components. In future work we could attempt to identify neurocognitive latent traits of users and create a map showing how those traits manifest when conducting various tasks in an analytic workflow. Such a mapping could eventually enable us to tailor TLDRs to a user's neurocognitive profile.

3.3. Automatic Summarization

Large language models are effective tools for evaluating automatically-generated summaries and could supplement human feedback.

Given the plethora of tools available that enable users to quickly and easily generate numerous summaries, SCADS 2023 work in the automatic summarization focus area heavily emphasized evaluating generated summaries. This was a slight shift from SCADS 2022, during which summarization efforts favored summarization generation techniques. We explored applications of text embeddings and natural language inference (NLI) in support of summarization-related tasks, such as enabling source attribution in summaries. We assessed multiple summarization evaluation metrics as mechanisms to understand summary quality and as tools to identify false information in an automatically-generated summary.

In the context of a TLDR, it is important that users are able to verify information presented in the TLDR against the source content on which that TLDR is based. We explored the use of text embeddings and NLI as mechanisms to identify which portions of an automatically-generated summary correspond to which portions within the original source content. (See 6.4) In our experiments, we found that text embeddings performed much better than NLI in identifying the corresponding portions of the summary and source texts, or, put another way, in attributing summary content back to the original source. In further exploratory applications of text embeddings, we compared the performance of different embedding models in multiple summarization-adjacent tasks. (See 6.2) These tasks aimed at calculating the similarity between two text passages, including identifying contradictory information, and classifying sentence content. We found that OpenAI's ada-002 embedding model [7] achieved high accuracy across these tasks. Based on these results, we recommend using embedding-based methods for implementing attribution techniques for automatically-generated summaries. Additionally, we posit that a well-designed embeddings-based summarization evaluation metric could surpass traditional metrics like ROUGE in its ability to assess a summary's quality as well as alignment with human preferences for summary style.

In general, summary evaluation metrics aim to quantify at least one of the following properties [8]:

- Conciseness - A summary should effectively convey the most important information from the content source while compressing its length.

- Relevance - The information presented in the summary should be relevant to the primary themes in the source content.
- Coherence - A summary should have a clear structure and flow of ideas that are easy to follow.
- Readability - A summary should be clear and easily understandable.

We explored a range of existing summary evaluation metrics and novel techniques for assessing evaluation quality. Traditional summarization metrics require a human-generated reference against which automatically-generated summaries can be compared. While human-generated summaries are valuable resources, this approach is not scalable to enable traditional assessment of large volumes of content that can be summarized correctly in numerous ways. In multiple experiments, we employed a variety of LLMs to generate multiple summaries for given source content and assess summary quality. We used ChatGPT, specifically, to evaluate summaries by providing descriptions of evaluation properties, then prompting it with source text and summary and asking for scores for properties (See 6.5) When summarizing content from the CNN/Daily Mail data set, ChatGPT's summary assessment scores most closely aligned with traditional summary evaluation metrics related to relevance and coherence. Additionally, metrics computed using ChatGPT as the evaluator were higher than traditional metrics over all property types. When investigating which metrics and implementation methods are most effective at identifying errors in summaries, we found that GPT4 and SummaC-Conv are effective at identifying semantic errors, such as discrepancies around entities, relationships, and circumstances mentioned in the source content. (See 6.6, 6.7) Those same models were also the most effective at identifying content verifiability errors, including hallucinations. Effectiveness at identifying other types of errors, such as coreference mistakes and hallucinations, varies across the metrics we tested. When we combined the metrics into different ensembles or combinations of metrics, we found that ensembles that included LLMs to compute metrics correlated highly with human judgments of those summaries. (See 6.9) This suggests that automated methods for evaluating summary quality could feasibly be used to assess the numerous summaries being generated for a TLDR and enable human evaluators to focus on the highest-priority cases.

3.4. Recommender Systems

Multiple options exist to explain recommender model output, but scaling challenges arise when training models to learning from additional data and tasks

TLDRs will need to be able to connect users with information that is relevant to them. Recommender models or systems are a fitting technology to deduce a user's interests and map incoming data to those interests. Building off of work completed during SCADS 2022, we explored the feasibility of training the NRMS (Neural News Recommendation with Multi-Head Self-Attention) from last year on additional data and applying additional explainability techniques to the model. We also investigated the effectiveness of LLMs as recommender models and implemented a multi-task-learning model that learns from different types of user interaction data.

The NRMS model implemented during SCADS 2022 was trained only on news article titles and user interaction data. In SCADS 2023, we sought to determine the effects on model performance when adding the article abstract and body into the training data. (See 5.2) When training the model using article titles and abstracts, we found that out-of-memory errors prevented us from training the model on more than 200 words per article when using the largest virtual machine available (AWS xlarge VM with 24 GB GPU). While it might be possible to parallelize training across multiple GPUs to complete training, these results suggest that it is not currently feasible to train the NRMS model using full article texts. Other options to explore to incorporate more information than headlines into model recommendations include employing automatic summarization to generate article summaries that are below a specified length threshold and then training on those

summaries, and exploring alternative models and architectures that are reportedly more computationally efficient.

Again using the NRMS model trained in 2022, we attempted to exploit the model's self-attention mechanisms to explain why the model recommended specific articles to specific users. (See 5.6) The model architecture includes one encoder that represents user behavior and another that represents article content. We isolated the attention mechanisms for each encoder and created interactive visualizations for each using BertViz. While the visualizations were interesting and informative, they fell short of providing quantified explanations for the user and article encoders. Additionally, a side effect of the model's architecture was that the method of isolating and visualizing the attention mechanism cannot sufficiently connect the user and article encoders to explain why a particular article was or was not recommended to a specific user.

We compared the accuracy of two LLMs applied to the MIND news recommendation task on which we trained the NRMS model. (See 5.3) Some motivations for hypothesizing that LLMs might succeed at news recommendation included language models' ability to understand contextual information, avoid the cold-start problem through zero- and few-shot tasks, and enable natural language queries and recommendations through conversation. To elicit recommendations from the models, we prompted the models to consider a user's interaction history and recommend the most relevant article for that user from a list of articles, and provided a set of article titles each for the interaction history and candidate articles. Both the GPT3.5-turbo and Flan-T5 small LLMs underperformed compared to baseline recommendations, such as random article selection and popularity ranking, when tested on the MIND demo data. The LLMs, however, did demonstrate the ability to provide natural language explanations of why certain articles were recommended. Our manual review of the LLM explanations deemed the explanations to be understandable and reasonable, though further investigation is needed regarding using LLMs to explain automated recommendations.

We investigated multi-task learning for training a recommender model on multiple types of user feedback, such as clicks, likes, and shares, versus the single type of user feedback that our NRMS model was trained on. (See 5.5) For this effort we focused on the entire space multi-task model (ESMM) [10] and training on different combinations of user feedback data from the Tenrec data set [9]. ESMM can learn multiple prediction tasks simultaneously and tends to learn well from sparse data compared to other models. We trained multiple ESMM models, introducing an additional type of user feedback in each training iteration, to produce three models trained on either two, three, or four types of user feedback. The model trained on only two types of feedback (clicks and likes) achieved the highest accuracy metrics of the three models, with model performance decreasing as we added feedback types. Incorporating additional feedback types into the model also increased the time required to train the model. Future efforts could enable a comparison of single- and multi-task model types on the same data set by training the NRMS model on the Tenrec data set, which would provide further insight into the utility of ESMM to recommend TLDR content.

3.5. Knowledge Representation

Knowledge graphs can serve as a useful tool for comparing incoming information to existing knowledge, and the structure of a knowledge graph can be used to constrain what information is included in summaries of the graph. We use the term knowledge representation broadly to mean storing information in a manner that is compatible with computing. During SCADS 2023 we focused on knowledge graphs (KG) as a tool to facilitate knowledge representation. We explored approaches for identifying information that is new in a dynamic knowledge graph and investigated the use of a knowledge graph taxonomy to establish guardrails with which generative summaries of a knowledge graph

could be constrained. Additionally, we explored metrics intended to assess knowledge graph completeness as part of the graph-creation process.

In one feasibility assessment of identifying new information in knowledge graphs, we assumed that new information in a knowledge graph would constitute relevant information that should be reported via TLDR as an update on an entity of interest. (See 7.2) We used ChatGPT as a writing assistant to help generate a fictional scenario and accompanying open-ontology knowledge graph to use as background information, and reviewed the output to check for consistency, sensibility, and validity throughout. We followed the same approach to generate incoming reports and associated knowledge graphs to be compared against the background information. To identify "new" information in the reports, we explored multiple approaches to determine whether the entities and relationships in the article KG referred to the same entities and relationships present in the background KG. We implemented edit distance- and embedding-based approaches to identify and resolve similar entities (nodes) in the KGs. We assumed new information to be relationships from the article KG that did not support information in the background KG, and employed embedding- and natural language inference (NLI)-based methods to identify similar relationships (edges) between entities and then assess whether the article KG relationships supported, contradicted, or was indifferent to those in the background KG. The combination of a high similarity threshold to resolve entities plus an NLI-based edge comparison approach generated productive output alerting users to new information from the articles.

As a step in creating knowledge graphs based on a fictional scenario and reports, we assessed a variety of metrics for quantifying knowledge graph completeness using the Ampligraph Python library [11]. (See 7.2) Graph completeness metrics can provide insight into the structural soundness and data quality of the graph, which in turn could help us determine whether we should continue iterating on the development of the knowledge graphs of fictional data. The different metrics we computed assumed a similar concept of completeness that revolved around the number of nodes to which a node of interest is somehow connected; they varied in assumptions regarding directionality within the graph and methods for selecting starting points for traversing paths to determine connections within the graph. We achieved the best completeness score for our knowledge graphs of the fictional data on the TransE metric and the worst with the HolE, as our graphs contained many directional connections between nodes. Understanding the implications of different graph completeness metrics will be helpful if incorporating knowledge graphs in a TLDR system, as they could provide insight into whether certain processes, such as entity and relationship resolution, might be productive when run on a particular knowledge graph.

In another case study, we implemented a prototype system that generates tailored summaries from a fixed-ontology knowledge graph based on incoming reports. We created the knowledge graph by prompting GPT4 to extract from cyber threat reports objects defined in the STIX taxonomy [12]. (See 7.3) For this prototype we assumed that some type of extraction verification or other manual quality control step occurs to maintain a well-curated knowledge graph. We employed GPT4 again to generate summaries tailored toward specific user needs and interests by including in the prompt details about constraining the summaries to only include information from the knowledge graph about a specific set of relevant STIX components. Users could select a persona, for example, junior analyst or discovery analyst, for which we defined a subset of STIX objects to include in a summary based on typical information needs of users in the roles represented by the personas. The prototype demonstrated the feasibility of the approach of applying GPT4, and presumably LLMs more generally, to generate knowledge graphs as well as constrained summaries of those graphs. Future efforts should leverage the corresponding cyber threat data set to assess the quality of the generated summaries by determining whether the summaries can be used to answer a set of questions based on information in the source reports.

3.6. Data Set Creation and Augmentation

We implemented a repeatable process for creating unclassified fictional, or synthetic, data that can serve as a proxy for classified data in an intelligent analyst's work. While vast amounts of publicly available data exist that are useful for developing and understanding initial versions of TLDR components, moving beyond initial versions to productive prototypes and operational systems will require data that applies to the entire TLDR workflow and its users. In the absence of a relevant unclassified data set that contains data in sufficient quantities to train and evaluate state-of-the-art recommender and summarization models, we developed a process for developing fictional, or synthetic, data to support TLDR system research and development that resemble realistic data in an analytic workflow.

For this work, we employed OpenAI's GPT3.5 and GPT4 models to generate the text portions of the synthetic data set. (See 8.2) We generated a base scenario to serve as context for the entire data set, then iteratively expanded on the base scenario to create additional entities, events, and other information related to the base scenario. We created a set of personas to represent analysts who exist within the scenario and have different roles regarding monitoring and reporting different aspects within the scenario. The reports attributed to these analyst personas serve as analytic reports that might be incorporated into a TLDR. We generated numerous reports following this process and created a knowledge graph to capture the information in those reports. Along the way, we identified many considerations when developing synthetic data in this manner. For example, including contextual information such as a background scenario was vital to generating data that had some common thread throughout. We were limited by the model implementations in the amount of background information we could provide as part of the model prompt, and experienced the model "forgetting" background information as the model's context window updated throughout the process. We tolerated a certain amount of model hallucination when generating the fictional data as it expanded, increased variety in the content, and planted potentially misleading information that should be handled appropriately by a TLDR system. In other applications of using LLMs to generate text, however, such hallucinations might be unacceptable and would have to be identified and addressed.

Real-world analytic scenarios involve other data types beyond text, so we implemented a process to generate a geographic scenario that accompanies and is incorporated into the generated text data. (See 8.3) Our general pipeline for generating geographic data consisted of the following procedures:

1. Create an initial base elevation map,
2. Simulate weather and erosion over time to refine the map,
3. Plot the drainage network based on the refined terrain, and
4. Predict locations for human settlements and territories within the map.

We overlaid geographic coordinates on the final map and created a knowledge graph that captured the geographic features in the map. With these additional representations of the geographic data, we were able to incorporate the geography of the fictional scenario into the LLM prompts to provide additional context to increase the realism of the synthetic data. In a separate effort, we augmented the Microsoft News Dataset with additional geographic information related to the articles in the data set. (See 8.4) This additional geographic information could be used to supplement article summaries or provide additional context to a recommender model.

References

1. National Security Commission. National Security Commission on Artificial Intelligence 2021 Final Report.
2. CSIS Technology and Intelligence Task Force. MAINTAINING THE INTELLIGENCE EDGE: Reimagining and Reinventing Intelligence through Innovation.

3. Office of the Director of National Intelligence. What is the PDB?
4. Office of the Director of National Intelligence. U.S. Intelligence Community Budget. 2022.
5. Laboratory for Analytic Science. 2022 Summer Conference on Applied Data Science Technical Report.
6. Maslow, Abraham H. Preface to motivation theory. *Psychosomatic Medicine* **1943**, *5*, 85–92.
7. OpenAI. text-embedding-ada-002.
8. Fabbri, Alexander R.; Kryściński, Wojciech; McCann, Bryan; Xiong, Caiming; Socher, Richard; Radev, Dragomir. SummEval: Re-evaluating Summarization Evaluation. ACL 2021.
9. Yuan, Guanghu; Yuan, Fajie; Li, Yudong; Kong, Beibei; Li, Shujie; Chen, Lei; Yang, Min; Yu, Chenyun; Hu, Bo; Li, Zang; Xu, Yu; Qie, Xiaohu. Tenrec: A Large-scale Multipurpose Benchmark Dataset for Recommender Systems. NeurIPS 2022 Datasets and Benchmarks Track.
10. Ma, Xiao; Zhao, Liqin; Huang, Guan; Wang, Zhi; Hu, Zelin; Zhu, Xiaoqiang; Gai, Kun. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. ACM SIGIR 2018.
11. Costabello, Luca; Bernardi, Alberto; Janik, Adrianna; Pai, Sumit; Le Van, Chan; McGrath, Rory; McCarthy, Nicholas; Tabacof, Pedro. AmpliGraph: a Library for Representation Learning on Knowledge Graphs. 2019.
12. OASIS Standard. STIX Version 2.1. 2021.