EventNarrative: A large-scale Event-centric Dataset for Knowledge Graph-to-Text Generation

Anthony Colas [*], Ali Sadeghian[*], Yue Wang, Daisy Zhe Wang,
Department of Computer Science, University of Florida
{acolas1, asadeghian, yue.wang1, daisyw}@ufl.edu

Abstract

We introduce EventNarrative, a knowledge graph-to-text dataset from publicly available open-world knowledge graphs. Given the recent advances in event-driven Information Extraction (IE), and that prior research on graph-to-text only focused on entity-driven KGs, this paper focuses on event-centric data. However, our data generation system can still be adapted to other other types of KG data. Existing large-scale datasets in the graph-to-text area are non-parallel, meaning there is a large disconnect between the KGs and text. The datasets that have a paired KG and text, are small scale and manually generated or generated without a rich ontology, making the corresponding graphs sparse. Furthermore, these datasets contain many unlinked entities between their KG and text pairs. EventNarrative consists of approximately 230,000 graphs and their corresponding natural language text, 6 times larger than the current largest parallel dataset. It makes use of a rich ontology, all of the KGs entities are linked to the text, and our manual annotations confirm a high data quality. Our aim is two-fold: help break new ground in event-centric research where data is lacking, and to give researchers a well-defined, large-scale dataset in order to better evaluate existing and future knowledge graph-to-text models. We also evaluate two types of baseline on EventNarrative: a graph-to-text specific model and two state-of-the-art language models, which previous work has shown to be adaptable to the knowledge graph-to-text domain.

1 Introduction

Natural language generation (NLG) is a rapidly developing area of natural language processing (NLP). With the advent of transformer-based language models, such as BERT [4], GPT-2 [31], XLNet [45], BART [20], UniLM [2] T-5 [32], and ERNIE [38], NLG has seen some recent advances in abstractive summarization, dialog response generation, and generative question answering. These tasks in NLG have all been accompanied by previously curated large-scale parallel datasets, where parallel denotes a tightly coupled input/output, allowing for generalized fine-tuning, including: the CNN/DM dataset [13, 25] and Gigaword [36] for abstractive summarization, Persona-Chat [46] and DSTC7 [6] for dialogue response generation, and CoQA [34] for generative question-answering. While the aforementioned NLG tasks have had a history of curated large-scale datasets to finetune on, the task of knowledge graph-to-text has been missing this scale of parallel data.

Knowledge graph-to-text generation is the process of taking structured data in the form of a knowledge graph (KG), which are a collection of subject-predicate-object (s, p, o) triples, and describing the graph through natural language sentence(s). KGs describe real-world entities and their properties and tend to be incomplete. There are over 1,300 publicly available KGs with over 100B triples [14, 24], containing structured knowledge about biomedicine, geography, socioeconomic, life sciences, chemistry, publications, etc., and many cross-domain KGs. Some popular KGs include Wikidata [40], YAGO [33], ICEWS [27], and DBpedia [19]. These KGs are both user curated and extracted from

Submitted to the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. Do not distribute.

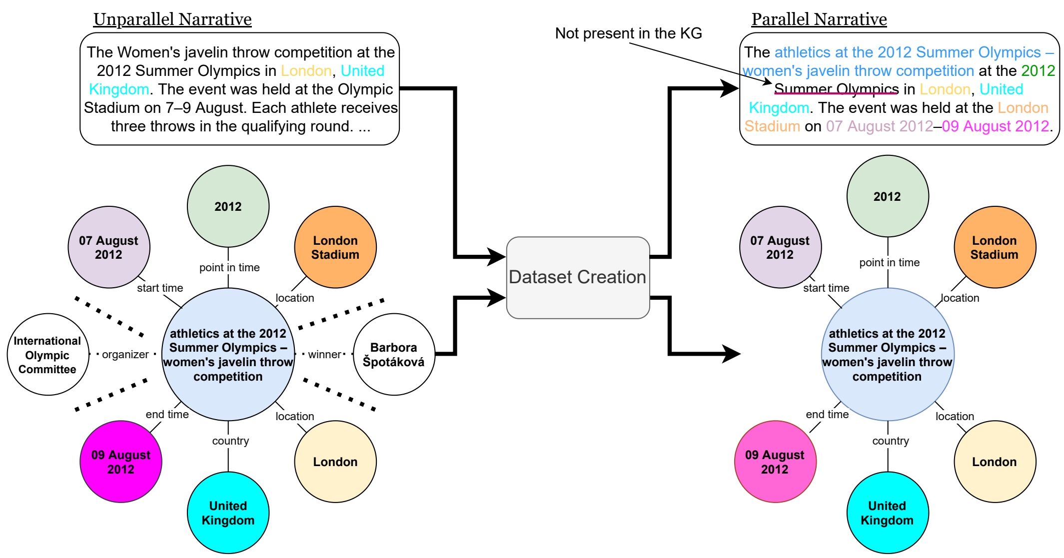


Figure 1: An example source and output narrative for the "Athletics at the 2012 Summer Olympics – Women's javelin throw" event. **Left:** The original unparallel narrative with its corresponding KG from EventKG/Wikidata. Entities in the KG and text are linked by their mutual colors. **Right:** The filtered text and KG for the corresponding event. These are both output from our Dataset Creation system. We highlight the corresponding entities in the texts and KGs.

Wikipedia text. However, there are large disconnects between the data and natural language text. Resolving these disconnects, enables us to better serve the vast amount of structured information in the KGs in a user friendly manner. One solution is finding methods to directly link the KG to its corresponding Wikipedia narrative when possible. We can then use the curated datasets to train NLP models to expand graph narrative generation to other KGs with fewer resources. Figure 1 illustrates an example Wikidata event graph with its corresponding narrative from Wikipedia.

The parallel datasets that currently exist for the knowledge graph-to-text generation are often small in size and do not take advantage of the KG ontology when creating the data. For example, the 2017 WebNLG Challenge dataset (WebNLG 2017) only contains 21,855 pairs of graph and texts, while requiring an expensive hand-annotated operation to generate text on a given graph [7]. Another prominent parallel dataset, the AGENDA dataset, contains 40,000 examples. However, this dataset is generated through the SciIE tool [23] on scientific article abstracts with only 7 different relations. This sparse ontology does not follow any standard KG ontology and induces sparse graphs paired with long texts [16]. Moreover, the dataset contains many entities that are isolated from their KG. We further discuss these and other knowledge graph-to-text datasets in Sections 3 and 5.

Additionally, current knowledge graph-to-text datasets are entity-centric, containing data which are incompatible to narrate events. Events involve multiple actors, complex relations, various lengths, and temporal information, making them more information dense and variant. Numerous event-centric KGs [27, 18, 9] exist, which are valuable to narrate, and recent work has also looked into how to best extract events from text [3, 5, 21]. We therefore develop a more comprehensive algorithm that matches event-centric KGs to their natural-language narration. Many similar events occur frequently but at different times. Thus, our entity matching algorithm has a date matching component to ensure the KG-text pairs contain date/time information. For example, there is a "2014 FIFA World Cup" and "2018 FIFA World Cup", where the event descriptions may have high overlap. We refer to each text instance as a *narrative* of the graph/event, as when describing an event, one is often described to be *narrating* the event [26].

Events are distinct in their length, occurrences, properties, and relations involved. Our dataset reflects this, containing events from different time periods, having thousands of types, and containing approximately 650,000 triples. Therefore, EventNarrative overcomes various shortcomings of existing datasets: it is approximately 6 times larger than the current largest parallel dataset, knowledge graph-text pairs are generated automatically via an existing rich ontology, and there are over 7,000 different

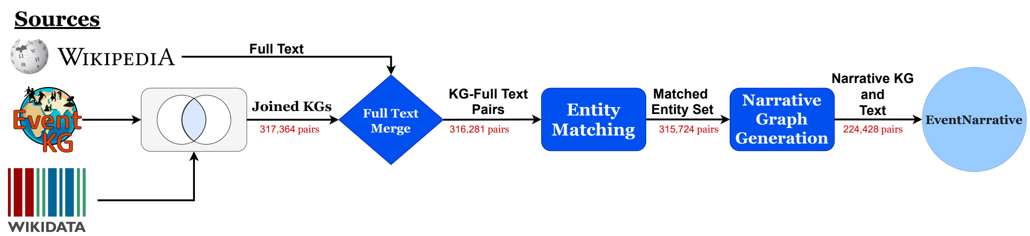


Figure 2: An overview of our data creation approach.

types of events, ranging from sports seasons to social media campaigns. Relations within EventKG include location, event type and start/end time.

We propose EventNarrative, a large-scale supervised graph-to-text dataset. EventNarrative is extracted and paired from existing large-scale data repositories, including Wikidata, Wikipedia, and EventKG [9]. EventKG is a multilingual event-centric temporal KG, which combines event graphs from Wikidata, DBpedia, and YAGO. We begin by first extracting events from EventKG, and then for each event, augment the data with additional corresponding Wikidata information. In total, EventNarrative contains approximately 220,000 data pairs.

We establish baselines on EventNarrative by comparing current state-of-the-art knowledge graph-to-text models on our automatically extracted test set. Future versions of the EventNarrative dataset will enrich the current dataset by incorporating other KGs such as DBpedia and YAGO.

Altogether, our contributions are as follows:

- A large-scale, event-centric, parallel knowledge graph-to-text dataset of over 230,000 KG-text pairs, spanning over 7,000 event types, and over 650,000 triples.
- A comprehensive entity matching and knowledge graph-to-text matching algorithm that automatically pairs KGs to natural language texts.
- Benchmark evaluations and baselines results on EventNarrative.

Our dataset can be found on <link_to_be_released_on_camera-ready_version>.

2 Dataset Creation

This section explains our data creation process. We detail the various data sources used to extract events, our algorithm to match entities and text, and the graph generation technique we used to produce our final narratives and KGs. An overview of our methodology can be found in Figure 2.

2.1 Sources

The knowledge graphs in EventNarrative are first sourced from EventKG [9], a multi-lingual event-centric KG which incorporates events from Wikidata, DBpedia, YAGO, the Wikipedia Current Events Portal, and the Wikipedia events list. Each event contains information on where it was extracted from and any aliases or alternative names for the event, if any. Properties of these events in EventKG include temporal information such as the *hasBeginTimeStamp*, *hasEndTimeStamp*, *startUnitType*, and *endUnitType* relations, as well as spatial information with *hasPlace*. Events are also connected with the *previousEvent* and *nextEvent* relations. We then filter and keep events which are extracted from Wikidata and linked to a Wikipedia page. We do so because Wikipedia articles, and thus the items found in the text/narrative, are directly used to construct Wikidata, which can be queried through the Wikidata query service using an article's Wikidata Q identifier (QID). In total we collect an initial set of *322,674* graph events that have both a Wikidata and Wikipedia resource link through EventKG.

EventKG contains a large number of events, with information pertaining to location, time, as well as actors involved in the events. Yet, many relations and properties found on Wikidata are not in EventKG. We therefore query Wikidata to get an event item's related properties, objects, and labels through the SPARQL-based Wikidata query service[1]. We do so by obtaining an event's Wikidata QID from EventKG and querying the Wikidata online query service, taking precautions to not overload the service by writing aggregate queries to obtain results for multiple events in parallel. We also query for each object's QID, is of use in the entity matching component. Because some relations and properties between EventKG and Wikidata overlap, we first normalize all temporal or location-based relations from EventKG into their corresponding Wikidata property names and remove any duplicates. Next, we filter any events which do not contain a type, e.g. sports season, battle, or election. This type is represented though the relation *wikidata_type_label* from EventKG or *instance of* from Wikidata. By keeping types, future research can filter EventNarrative by event type or incorporate the types into their knowledge graph-to-text models. After filtering, the dataset contains *317,364* graph events.

When narrating an event, important details such as actors involved or sub-events, often lie deeper inside the Wikipedia article itself. While previous work on both table-to-text and knowledge graph-to-text has limited the size and locality of the textual data [17, 42, 15], we extract an event article's whole Wikipedia text to capture all of an event's textual details. We then filter out extraneous details contained within square or curly braces which typically denote altered or omitted information. Consequently, some Wikidata events have no Wikipedia article. After retrieving the whole text, there are *316,281* knowledge KG-text pairs.

2.2 Entity Matching

To match the in-text tokens with its corresponding KG triples, we devise an extensive entity matching technique which is specialized for event data, but can be fitted to capture other types of data. First, similar to Wang et al. [42], we locate all of the hyperlinks in the Wikipedia text and identify their QIDs. After doing so, we match these QIDs with the ones captured from the Wikidata graph in the previous step, preserving those entities which have a match. In-text entities in Wikipedia articles often do not have any links to Wikidata. To overcome this, we check for exact matches between the Wikidata property items and in-text tokens, saving those with a match. One drawback of the exact match method, is that Wikidata entities are shorter in length and may overlap with Wikipedia text that was better suited for a longer Wikidata entity. We therefore first sort all of the Wikidata property entities by length, longest to shortest, before executing our exact match step. This step also allows us to match properties from the Wikidata graphs that are numerical or dates, but only if the date format exactly matches that within Wikipedia text.

Dates are crucial components for any event-centric dataset. We therefore design a separate module to match Wikidata dates within our KG set to those from the Wikipedia text, making our entity matching algorithm biased towards event-centric data. In order to construct an exhaustive search algorithm for in-text Wikipedia dates, we refer to the Wikipedia Style Manual [2] which defines acceptable date formats when editing Wikipedia articles. We write regular expression patterns (RegEx) to match these formats as well as others observed when manually reviewing the EventNarrative dataset. By manually iterating through both the Wikidata graphs and Wikipedia text, we note that many dates only have an overlap in month or year. Therefore, if a date from Wikidata is not found through the RegEx patterns, we match the date if it contains a monthly or yearly overlap. We replace those matches in the narrative with their corresponding match in the KG in order to normalize the graph-text pairs. After performing the entity matching step, we filter out those pairs for which we found no matches, keeping *315,724* knowledge KG-text pairs.

Our entity matching technique prioritizes a high recall, where dates and entities may overlap but not correctly match. To verify our entity matching technique, we sampled 500 events from our final dataset and recruited workers to note any errors in the data related to the linked entities and matching relations. Details can be found in Section 3.

2.3 Narrative KG Generation

For every graph-text pair ($\boldsymbol{G}, \boldsymbol{T}$) obtained so far, we recursively discard sentences (and nodes) from $\boldsymbol{T}$ (and $\boldsymbol{G}$) until all remaining sentences have at least 2 entities in $\boldsymbol{G'}$. This is necessary in order to reduce textual noise. Otherwise we would have texts that could not be generated given the information contained in the graph. Figure 1, shows an example parallel narrative, which contains triples from the

[1] https://www.mediawiki.org/wiki/Wikidata_Query_Service

[2] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Dates_and_numbers#Formats

Table 1: Overview of current knowledge graph-to-text datasets. A dataset has an Entity Match if all entities from the KG are contained in the narrative. A dataset has a Triple Match if all entities are linked to the KG's triples.

Dataset	KGs	Entity Match	Triple Match	Domain	Ontology	Text	Parallel
WebNLG 2017	9,674	✓	✓	15 Categories	DBpedia	Crowdsourced	✓
AGENDA	40,720	✓	✗	Semantic Scholar	N/A	Scientific abstracts	✓
GenWIKI (full)	1,336,766	✗	✗	General Domain	DBpedia	1-10 Wiki sentences	✗
GenWIKI (fine)	757,152	✗	✗	General Domain	DBpedia	1-10 Wiki sentences	✗
EventNarrative	224,428	✓	✓	Events	Wikidata	Full Wiki text	✓

KG. During this process if any of the $\boldsymbol{G}$ or $\boldsymbol{T}$ become empty, we discard the pair. The result is a fully connected graph which represents the current filtered narrative text. One could achieve similar results using a simpler sentence level approach. However, this rather intricate approach of constructing the graph-text pairs using the complete text, preserves triples that are not fully explainable through a single sentence. In the end, we are left with **224,428** KG-text pairs. Further in depth analysis on the final EventNarrative dataset are presented in Section 3.

2.4 Limitations

Our dataset construction methodology is not without limitations. The narrative KGs are of course limited to the elements and relations found in Wikidata, which itself is incomplete, causing it to miss entities found within the narrative text. We knowingly discard sentences which may contain co-references to events. While initially creating the dataset, we experimented with current state-of-the-art co-reference resolution systems, but they performed poorly on event-centric data. This may be because of the overlap between event names and their property names, e.g. *2013-2014 Manchester City F.C. season* and *Manchester City*. Because event names and their properties may have a high overlap, we do not perform any fuzzy matching techniques to extract events.

3 Dataset Analysis

We compare EventNarrative with three popular knowledge graph-to-text datasets, including: WebNLG 2017 [7], AGENDA [16], and GenWiki [15]. Note that because WebNLG 2017 contains multiple texts per graph with an n:1 relationship, we decouple each text from its corresponding graph before analazying the data. Though GenWiki is a non-parallel dataset primarily constructed for unsupervised learning, we wish to also highlight other key differences. We include both renditions of GenWiki, *full* and *fine*, where *fine* has a tighter entity overlap threshold.

3.1 Dataset Synopsis

We begin by first performing a high-level analysis of current knowledge graph-to-text datasets. Among all the parallel datasets in Table 1, our proposed EventNarrative is the largest, having 6 times more KGs than the second largest dataset (AGENDA) and 25 times more than the manually annotated WebNLG. Although GenWiki is larger than EventNarrative, many entities and relations from the triples are not contained within the text and many entities from the text are not found in the graphs. The dataset is purposely created for unsupervised learning and does not model the knowledge graph-to-text supervised task. As AGENDA contains isolated entities that do not belong to triples, the only other dataset with perfect matches between the text and graphs, WebNLG 2017, was handcrafted, limiting the number of samples generated. Since the text was handcrafted in WebNLG-2017, iterative improvements on the dataset such as standardizing the entities and relations within the text based on an ontology becomes extremely challenging. Conversely, because the narratives found in EventNarrative are sourced from Wikipedia, various existing tools can be used to improve the entity matching algorithm. Therefore, EventNarrative is the only dataset that is large-scale, ontology-based, utilizes an open real-world KG, may be used for supervised learning, and contains fully connected graphs that are fully contained within a text narrative.

[2]Note that we compare the WebNLG 2017 dataset released after the challenge which contains 15 categories found on: `https://webnlg-challenge.loria.fr/challenge_2017/.`

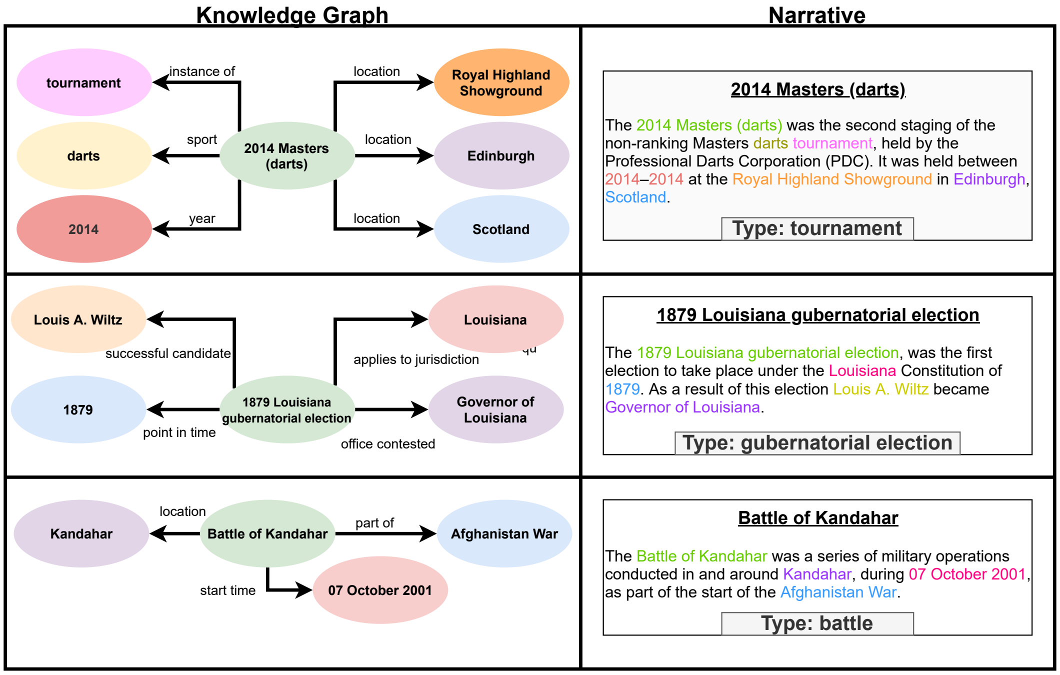


Figure 3: Examples of KG-narrative pairs from EventNarrative. Linked entities are color coded.

Table 2: Statistics of the current parallel knowledge graph-to-text datasets. We report the total number of KG components, number of tokens in the narratives (including the percentage which encompass entity tokens), and overall mean statistics for both the KGs and narratives.

	KG			Tokens			Mean			
Dataset	Entities	Relations	Triples	Total	Unique	Entity	Triples	Text Length	Entities in Text	Triples/Sentence
WebNLG 2017	2730	354	81,927	623,902	8,075	**60%**	2.95±1.55	22.50±11.24	**57%±23**	**2.12±0.98**
AGENDA	159,691	7	180,603	5,760,660	77,896	27%	**4.43±3.19**	**141.47±40.92**	27%±7%	0.81±0.74
EventNarrative	**305,685**	**672**	**656,302**	**11,352,387**	**222,338**	25%	2.92±2.48	50.58±58.59	32%±13%	1.82±1.12

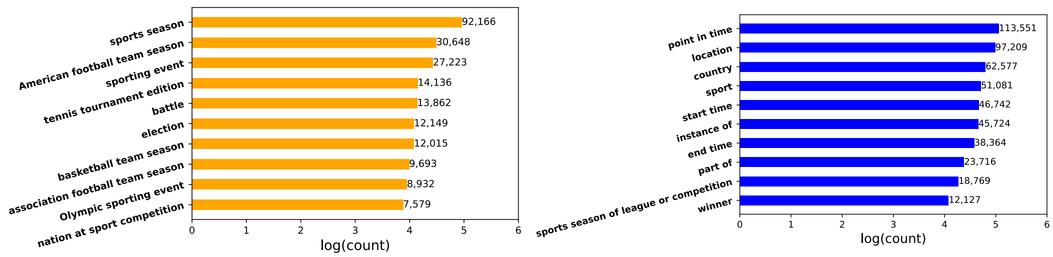
3.2 Statistical Analysis

We now take a closer look at EventNarrative, demonstrating that our dataset contains a large amount of variable data, with closely aligned KG-narrative pairs. Table 2 presents some in-depth statistics between the current supervised knowledge graph-to-text datasets, including: WebNLG 2017, AGENDA, and the proposed EventNarrative. We exclude GenWiki from this analysis because the dataset does not meet the requirements of being parallel, and its entities/triples do not completely align with the text.

From table 2 we see that EventNarrative has approximately 110 times more entities, 2 times more relations, and 8 times more triples than WebNLG 2017, the only other dataset containing no disconnected entities between the KG and text. Unlike [15], we choose not to limit our relations for two reasons: (1) our dataset is not open-domain as in [15], (2) our aim is to build a challenging dataset which simulates real-world KGs. Even so, with 305,685 entities and 224,428 graph-narrative pairs, 672 relations is tractable.

While the AGENDA dataset contains more triples on average per sample, it contains less than one triple on average per sentence, while also holding the longest mean text length throughout all of its samples[3]. EventNarrative contains an average of about two triples per sentence, and contains a more stable text length at approximately 51 tokens. Though both the AGENDA and EventNarrative datasets are almost equivalent in the percentage of text tokens that are entities, recall that in AGENDA not all entities belong to a triple, with approximately 49% of the entities missing from their KG.

[3]All data were tokenized with NLTK: https://www.nltk.org/



(a) Distribution of top 10 event types.

(b) Distribution of top relations.

Figure 4: Distribution of most frequent types and relations in EventNarrative.

As expected, WebNLG 2017 has the highest percentage of entity tokens in the overall text, average percentage of entity tokens in its text per sample, and average number of triples per sentence, as the dataset is hand-crafted and human verified. Though automatically generated, EventNarrative is comprable to WebNLG 2018 in number of triples per sentence, suggesting that our dataset creation process may closely simulate human annotation.

Another notable aspect of EventNarrative is its high variability between samples, as shown by the variance in the number of triples and text length. This makes EventNarrative a complex, yet practical dataset for modeling the knowledge graph-to-text problem, thus verifying our claim that event data is highly variable in length.

We illustrate the distributions of event types and relations in figure 4. The top 10 event types include: sports season, American football season, sporting event, tennis tournament edition, battle, election, basketball team season, association football team season, Olympic sporting event, and nation at sport competition. The top 10 relations include: point in time, location, country, sport, start time, instance of, end time, part of, sports season of league competition, and winner. While EventNarrative is highly variable in its length per sample, the type and relation distributions reveal that a plurality of the events are sports related. Intuitively, this makes sense, as there are many yearly and even monthly sports-related events. EventNarrative also includes a substantial amount of battles, award ceremonies, legal cases, finals, bilateral relations, and elections. Future work using GraphWriter can filter events by types based on their needs.

3.3 Qualitative Analysis

In this section, we evaluate our dataset based on how closely it resembles human annotations. To do so, we recruit 3 annotators to manually check 500 randomly sampled KG-text pairs from EventNarrative. To ensure quality, we recruited annotators that have experience in KG research. We first ask the annotators to verify (and count) if all entities and relations contained within the KG are correct with respect to the narrative. Table 3 demonstrates that both the entities and relations from the KGs are indeed tightly coupled with their respective narratives. Approximately 96% of all entities, and 95% of all relations in the sample were deemed correct. On a coarser level, 397 of the KGs had no errors of any type. Also, annotators reported that most entities which are incorrectly paired are either mischaracterized by Wikipedia or are dates that have different scopes between the KG and source narrative; i.e. containing the day and month of an event within the text, but only containing the year within the KG.

Three different types of events from EventNarrative are shown in figure 3, tournament, gubernatorial election, and battle. The first row shows an example error in our date processing. The original text stated "1–2 November 2014" but is incorrectly replaced with "2014-2014" because of the different scope of the date between the KG and narrative.

4 Benchmark Evaluation

Our aim is to establish a dataset which can be used to help advance the state-of-the-art in transforming knowledge graphs to natural language narratives. We begin this work by comparing established supervised knowledge graph-to-text baselines on EventNarrative. Currently, there are two approaches

Table 3: Qualitative analysis results: entities correct (EC), entities incorrect (EI), relations correct (RC), relations incorrect (RI).

(a) Correct/incorrect per sample.

	EC	**EI**
RC	397	57
RI	30	16

(b) Total correct entities and relations (percentage).

% CE	**% RC**
95.56	95.50

Table 4: Narrative generation results for EventNarrative. The best result for each metric is bolded.

	BLEU	chrF++	CIDEr	METEOR	ROUGE	BERTScore
GraphWriter	30.78	47.91	**4.59**	**27.72**	**71.92**	92.12
$T5_{base}$	12.8	56.76	3.00	22.77	52.06	89.59
$BART_{base}$	**31.38**	**64.71**	3.31	26.68	62.65	**93.12**

to this task: first, modeling the problem with a graph transformer-based network; second, treating the problem as a summarization task by finetuning on pretrained language models (PLMs). The transformer-based network we experiment with is GraphWriter [16], a graph transformer model which can capture local and global information when encoding a graph. Second, we experiment by finetuning on two prominent pretrained language models (PLM), BART [20] and T5 [32], which have been shown to outperform graph-to-text specific models on the AGENDA and WebNLG 2017 datasets [35].

4.1 Experimental Setup

We divide the dataset into a 80/10/10 train/dev/test split. However, for the experiments on BART and T5, we sample 1,000 samples from the dev set and use this for evaluation instead. We do so because of the computational overhead imposed by the decoding step. All experiments were performed on NVIDIA RTX 2080 Ti GPUs.

For the GraphWriter model, we use the version provided by Guo et al. [12] which utilizes the Deep Graph Library (DGL) [41]. We keep its default parameters of: a learning rate of $2{\cdot}10^{-4}$ size batch size of 32, a beam size of 5, 4 attention heads, and train for 30 epochs[4].

To finetune on EventNarrative with BART and T5, we follow a procedure similar to that of [35], prepending "translate from Graph to Text" to the source (graph) data for the T5 model and adding the subject <*S*>, predicate <*P*>, object <*O*> tokens into the vocabulary for both PLMs. All of our PLM experiments are done using the base models released by HuggingFace [44]. As in [35], we use the Adam optimizer and linearly decreasing learning rate scheduler without warm-up with an initial learning rate of $3{\cdot}10^{-5}$. We use a batch size of 2 and beam search size of 3. The dev set's BLEU score is used for model selection.

4.2 Results

We evaluate EventNarrative on frequently used NLG evaluation metrics: BLEU [28], chrF++ [29, 30], CIDEr [39], METEOR [1], and ROUGE [22]. Additionally, as in [35], we also evaluate the test set using BERTScore [47] which computes text similarity based on contextualized embeddings. Table 4 presents the results for each baseline model. Overall, for the EventNarrative dataset, the GraphWriter and BART models give similar results, both significantly outperforming T5. The GraphWriter model outperforms BART on CIDEr, METEOR, and ROUGE_L, while BART performs best on BLEU and BERTscore. The poor results of T5 may be because of the difference in data that BART and T5 were originally trained on. BART was trained on news articles, which can closely resemble events. Overall, our results show that graph-to-text specific models are still competitive to PLMs and deserve further investigation.

[4] For more details, see `https://github.com/QipengGuo/CycleGT`

5 Related Datasets

One of the early knowledge graph-to-text datasets, WebNLG 2017 [7], is a human-annotated parallel dataset that consists of 27,731 graph/text pairs and 9,674 unique graph instances, therefore containing multiple text samples per graph. After the WebNLG 2017 challenge, the dataset was expanded from 9 to 15 categories. As shown in table 1, each KG was extracted from DBpedia. Because the dataset is handcrafted, there is a high precision between the matches in the KGs and text, but the dataset does not scale.

The AGENDA dataset, first introduced by Koncel et al. [16] was constructed by first collecting approximately 40,000 scientific articles from Semantic Scholar, then extracting KGs from the text using the SciIE [23] tool. The dataset contains only 7 relations and many entities which are not connected to a KG. Moreover, the dataset is not bound to any ontology, making it difficult to expand the KG component of the dataset in a standardized fashion.

A recent established non-parellel dataset, GenWiki, is constructed by matching Wikipedia articles with DBpedia entities [15]. However, its goal is to develop a large-scale, non-parallel dataset for unsupervised graph-to-text learning, where all elements in the graphs are not necessarily contained in the text. GenWiki also has a third component, namely entities, which are extracted from the text, but not necessarily contained within the graph. These entities are used to construct both the graph and text for both the graph generation and text generation tasks, respectively. While valuable for unsuperivsed learning in knowledge graph-to-text, we note that this may not model the knowledge graph-to-text problem, where all entities should be contained within the KG and one may not have access to the text in order to extract the shared entities.

EventNarrative sources event items from the more recently established EventKG [9], an event-centric fusion-based KG. Previous works employing EventKG include work on timeline generation [10], event series completion [11], and event-centric question answering [37]. While EventKG contains facts which involve location, time, and event actors, we enrich EventNarrative by extracting all related facts from a given event item, so that the graphs can better align with their corresponding narratives.

6 Discussion and Conclusion

EventNarrative closely resembles the available KGs, and can be used to generate narratives in a supervised manner. This is enabled by its large size, rich ontology, and variability within the data. Our human qualitative analysis verified that about 96% of entities and relations are correctly matched.

Our dataset generation framework is automated, therefore EventNarrative can be re-assembled and extended with other ontological KGs such as DBpedia or YAGO. We will periodically improve and update EventNarrative, as the nature of the dataset depends on continuously adding new events. The dataset generation framework can also be adapted to other types of entity-centric data in order to generate rich and more tightly-coupled sets of knowledge graph-to-text data.

EventNarrative provides the community with new challenges because of the variety within its data, and can also provide new insights into knowledge graph-to-text baselines. Previous parellel datasets have been lacking because of their size, loosely coupled triples, and sparsity, all of which can saturate the results of current baselines. EventNarrative is tightly coupled, allowing researchers to focus on generating proficient models which narrate real-world KGs. We hope that EventNarrative can enable ground-breaking new work in knowledge graph-to-text and event-centric research.

7 Broader Impact

EventNarrative can assist researchers in other fields in studying different graph structured data (E.g. events that are represented as graphs) by providing them with an easily readable narratives. However, as a text generation dataset, there are risks concerning generating fake news and disinformation, specifically related to recent or current events which may appear in the dataset. This can especially occur if the language produced from the KG looks fluent but is completely fabricated [43]. While all of our narratives are extracted from Wikipedia and this issue may not be apparent, we discourage anyone from substituting any text with those that may spread disinformation.

References

[1] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Preprint*, 2020.

[3] Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[5] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, 2020.

[6] Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and Bill Dolan. Grounded response generation task at dstc7. In *AAAI Dialog System Technology Challenges Workshop*, 2019.

[7] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

[8] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, III Hal Daumé, and Kate Crawford. Datasheets for datasets. 2018.

[9] Simon Gottschalk and Elena Demidova. Eventkg: A multilingual event-centric temporal knowledge graph. In *European Semantic Web Conference*, pages 272–287. Springer, 2018.

[10] Simon Gottschalk and Elena Demidova. Eventkg–the hub of event knowledge on the web–and biographical timeline generation. *Semantic Web*, 10(6):1039–1070, 2019.

[11] Simon Gottschalk and Elena Demidova. Happening: Happen, predict, infer — event series completion in a knowledge graph. In *Proceedings of the International Semantic Web Conference (ISWC 2019)*. Springer, 2019.

[12] Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 77–88, Dublin, Ireland (Virtual), 12 2020. Association for Computational Linguistics.

[13] Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1693–1701, 2015.

[14] Anja Jentzsch. Linked open data cloud. pages 209–219. Springer Berlin Heidelberg, 2014.

[15] Zhijing Jin, Qipeng Guo, Xipeng Qiu, and Zheng Zhang. Genwiki: A dataset of 1.3 million content-sharing text and graphs for unsupervised graph-to-text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2398–2409, 2020.

[16] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, 2019.

[17] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, 2016.

[18] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013.

[19] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[21] Sha Li, Heng Ji, and Jiawei Han. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, 2021.

[22] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[23] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, 2018.

[24] John P. McCrae. In *https://lod-cloud.net/#subclouds*, 2021.

[25] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016.

[26] Neal R Norrick. Twice-told tales: Collaborative narration of familiar stories. *Language in society*, pages 199–220, 1997.

[27] Sean P O'brien. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International studies review*, 12(1):87–104, 2010.

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[29] Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, 2015.

[30] Maja Popović. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618, 2017.

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[32] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[33] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer, 2016.

[34] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[35] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv e-prints*, pages arXiv–2007, 2020.

[36] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

[37] Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova. Event-qa: A dataset for event-centric question answering over knowledge graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3157–3164, 2020.

[38] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*, 2019.

[39] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[40] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[41] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, et al. Deep graph library: Towards efficient and scalable deep learning on graphs. 2019.

[42] Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, 2018.

[43] Claire Wardle and Hossein Derakhshan. Information disorder: Toward an interdisciplinary framework for research and policy making. 2017.

[44] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[45] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763, 2019.

[46] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, 2018.

[47] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 2.4.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See section 7.
(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported because it would be too computationally expensive given our resources.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See section 2 and 5. We have also linked all existing assets used in the supplementary material.
   (b) Did you mention the license of the assets? [Yes] See the Appendix
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We curated a new dataset. We are releasing the data as well as the code for curating it.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] All our sources are publicly available and from Wikipedia/Wikidata.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]