When and Where Did it Happen? An Encoder-Decoder Model to Identify Scenario Context

Enrique Noriega-Atala[1], Robert Vacareanu[1], Salena Torres Ashton[2], Adarsh Pyarelal[2], Clayton T. Morrison[2], Mihai Surdeanu[1]

[1]Department of Computer Science, [2]College of Information Science
The University of Arizona
{enoriega,rvacareanu,salena,adarsh,claytonm,msurdeanu}@arizona.edu

Abstract

We introduce a neural architecture finetuned for the task of *scenario context* generation: The relevant location and time of an event or entity mentioned in text. Contextualizing information extraction helps to scope the validity of automated finings when aggregating them as knowledge graphs. Our approach uses a high-quality curated dataset of time and location annotations in a corpus of epidemiology papers to train an encoder-decoder architecture. We also explored the use of data augmentation techniques during training. Our findings suggest that a relatively small fine-tuned encoder-decoder model performs better than out-of-the-box LLMs and semantic role labeling parsers to accurate predict the relevant scenario information of a particular entity or event.

1 Introduction

We present an approach to contextualizing information extraction (IE) that focuses on enhancing events and entities with *scenario context*: the location and time relevant to extracted elements.

Knowing *when* and *where* an event occurs has become increasingly relevant due to the wide adoption of large-scale machine reading technology. Decision makers in high-stakes areas,[1] like epidemiology, public health or climate sciences, are increasingly turning to natural language processing (NLP) technologies to help guide their decision making through automatic evidence discovery and aggregation. In light of this, properly scoping automated IE becomes very valuable to the users of these tools.

One example of a domain when scenario context information is relevant is the modeling of epidemic dynamics, where the literature describes different outbreaks using different mathematical models, such as variations of the susceptible-infected-recovered (SIR) compartmental model. The different scenarios have different parameters and it is useful to contextualize the relevant event to have a better picture of the scenario described. Another example is the domain of climate and climate change, where changes in the climate of different geographical regions over time is studied by the geosciences community.

Scenario information is often explicitly found in the periphery of the text describing an extraction, but not necessarily in the same sentence—thus, it is a form of *inter-sentence* relation extraction (see Figure 1 for examples).

In this work, we tackle the problem as a *generative* task using an encoder-decoder transformer based on T5 (Raffel et al., 2019). Given the locations and temporal phrases in an input passage, we prompt the model to chose and generate the relevant scenario information with respect to a specific entity or event. The main contributions of this work are the following:

(1) An encoder-decoder model finetuned for generating *scenario context*, i.e., the spatial and temporal context of a particular event or concept within a larger phrase.

(2) A high-quality, hand-curated dataset of location and temporal relations with intra- and inter-sentence relations, used to train and evaluate the aforementioned model.

(3) An error analysis of the predictions of the model, shedding light on potential future improvements to this method.

All artifacts used to train and evaluate the model are publicly available.[2]

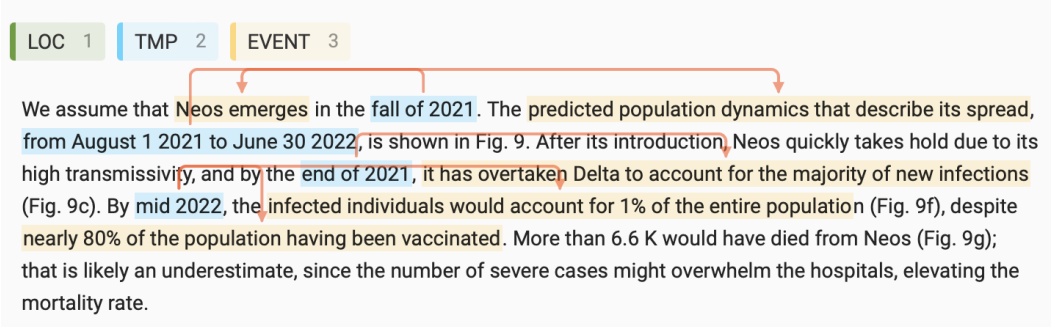
2 Related Work

Annotating *when* and *where* an event occurs is closely related to semantic role labeling (SRL) (Levin and Hovav, 2005; Gardner et al.,

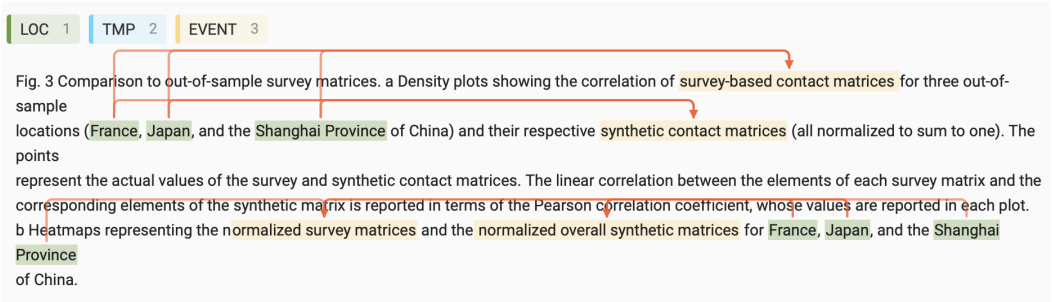
[1] https://www.darpa.mil/program/automating-scientific-knowledge-extraction-and-modeling

[2] Code, data and artifacts available for download from https://github.com/ml4ai/scenario-context.

Findings of the Association for Computational Linguistics: EACL 2024, pages 3821–3829



(a) Temporal annotations



(b) Location annotations

Figure 1: Example annotations in our dataset. Predicates highlighted in yellow represent 'events' with scenario context information assigned to them, text highlighted with cyan represents temporal context, and text in green location context. The arrows connect a context expression to an event they are associated with. The scenario context to event associations are effectively many-to-many relations. Figure 1a shows a passage with several temporal scenario contexts and Figure 1b shows several location scenario contexts.

2017) and document-level relation extraction (Sahu et al., 2019; Xu et al., 2022; Delaunay et al., 2023). An important distinction, however, is that SRL operates on parsing the structure of a sentence and assign roles to phrases within it. Our proposed approach is generative in nature, and is not constrained to the structural elements identified during parsing.

The typical approach for IE is finetuning a pretrained transformer (Vaswani et al., 2017) such as T5 on the data of interest. We build upon the family of encoder-decoder architectures (Bahdanau et al., 2014) to map an input passage to the expected output that represents the relevant scenario information.

Recently, large language models (LLMs) (Brown et al., 2020; Jiang et al., 2023; Touvron et al., 2023) have gained traction, but concerns remain about their tendency to hallucinate. Our work relies on supervised learning via finetuning instead of attempting to mitigate hallucinations.

Contextualizing IE has been of interest to the research community for a while, particularly in the biomedical domain (Noriega-Atala et al., 2018; Sosa and Altman, 2022; Noriega-Atala et al., 2021). In this work we focus instead on the more general class of *scenario information*: the relevant location and time of an entity or event.

3 Dataset

In order to train our model, we created a dataset that contains location and temporal context annotations at both intra- and inter-sentential levels. We focused on 22 epidemiology research articles, including ones that involve modeling the dynamics of infection and outbreak case-studies, published between 2020–2022.

These articles often describe parallel scenarios to compare and contrast the behavior of different outbreaks, making the inference of the scenario context of relevant concepts and events in these papers non-trivial. Correctly understanding the location and time period for specific events is important for the accuracy of any inference drawn from these studies. We excluded any temporal mentions that were abstract or relative, and any location mentions that were modifiers or adjectives. Figure 1 shows an example of an annotated passage of each kind.

The dataset comprises 383 passages, ranging in length from a single sentence to a couple of para-

graphs. A total of 1,382 relations were annotated with 833 (60.3%) location context relations and 549 (39.7%) temporal context relations.

A key aspect of the dataset is the presence of annotations for inter-sentential relations—i.e., relations where the event/concept of interest and its scenario context are in different sentences. These comprise 18% of scenario context relations. Approaches that rely on syntax, e.g., SRL, are not well suited for inter-sentential relations. Figure 2 shows the distribution of sentence distances for the inter-sentential subset of annotations.

Figure 2: Number of sentences between the relevant entity/event and its corresponding context.

A random sample consisting of 13% of the relations was used to measure inter-annotator agreement using the Cohen's Kappa method (Cohen, 1960), resulting in a score of $\kappa = 0.79$.[3]. Details about the annotation guidelines are found in Appendix A

3.1 Data Augmentation

Manually annotating data is time-consuming and labor-intensive. To address this challenge and scale up the amount of data available for the scenario context generation task, we explored two techniques for data augmentation using LLMs—paraphrasing and procedural generation. The details of prompts and generation procedures are provided in Appendix D.

Paraphrasing. To increase the lexical and syntactic diversity of the gold annotations, we generated variations of each passage in the dataset using GPT-4. Additionally, we substituted the temporal and location arguments with alternatives while keeping track of the relations present in them. This process resulted in 434 additional scenario context relations.

Procedurally generated relations. We used GPT-4 to procedurally generate passages containing one or more fictional events with temporal and location context. We used the prompt to control the topic, role of the narrator, length, and number of scenario contexts in each passage. This procedure allows to scale up the amount of training data proportional to one's budget. This approach resulted in an additional 1,361 scenario context relations.

4 Experiments and Results

We trained[4] an encoder-decoder model (Sutskever et al., 2014; Vaswani et al., 2017) based on T5 to generate the location and temporal information relevant to a specific event from its surrounding context. For each relation in the dataset, we prompted (see Appendix B for details) the model to decode the context information of the specific event. Each event may have zero, one or more context relations of each type. The model decoded all of them simultaneously.

We held out a random sample of 20% of the annotions for testing and fine-tuning `t5-base`. Table 1 contains the main results averaged across three runs with different random seeds. Since a particular event may have zero or more annotations of each type, we compute precision, recall, and F1 individually for each and average them across the testing set. We report two variants of this evaluation: (i) span-level, and (ii) token-level. At the *span level*, a generation is considered correct only if it exactly matches the gold standard annotation. In order to ignore minor lexical variations, we applied a basic normalization procedure before comparing strings: converting to lowercase, trimming spaces on both ends, and removing commas. Nevertheless, having a partially correct prediction may still be useful (e.g., `july 5 1987` vs `july 1987`), therefore the *token level* evaluation reports the precision, recall, and F1 scores at the token level, similar to SQuAD (Rajpurkar et al., 2016).

Table 1 contains the results of models trained (i) with manual annotations only and (ii) with the augmented dataset. In general, our approach achieves better results when predicting location than temporal context. Training only with the manual annotations results on the best performance for locations; training with the augmented data decreases the performance for location, but improves the per-

[3]Agreements over 0.61 are considered substantial and over 0.81 are considered almost perfect (Landis and Koch, 1977).

[4]The model was finetuned using a workstation with an RTX 3090Ti GPU, a Threadripper 3960X 24-Core CPU and 128 GB of system memory. Each model was finetuned from `t5-base` using HuggingFace's `Seq2SeqTrainer` for 10,000 steps, 3e-5 learning rate, linear weight decay of 0.1 and batch size of 4.

	Span-level						Token-level					
	Location			Temporal			Location			Temporal		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Annotations	**0.81**	**0.80**	**0.80**	0.76	0.73	0.74	**0.84**	**0.84**	**0.83**	0.76	0.80	0.77
+paraphrases	0.80	0.79	0.78	0.78	0.76	0.76	0.83	0.82	0.82	**0.79**	**0.81**	**0.79**
+synthetic	0.78	0.77	0.76	**0.79**	**0.78**	**0.78**	0.81	0.81	0.80	**0.79**	**0.81**	**0.79**
+para & synth	0.77	0.76	0.75	0.71	0.70	0.7	0.81	0.81	0.80	0.74	0.75	0.74

Table 1: Scenario context evaluation results. P = Precision, R = Recall.

	Location			Temporal		
Method	P	R	F1	P	R	F1
GPT-4o	**0.29**	**1**	**0.22**	**0.33**	0.89	**0.24**
Mistral-7B	0.25	**1**	0.20	0.23	**0.98**	0.19
SRL	0.09	0.01	0.01	0.08	0.05	0.03

Table 2: SRL and LLM baseline results. P = Precision, R = Recall, Mistral-7B = Mistral 7B Instruct v0.2.

formance for temporal context. We hypothesize that this is a consequence of the differences between expressing named locations vs. temporal expressions. There is less variance in how locations are written; they are usually proper nouns or adjectives, whereas time expressions are much more varied— they could be expressed as a standalone year, full date, date range, season, relative temporal phrase, etc. Data augmentation may help increase the diversity of temporal context phrases shown during training, leading to less overfitting to the lexicon compared to location.

Baselines We compare our methods with a decoder-based LLM approach and an SRL system. Table 2 contains the baselines' results. For the LLM baseline, we tested GPT-4o (OpenAI, 2023) and Mistral 7B (Jiang et al., 2023). We asked the models[5] to generate the scenario context for each event and computed the span level results. We find that the LLMs successfully identify time spans and locations relevant to concepts and events, but also tend to predict spurious relations that are not related to the focus of the query. This is reflected in the high recall and low precision exhibited by the LLMs. These observations support the use of supervised learning approaches when feasible.

SRL assigns roles between the clauses in a sentence. We used it as an alternative baseline to the other generative approaches. To test for scenario context detection, we used AllenNLP's structured prediction pre-trained model[6] to parse the sentences containing events in each passage of the ground truth dataset. We considered a scenario context relation 'extracted' if the appropriate context is contained within a predicted modifier argument (`ARGM-LOC`, `ARGM-TMP`) and the text of the event is contained in the union of another argument with the predicate. We found that SRL is not well-suited for this task—it often failed to select the event of focus within an argument.

Error Analysis We performed an error analysis on a sample of the testing predictions of the model trained only with human annotations. Table 3 contains different types of prediction errors broken down by scenario context type. *Spurious predictions* occur when there is no context annotation, but the model generates a prediction; conversely, a *Missing prediction* happens when there is a gold annotation but no prediction from the model. *Mistaken predictions* are when there is both a gold annotation and a prediction, but the model was outright wrong about it. *Partial predictions* occur when the generated text is properly contained in the annotation's text, but is not an exact match—e.g., an event with a location context annotation of "`Western and Northern Europe, United Kingdom`" where the model predicted "`Western and Northern Europe`" is a partial prediction; *Overprediction* errors are the opposite. These instances are considered false positives for the span-level results in Table 1, however their partial, accurate predictions are accounted for in the token-level evaluations. *Other* errors are artifacts of the generative nature of the task. Consider the gold annotation "`California, Indiana, New York`" and the prediction of "`California (CA), Indiana (IN), New York (NY)`"—clearly the prediction is correct; however, the model decoded state acronyms

[5]Prompting details in Appendix C

[6]https://storage.googleapis.com/allennlp-public-models/structured-prediction-srl-bert.2020.12.15.tar.gz

	Context Type	
Error Type	Location	Temporal
Spurious	11	12
Missing	10	7
Mistaken	4	6
Partial	7	3
Over	8	7
Other	3	4
Total	49	39

Table 3: Prediction error types.

alongside the names, which fail an exact string match. The token-level evaluation is able to pick up the full state names. For temporal context, *Mistaken* predictions mostly stem from the variety of ways time intervals can be expressed in text and the inability of the model to abstract that information, e.g., the gold annotation "`between 2009 and 2014`" was predicted as "`2009, 2014`", which correctly includes the endpoints of the range, but fails to specify the crucial detail that this is an inclusive range including the years between them.

5 Conclusions and Future Work

In this work, we introduced an encoder-decoder model finetuned to generate location and temporal context associated with a particular concept or event. We are releasing a dataset of hand-curated annotations from a collection of academic papers in the epidemiology domain that describe the dynamics of outbreaks in different locations and times. We found that our method more accurately recognizes the relevant context than out-of-the-box LLMs or SRL. We also explored the use of data augmentation methods, finding that they resulted in modest improvements in temporal context extraction.

There are at least two promising avenues for future work. The first is expanding the dataset to include more curated annotations from additional domains. This will foster the development of more accurate models with better generalization capabilities. The second is exploring other network architectures, such as span-prediction or decoder-only models. The former is useful for attributing the source of the context prediction and the latter can benefit from the transfer learning potential exhibited by open-source LLMs.

6 Limitations and Ethical Considerations

While the methods described in the paper are not specific to a particular domain, the annotations focus on scientific literature in the domain of epidemiology. The evaluations carried out in this work did not test for generalization capabilities on different domains. Additionally, all of the information used in this work was written solely in English, limiting the potential impact and applications of our contributions. While we evaluated the performance of LLMs for this task, we only tested two different models: GPT-4o and Mistral-Instruct. We recognize that the landscape of LLMs changes quickly and that the state-of-the-art is fleeting. Due to this, our baseline results may be rendered obsolete in the near future.

Acknowledgments

Research was sponsored by the Defense Advanced Research Projects Agency and was accomplished under contract number HR00112290092.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Julien Delaunay, Thi Hong Hanh Tran, Carlos-Emiliano Gonz'alez-Gallardo, Georgeta Bordea, Nicolas Sidère, and Antoine Doucet. 2023. A comprehensive survey of document-level relation extraction (2016-2023). *ArXiv*, abs/2309.16396.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew

Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

J Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33 1:159–74.

B. Levin and M.R. Hovav. 2005. *Argument Realization*. Research Surveys in Linguistics. Cambridge University Press.

Enrique Noriega-Atala, Paul Douglas Hein, Shraddha Satish Thumsi, Zechy Wong, Xia Wang, and Clayton T. Morrison. 2018. Inter-sentence relation extraction for associating biological context with events in biomedical texts. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 722–731.

Enrique Noriega-Atala, Peter M. Lovett, Clayton T. Morrison, and Mihai Surdeanu. 2021. Neural architectures for biological inter-sentence relation extraction. *CoRR*, abs/2112.09288.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.

Daniel N. Sosa and Russ B. Altman. 2022. Contexts and contradictions: a roadmap for computational drug repurposing with knowledge inference. *Briefings in Bioinformatics*, 23.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *ArXiv*, abs/1409.3215.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Wang Xu, Kehai Chen, Lili Mou, and Tiejun Zhao. 2022. Document-level relation extraction with sentences importance estimation and focusing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2920–2929, Seattle, United States. Association for Computational Linguistics.

A Annotation Guidelines

We used LabelStudio[7] to manually annotate scientific articles with scenario context information. LabelStudio was set up with 383 *tasks*, where each contains a section of the article's text containing either location or temporal scenario information of a specific event. At least one annotator carefully read each passage, selecting all the events with a designated location and/or temporal information. The annotator then proceeded to select and link each piece of scenario context information to the relevant event. Figure 1 displays two examples of the user interface of LabelStudio with different types of scenario context and event information.

Two other independent annotators worked in a sample comprising 13% of the tasks. Using these additional annotations, we computed an inter-annotator agreement metric using Cohen's Kappa of $\kappa = 0.79$.

B Model Inputs and Outputs

Figure 3 shows the format of the prompt used as input to the scenario context model. Fields between double curly braces are substituted with the text containing the entity or event of focus in `{{event}}` and the complete passage from which relevant scenario context will be retrieved in `{{context}}`.

Figure 4 shows the output format produced by the model. The model will generate zero or more relevant locations and time expressions per input. Double curly braces are placeholders for the actual predicted values decoded by the model.

[7] https://labelstud.io

```
Text: {{event}}

Context: {{passage}}
```

Figure 3: Input prompt format used by the scenario context encoder-decoder model.

```
location: {{loc 1}}, ..., {{loc n}};
time: {{tmp 1}}, ..., {{tmp n}}
```

Figure 4: Output sequence format decoded by the scenario context encoder-decoder model.

C LLM Baseline Prompt Template

The prompt template shown in Figure 5 was to generate the scenario context predictions from both LLMs in § 4. At run-time, `{{event}}` was substituted by the entity/event of focus and `{{pre_context}}` and `{{post_context}}` were subsituted by the passage's text before and after it, respectively.

The output of the LLMs was parsed as a `JSON` object and used to compute the baseline scores.

```
For the following phrase, look at the
event or concept surrounded by ``` and
tell me the locations and time periods
that relevant to the element surrounded
by ```.
The output format should be a json
object with an array of strings for
type of context. If there is not any
element of a specific type, you will
put an empty array in its value.
Output format:
 {
  "locations": [],
  "time periods": []
 }
Phrase:
{pre_context}```{event}```{post_context}
```

Figure 5: Prompt used to elicit scenario context using an LLM.

D Data Augmentation Procedures

D.1 Paraphrasing Annotations

We used GPT-4 to generate paraphrases of the annotated dataset. Each passage was used as a seed to geenrate multiple paraphrases using the prompts listed in Figure 6.

```
- Please give me a location that is either
close or similar in nature with:
`{location}`.
Please do not return any additional
information.

- Please give me a date that is either close
or similar in nature with: `{date}`.
Please do not return any additional
information.

- Please rephrase the following text, while
keeping the following the following phrase
fixed:`{phrase}`
and maintaining the overall message and
length

{text}

- Please rephrase the following text,
maintaining the overall message and
length

{text}

- Please replace word `{word}` and its
derivatives with the word `{replacement}`
and its appropriate derivatives the
following text:

{text}
```

Figure 6: Prompts used for parapghrasing sequences in the original dataset

D.2 Procedurally Generated Data

We used GPT-4 to procedurally generate synthetic data.

First, we seed the procedure with a set of event types. In our experiments we defined these to be `historical events`, `tech conferences`, and `public health emergencies`. Then, for each event type, we repeat the following steps:

1. For each event type, we ask the LLM to generate ten different *fictional* event names.
2. We ask the LLM to generate five different *narrator roles*—e.g., news reporter, high school student, historian, etc.

3. We prompt the LLM to narrate each event, assuming each of the roles using a predefined set of numbers of paragraphs.

Figure 7 contains a code snippet from a Jupyter notebook used to generate the synthetic data.

```
# Get the number different types of events to develop context for

events_prompt = ChatPromptTemplate.from_messages([
        ("user", "Generate a list of 10 different fictional {event_type} names.\
        Don't any include details, locations, names or dates. You must provide a comma-separated\
        list as a result.")
])

narrator_prompt = ChatPromptTemplate.from_messages([
        ("user", "Generate a list of 5 different narrator roles that describe {event_type}.\
         For example, if we are describing a political event, a narrator role could be\
         a news reporter; if it is a historical event, the narrator could be a historian\
         writing a book, a highschool student writing a homework assignment or a PhD\
         scholar writing a dissertation. You must provide a comma-separated list as the output.\
         Don't include the event type, just the narrator role")
])

generation_prompt = ChatPromptTemplate.from_messages([
        ("system", "You are a {role} describing {event_type}"),
        ("user", """Write {length} about {event}.
         Whenever you mention the event or refer, either explicitly or through pronoums, you
         must wrap between <evt></evt> markup tags. You must include one location context in
         your description. This location context represents where the event took place and
         it could be a geographical region, country, city or location coherent with your
         argument. Any mention of the geographical context must be  wrapped between
         <loc></loc> markup tags. You must include {loc_distractors} other distractor
         locations that are not related to the location context. It must be unambiguous, but
         subtle, that these distractior locations are not where the event took place.
         The distractor locations must be wrapped by <nloc></nloc> markup tags. You must
         also include one temporal context in your description. This temporal context
         represents the specific time or time frame in which the event took place. Any
         mention of the temporal contect must be wrapped between <tmp></tmp> markup tags.
         You must include {tmp_distractors} other distractor times that are not when the
         event took place. It must be unambiguous, but subtle, that these distractor times
         are not when the event happened. The distrator times be wrapped
         by <ntmp></ntmp> markup tags.""".strip())
])

events_chain  = events_prompt | llm | list_parser
roles_chain  = narrator_prompt | llm | list_parser
generation_chain = generation_prompt | llm | str_parser
```

Figure 7: Python code snippet with the prompts used to narrate a fictional event in order to procedurally generate data.