EventGround: Narrative Reasoning by Grounding to Eventuality-centric Knowledge Graphs

Cheng Jiayang[♠], Lin Qiu[◇], Chunkit Chan[♠], Xin Liu[♠], Yangqiu Song[♠], Zheng Zhang[◇]
[♠]The Hong Kong University of Science and Technology, [◇]Amazon AWS AI
{jchengaj, yqsong}@cse.ust.hk, zhaz@amazon.com

Abstract

Narrative reasoning relies on the understanding of eventualities in story contexts, which requires a wealth of background world knowledge. To help machines leverage such knowledge, existing solutions can be categorized into two groups. Some focus on implicitly modeling eventuality knowledge by pretraining language models (LMs) with eventuality-aware objectives. However, this approach breaks down knowledge structures and lacks interpretability. Others explicitly collect world knowledge of eventualities into structured eventuality-centric knowledge graphs (KGs). However, existing research on leveraging these knowledge sources for free-texts is limited. In this work, we propose an initial comprehensive framework called EventGround, which aims to tackle the problem of grounding free-texts to eventuality-centric KGs for contextualized narrative reasoning. We identify two critical problems in this direction: the *event representation* and *sparsity* problems. We provide simple yet effective parsing and partial information extraction methods to tackle these problems. Experimental results demonstrate that our approach consistently outperforms baseline models when combined with graph neural network (GNN) or large language model (LLM) based graph reasoning models. Our framework, incorporating grounded knowledge, achieves state-of-the-art performance while providing interpretable evidence.

Keywords: Knowledge grounding, Eventuality-centric Knowledge Graphs, Reasoning

1. Introduction

Narrative reasoning, such as predicting story endings and reasoning with scripts, is a fundamental task in natural language understanding (Mostafazadeh et al., 2016; Li et al., 2018; Mori et al., 2020). Reasoning with narratives depends on the understanding of eventualities[1][2]. Consider the following story:

> "Tom was tired and wanted to have fun. He bought a movie ticket for Harry Potter."

It can be broken down into multiple sub-sentences:

> (**E1**) Tom was tired. (**E2**) Tom wanted to have fun. (**E3**) He bought a movie ticket for Harry Potter.

where each of them can be regarded as an *event* with a verb and one to several arguments. These events, which are considered as *basic semantic units* in various NLP research (Zhang et al., 2020; Yu et al., 2020; Zhong et al., 2022; Zhang et al., 2022), convey the majority of the meaning within their respective contexts.

For human beings, the comprehension of these semantic units is found to heavily rely on our background *world knowledge* beyond contexts (Day et al., 1998). For instance, given **E1** and **E2**, we may infer that Tom might have just finished his work. Since we know watching movies is a lot of fun, we find it reasonable that Tom chose to do so (from **E2** to **E3**). We can also reason from **E3** that Tom would have to arrive at the theater before the movie started.

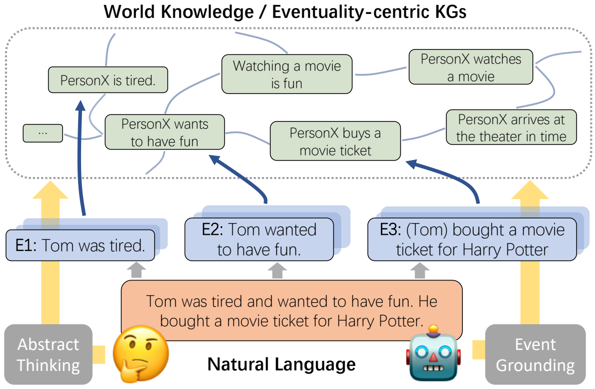

World Knowledge / Eventuality-centric KGs
PersonX is tired.
Watching a movie is fun
PersonX watches a movie
...
PersonX wants to have fun
PersonX buys a movie ticket
PersonX arrives at the theater in time
E1: Tom was tired.
E2: Tom wanted to have fun.
E3: (Tom) bought a movie ticket for Harry Potter
Tom was tired and wanted to have fun. He bought a movie ticket for Harry Potter.
Abstract Thinking
Natural Language
Event Grounding

Figure 1: Given a piece of story, our goal is to ground it to eventuality-centric KGs to retrieve contextualized background world knowledge for better narrative understanding.

To model such world knowledge on machines, most existing work fall into two paradigms. One is to implicitly model event knowledge by pretraining LMs with event-aware objectives (Yu et al., 2020; Zhou et al., 2021, 2022b,a). This paradigm, however, sacrifices transparency and explanability of reasoning in its philosophy of design. In compar-

[1] We use the linguistic term "eventuality," which includes events, states, and activities (Mourelatos, 1978; Bach, 1986). For simplicity, we use the terms "event" and "eventuality" interchangeably.

[2] Jiayang completed this work while interning at Amazon AWS AI Lab.

LREC-COLING 2024, pages 6622–6642
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

ison, another paradigm focuses on modeling the explicit symbolic event knowledge, usually in the form of eventuality-centric knowledge graphs (KGs, such as ASER (Zhang et al., 2022) and ATOMIC (Sap et al., 2019)). In this direction, how to leverage the symbolic event knowledge in these KGs for reasoning remains under-explored. The handful research here only work on a restricted format (*subject-verb-object*) of texts and could not generalize to free-texts (Li et al., 2018; Lv et al., 2020; Lee and Goldwasser, 2019; Lee et al., 2020).

In this paper, we make a step forward to examine the problem of grounding[3] free-texts to eventuality-centric KGs. This problem is non-trivial due to the distinct characteristics of events, including:

1. *Difficulty in representing events.* First, events appear entangled in texts. They tend to share arguments with other events in the same context (e.g., **E1** and **E2**). Second, when separated from the context, events lose co-reference information in the argument level. For instance, it is hard to discern whether the pronoun "he" in event **E3** refers to "Tom" in **E1** and **E2** or not.
2. *Sparsity of events.* Events are sparse in natural language. For instance, by adding or removing details, one could paraphrase **E3** into infinite events describing the same scenario, such as *"he purchased a ticket online for the latest Harry Potter"* or *"he booked a ticket"*. Given the incomplete nature of eventuality-centric KGs, matching arbitrary events to KGs has rather high failure rate.

To tackle the above problems, we propose the very first framework to explicitly ground free-texts to eventuality-centric KGs. For the *event representation* problem, we equip semantic parsing based event extraction with an event normalization module, which separates events from contexts while preserving co-reference information. Motivated by humans' abstract thinking process, we propose a partial information extraction approach to tackle the *sparsity* problem. This approach conceptualizes events into multiple partial events by omitting argument details. Interestingly, we empirically demonstrate that these solutions significantly alleviate the sparsity problem. Further, we ground the partial events to KGs to get joint reasoning subgraphs. Subsequently, we employ two common graph reasoning models to leverage this knowledge. In addition to a model based on graph neural networks (GNN), we also utilize a model based on a large language model (LLM). Experimental results on three narrative reasoning tasks show that our framework consistently outperforms current state-of-the-art models. Lastly, we provide a qualitative study to showcase how our approach can provide interpretable evidence for model predictions.

To summarize, the paper's contributions are[4]:

1. We develop an initial formulation for the problem of grounding free-texts to eventuality-centric KGs.

2. We propose EventGround, a systematic approach, to solve the *event representation* and *sparsity* problems, and perform narrative reasoning based on the grounded information.

3. Experimental results show that our approach outperforms strong baselines and achieves new state-of-the-art performance on three datasets, while providing human-interpretable evidence.

2. Related work

Reasoning on narratives is a fundamental task (Mostafazadeh et al., 2016; Li et al., 2018; Mori et al., 2020; Jiayang et al., 2023) and has attracted much interest in the NLP community. The most crucial problem in narrative reasoning is modeling the relationship between events, which often requires background world knowledge (Day et al., 1998; Mostafazadeh et al., 2016). Many large scale knowledge graphs (KGs) such as ATOMIC (Sap et al., 2019), ConceptNet (Speer et al., 2017), ASER (Zhang et al., 2020, 2022) and GLUCOSE (Mostafazadeh et al., 2020) have been constructed in recent years. Current solutions on leveraging the knowledge in these resources can be coarsely categorized into the following two groups. An overview of the two paradigms is presented in Figure 6.

The knowledge model paradigm leverages external KGs by pretraining LMs with carefully designed objectives. Most existing knowledge enhanced LMs focused on using entity-centric KGs (Zhang et al., 2019; Peters et al., 2019; Févry et al., 2020; Verga et al., 2020; Xiong et al., 2020; Sun et al., 2019b, 2021; Joshi et al., 2020). As for using external event knowledge, the knowledge model paradigm focus on finetuning language models on event-aware KGs, such as event-pair relation modeling (Bosselut et al., 2019; West et al., 2021; Zhou et al., 2021), whole event recovering/masking (Zhou et al., 2022b; Yu et al., 2020), and correlation-based event ranking (Zhou et al., 2022a).

[3] Here, the term "grounding" refers to a process similar to "linking" used in "entity linking", where the target is the eventuality-centric KGs.

[4] The code and data are available at `https://github.com/HKUST-KnowComp/EventGround`.

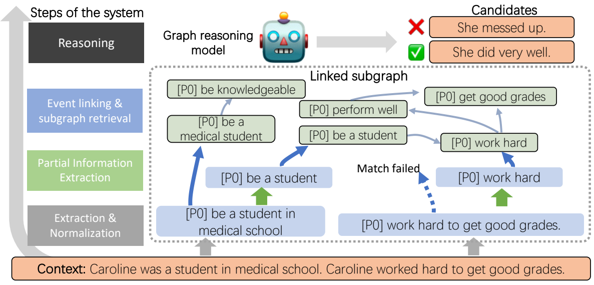

Steps of the system
Reasoning
Event linking & subgraph retrieval
Partial Information Extraction
Extraction & Normalization
Graph reasoning model
Candidates
She messed up.
She did very well.
Linked subgraph
[P0] be knowledgeable
[P0] be a medical student
[P0] perform well
[P0] be a student
[P0] get good grades
[P0] work hard
[P0] be a student
Match failed
[P0] work hard
[P0] be a student in medical school
[P0] work hard to get good grades.
Context: Caroline was a student in medical school. Caroline worked hard to get good grades.

Figure 2: An overview of EventGround.

The retrieval-and-integration paradigm, in contrast, explicitly retrieves triples or subgraphs from external KGs. Recent work on reasoning with external KB and texts have explored grounding entities to KGs, such as (Sun et al., 2018, 2019a; Xiong et al., 2019; Min et al., 2019; Lee et al., 2021), and (Lin et al., 2019; Feng et al., 2020; Yasunaga et al., 2021) in open-domain QA, commonsense QA, and narrative reasoning. However, most of them ground to entity-centric KGs (e.g. the entity part of ConceptNet (Speer et al., 2017)), which have little or no event knowledge. Although some (Lv et al., 2020; Lee and Goldwasser, 2019; Lee et al., 2020; Li et al., 2018) on script reasoning have investigated the usage of events, their methods are restricted to the "subject-verb-object"-like structured texts in the MCNC task, and have difficulty extending to general free-texts. In comparison, we tackle the more difficult problem of grounding events in free-texts to eventuality-centric KGs. The wide adoption of AI critically needs explainability (Hoffman et al., 2018). Thus, despite the appeal of a simpler pipeline (aided by the availability of large LMs), this work extends the retrieval-and-integration paradigm for grounding free-texts to eventuality-centric KGs for narrative reasoning.

As opposed to event grounding, a similar term "event linking" has been used in the literature, where they either focus on cross-document event co-reference (Nothman et al., 2012; Krause et al., 2016), or event co-reference to Wikipedia pages (Yu et al., 2021). Moreover, their "event" refers to specific happenings such as "World War II" rather than the more general eventualities in this work.

3. EventGround: Grounding free-texts to eventuality-centric knowledge graphs

In this section, we present our proposed framework, EventGround. An overview is presented in Figure 2. To tackle the *event representation* problem, we equip semantic parsing based event extraction (§ 3.1.1) with an event normalization module (§ 3.1.2) to separate events from contexts while preserving their arguments' co-reference information. We solve the *sparsity* problem by with a partial information extraction approach (§ 3.1.3). We empirically prove that these solutions largely alleviate the sparsity problem in § 4.5. At the end of this section, we discuss grounding the partial events to KGs to obtain joint reasoning subgraphs in § 3.2, and present both the GNN-based and LLM-based reasoning models in § 3.3.

3.1. Obtaining events

The proposed event acquisition pipeline includes event extraction (§ 3.1.1), normalization (§ 3.1.2) and partial information extraction (§ 3.1.3).

3.1.1. Event extraction

As shown in the previous example, events do not naturally exist in free texts. Instead, an event may share arguments with (e.g., **E1** and **E2**) or contain another event. Therefore, a special extraction step is needed to separate events from their contexts.

In this work, we consider the semantic parsing based methods to extract events from their contexts. For each piece of text $s = [s_1, s_2, \cdots, s_n]$ with n sentences, we conduct semantic role labeling (SRL) on the text to extract a series of verb-centric events $\mathcal{P} = \{p_1, p_2, \cdots, p_m\}$, where each event $p_i = (verb^i, \mathcal{A}^i)$ has a trigger $verb^i$ and a set of arguments $\mathcal{A}^i$. Each argument $a^i_j \in \mathcal{A}^i$ has a semantic role $role(a^i_j) \in \{ARG_0, ARG_1, \cdots, ARG_M\}$[5]. In addition, we define the operator $text(p_i)$ to obtain the text of p_i.

3.1.2. Event normalization

It is noteworthy that the extracted events suffer from the loss of co-reference information. For instance, here are three events extracted from a text:[6]

(1) The general had some wine at a party.
(2) He felt sleepy.
(3) He said goodbye to them.

where "*the general*" and "*he*" refer to the same person, while "*them*" refers to another group of people. A system would not be aware of this co-reference

[5] The annotation follows the PropBank (Palmer et al., 2005) annotation guideline, where the numbered arguments in general correspond to the roles: ARG_0-agent; ARG_1-patient; ARG_2-instrument, benefactive, attribute; ARG_3-starting point, benefactive, attribute; ARG_4-ending point; ARG_M-modifier.

[6] For simplicity, we do not explicitly show verbs and arguments of the events. All the words in events are lemmatized in our pipeline, which is not shown in the examples.

relationship without contexts. This makes it difficult to reason on the extracted events.

Motivated by previous work (Sap et al., 2019; Fang et al., 2021) in constructing commonsense KGs, we replace tokens referring to people with special tokens[7] (e.g., "`[P0]`," "`[P0's]`," "`[P1]`," where different numbers refer to different people). For instance, "*the general*" and "*he*" are replaced by "`[P0]`," and "*them*" is replaced by "`[P1]`." Through this normalization process, the co-reference information is preserved:

(1) `[P0]` had some wine at a party.
(2) `[P0]` felt sleepy.
(3) `[P0]` said goodbye to `[P1]`.

In addition, the normalization helps reduce event sparsity by removing details in the personal words. For instance, "*the general felt sleepy*," "*Joe felt sleepy*," and "*he felt sleepy*" will all be normalized to "`[P0]` *felt sleepy*." This increases their probability of being successfully grounded to KGs.

3.1.3. Partial information extraction

The normalized events retain rich contextual details from the original texts, which are important for downstream reasoning processes. However, the sparsity of events can pose challenges in event grounding, especially when most knowledge graphs (KGs) are far from complete (Min et al., 2013; Xiong et al., 2019). For example, a KG is more likely to include a general event like "*a person is drinking*" than "*the general is drinking Sauvignon Blanc on the balcony*," because the former is more general and likely to occur frequently.

Humans strongly depend on conceptual abstraction to identify similarities among seemingly different concepts and events, which enables generalizations to unfamiliar situations (Murphy, 2004). For instance, we can learn that there is common abstraction between "*buy a ticket for 'Avengers'*" and "*buy a ticket for 'Harry Potter'*," and that how the commonality "*buy a ticket*" relates to other events such as we should "*arrive at the theater in time*". With this concept in mind, we use a partial information extraction (PIE) phase to obtain partial events as a method of controllable abstraction.

The partial information extraction is based on the importance of event arguments in semantic role labeling (Palmer et al., 2005). For instance, ARG_0 and ARG_1 have the highest importance as they usually specify the subject and objects. In contrast, the modifier argument ARG_M express the least information, as it usually defines additional constraints of the predicate, such as when and where the event happens. Specifically, we propose to drop the event arguments in the descending order of their importance. For event $p = (verb, \mathcal{A})$ with $|\mathcal{A}| = k$, we iteratively drop its argument $a_j \in \mathcal{A}$, such that the roles of dropped arguments follow the order: (1) ARG_M[8], (2) ARG_2, ARG_3, ARG_4, (3) ARG_1 and (4) ARG_0. The partial information extraction on event set $\mathcal{P}$ results in a new set of partial events $\mathcal{P}_{abs}$, where $\mathcal{P}_{abs} = \{\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_m\}$. Each element $\hat{p}_i = [p_i^0, p_i^1, \cdots]$ is a sequence of partial events correspond to event $p_i \in \mathcal{P}$ ($p_i^0 = p_i$).

Below is an example of $\hat{p}$:

p^0 ARG0: <u>`[P0]`</u> V: <u>evacuated</u> ARG2: <u>to a relative 's house</u> ARGM: <u>last night</u>.

p^1 ARG0: <u>`[P0]`</u> V: <u>evacuated</u> ARG2: <u>to a relative 's house</u>.

p^2 ARG0: <u>`[P0]`</u> V: <u>evacuated</u>.

p^3 V: <u>evacuated</u>.

Each time an argument is dropped, the abstract level of the partial event increases. Meanwhile, partial events on higher abstract level (e.g. p^2, p^3) are more likely to have been recorded in KGs, which alleviates the sparsity problem. In § 4.5, we empirically show that the partial information extraction improves the model performance by drastically increasing the hit rate of event grounding.

3.2. Grounding to eventuality-centric KG

In this section, we discuss the event grounding approach. In § 3.2.1, we describe how to map events to eventuality-centric KGs to get the anchor events that have the closest semantic meaning. In § 3.2.2, we describe how to retrieve grounded subgraphs based on the anchor events.

3.2.1. Event matching

Suppose we have an eventuality-centric KG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ and $\mathcal{E}$ are the node set and the edge set, respectively. Each node $v_i \in \mathcal{V}$ is an event with a text attribute $text(v_i)$. Then, for each event $p \in \mathcal{P}_{abs}$, our goal is to find the node $v \in \mathcal{V}$ (which we term as "*anchor event*") that is the most similar to p:

$$v = \arg\min_{v \in \mathcal{V}} d(p, v), \tag{1}$$

where $d(\cdot, \cdot)$ denotes the distance between events.

[7] Specifically, the spans of personal words are detected by syntactic parsing and animacy classification. We then employ the co-reference information between these spans to normalize all spans that refer to persons.

[8] We do not drop the negation (e.g., *not*, *n't*, *never*) and modals (e.g., *will*, *may*, *can*) modifier arguments, since they are crucial building blocks in discourse as revealed in the linguistics study (Jordan, 1998).

To define the similarity, previous work have explored *token-level similarity* by computing the cosine distance for TF-IDF or BM25 vectors (Lv et al., 2020). However, this method overlooks the semantics of events, and constantly fails by mapping to events with high inverse document frequency terms (e.g. "`[P0's]` *lung gets punched*" is matched with "`[P0]` *has lung cancer*"). Therefore, we turn to use *semantic similarity* to match events.

Specifically, we encode event p and v with sentence transformers (Reimers et al., 2019),[9] and compute $d(p, v)$ by the L2 distance:

$$d(p, v) = ||\text{SBERT}(text(p)), \text{SBERT}(text(v))||_2. \tag{2}$$

In practice, not every event can be successfully matched with the correct ones. We empirically set a threshold l over $d(p, v)$ to filter out the failed matches.[10] As a result, partial events in $\mathcal{P}_{abs}$ are matched to their anchor events in $\mathcal{G}$, which we denote by $\mathcal{C}$. $\mathcal{C} = \{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_m\}$, where each $\hat{c}_i$ is a sequence of anchor events matched from $\hat{p}_i$.

3.2.2. Joint subgraph construction

Knowledge subgraph retrieval Based on the anchor events from the matching results in § 3.2.1, we aim to retrieve a subgraph $\mathcal{G}_{sub} = (\mathcal{V}_{sub}, \mathcal{E}_{sub})$ from $\mathcal{G}$. Ideally, $\mathcal{G}_{sub}$ should contain the background world knowledge related to the reasoning, meanwhile cover minimal number of additional eventualities. Finding such a subgraph is essentially trying to solve an NP-complete Steiner tree problem (Garey and Johnson, 1977; Lin et al., 2019), which is intractable. As a workaround, we search for the shortest path within γ-hops between each event pair in $\{(v_a, v_b) : v_a \in \hat{c}_i, v_b \in \hat{c}_j; \hat{c}_i, \hat{c}_j \in \mathcal{C}\}$. For any path obtained, the nodes and edges along the path are added to $\mathcal{G}_{sub}$.

Joint subgraph construction Based on $\mathcal{G}_{sub}$, we construct a joint knowledge enhanced subgraph $\mathcal{G}_{joint} = (\mathcal{V}_{joint}, \mathcal{E}_{joint})$ for reasoning. Specifically, $\mathcal{G}_{joint}$ includes all the nodes and edges in $\mathcal{G}_{sub}$. In addition, we add the context events in $\mathcal{P}$ as nodes to $\mathcal{G}_{joint}$, where their grounding relation to anchor events in $\mathcal{C}$ as well as the context relation (between the previous and latter events in the order that they appear in context) are added as edges.

3.3. Graph reasoning models

The retrieved subgraphs are then used for reasoning using either a GNN-based reasoning model or an LLM-based reasoning model.

GNN-based reasoning model. We first encode the text s and node $v \in \mathcal{V}_{joint}$ using the language model representation:

$$\begin{aligned} \mathbf{v} &= f_{\mathsf{LM}}(text(v)), \\ \mathbf{s} &= f_{\mathsf{LM}}(s). \end{aligned} \tag{3}$$

Then, we employ a GNN module to perform reasoning on the joint subgraph $\mathcal{G}_{joint}$. We choose the relational graph convolutional networks (RGCN) (Schlichtkrull et al., 2018) so that the relational information in $\mathcal{G}_{joint}$ can be well modeled. Specifically, for each layer l in an L-layer GNN, the representation $\mathbf{h}_i^{(l)}$ of node $i \in \mathcal{V}_{joint}$ is updated by

$$\mathbf{h}_i^{(l+1)} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|}\mathbf{W}_r \cdot \mathbf{h}_j^{(l)}\Big), \tag{4}$$

where $\mathcal{R}$ is the set of edge types in $\mathcal{E}_{joint}$, $\mathcal{N}_r(i)$ denotes the neighborhood with relation r of node i, and $\sigma(\cdot)$ is an non-linear activation. Then, we obtain the vector representation for $\mathcal{G}_{joint}$ by pooling the hidden node embeddings from the last layer

$$\mathbf{g} = \text{Pooling}(\{\mathbf{h}_i^L : i \in \mathcal{V}_{joint}\}). \tag{5}$$

The final prediction comes from

$$p(s) \propto \textbf{MLP}(\mathbf{s} + \mathbf{g}), \tag{6}$$

where **MLP** means a multi-layer perceptron module to predict the probability of the output.

LLM-based reasoning model. We also explored fusing the eventuality knowledge subgraph $\mathcal{G}_{joint}$ into LLMs. Since LLMs only receive sequence inputs, we conduct sequentialization on subgraphs in a format similar to (Madaan and Yang, 2021; Sakaguchi et al., 2021). Using a transformation function $t(\cdot)$, a subgraph $\mathcal{G}_{joint}$ is transformed into a piece of text $s_{\mathcal{G}_{joint}}$ ($s_{\mathcal{G}_{joint}} = t(\mathcal{G}_{joint})$), which is then fed into LLM as part of the prompts. We discuss variations of $t(\cdot)$ and other details in § 4.3.

4. Experiments

4.1. Datasets

We conduct experiments on three downstream tasks on narrative reasoning. The statistics are presented in Table 1.

- **Story Cloze Test v1.0** (SCT-v1.0) was proposed by Mostafazadeh et al. (2016) to evaluate the understanding of relations between events. Given four consecutive sentences, the task is to predict the correct ending from two possible choices.
- **Story Cloze Test v1.5** (SCT-v1.5) Later, Sharma et al. (2018) introduces a new version to correct the artifacts in the previous release. For both versions, we follow the common practice (Li et al., 2019;

[9] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[10] We sample 100 matching results and empirically set l=0.65 that filters out the most failed cases.

Name	Train	Valid	Test
SCT-v1.0	1,771	100	1,871
SCT-v1.5	1,471	100	1,571
MCNC	140,331	10,000	10,000

Table 1: Statistics of datasets.

Yu et al., 2020) to randomly select 100 validation samples for validation, and use the rest for training.
- **Multiple Choice Narrative Chain** (MCNC) (Granroth-Wilding and Clark, 2016; Li et al., 2018) is a 5-way multiple choice task that requires a system to predict the ending event given its previous context event sequence.

4.2. Eventuality-centric knowledge graphs

There are eventuality-centric KGs such as ATOMIC (Sap et al., 2019), GLUCOSE (Mostafazadeh et al., 2020) and ASER (Zhang et al., 2020, 2022). In this paper, we conduct experiments on ASER. The nodes in ASER are eventualities, and the edges between them are the discourse relations (e.g. "Precedence", "Contrast" and "Reason") defined in Penn Discourse Tree Bank (Prasad et al., 2008). To enable grounding normalized events to KGs, we normalize and aggregate eventualities in the ASER-core-100 version[11] by detecting and replacing the personal words with aforementioned special tokens. The resulting normalized ASER graph contains $193k$ nodes and $6.6m$ edges.

4.3. Experimental Setup

We implement the event extractor with AllenNLP SRL tools.[12] To normalize the events, the syntactic parser, animacy classifier, and co-reference tools are from Stanford CoreNLP.[13] In our implementation of the event matching module, due to the large scale of $|\mathcal{V}|$, we employ Faiss (Johnson et al., 2019) to accelerate the similarity search. When retrieving subgraph, we set the shortest path length limit γ to 3, meaning that there are at most 2 intermediate nodes between any two anchor nodes along the path.

We implement the GNN-based reasoning model with Deep Graph Library (Wang et al., 2019) and Huggingface-Transformers (Wolf et al., 2020). For finetuning the supervised models, we conduct grid-search over model hyper-parameters. The number of convolutional layers L are searched within $\{2, 3, 4\}$, and the hidden size of convolutional layers $\in \{64, 128, 256, 512\}$. For relational convolutional layers, the number of bases is searched within $\{-1, 10, 30\}$. We use the Adam (Kingma and Ba, 2015) optimizer with cosine learning rate schedule to optimize the models. The learning rate is set to $1e-5$ for all the "base" models, and $5e-6$ for all the "large" models. All the experiments are run on 4 NVIDIA Tesla-V100 GPUs.

For the LLM-based reasoning model, we adopt ChatGPT (OpenAI, 2022). [14] We consider three implementations for the graph sequentialization function $t(\cdot)$: (1, DOT) using the DOT language to represent graphs (Gansner et al., 1993; Madaan and Yang, 2021; Sakaguchi et al., 2021); (2, Node & Edge) instead of using node indexing as in DOT, we try directly inputing all the nodes and edges (e.g., "`[P0] buy a boat -> [P0's] nearby marina have a race; [P2] prepare -> [P2] go to sleep; ...`"); (3, Node) only the nodes are fed into ChatGPT (e.g., "`[P0] buy a boat; [P0's] nearby marina have a race ...`"). The prompt template is: "`Event knowledge on narrative choice A:` $\{t(\mathcal{G}_{joint,A})\}$ `\n Event knowledge on narrative choice B:` $\{t(\mathcal{G}_{joint,B})\}$ `\n Question:{} \n Answer:`". As a baseline, we also test ChatGPT without the additional knowledge (denoted by "ChatGPT$_{\text{Vanilla}}$"). For SCT-v1.0, we report results on its test set (sampled 500 instances). Since the test set of SCT-v1.5 is no longer publicly available[15] at the time we ran this experiment, we report the results on its validation set. We do not report the performance on MCNC because the lengths of most instances in this set exceed the maximum input length.

4.4. Main results

The main results on the three datasets are presented in Table 2 and 12. Per-task performance comparisons are presented in Appendix A.

As shown in Table 2, when coupled with a GNN-based reasoning model, our proposed framework achieves consistent performance gain over different backbone models. Moreover, compared with existing knowledge enhanced models, we achieve SOTA performance in three narrative reasoning tasks. The knowledge also benefits our LLM-based reasoning model (Table 12), especially when the subgraphs are transformed using the "Node & Edge" setting.

4.5. Ablation study

We conduct ablation studies to investigate the contribution of each component in our framework.

[11] We obtain the core-100 version by filtering out nodes with frequency lower than 100 from ASER-core: https://hkust-knowcomp.github.io/ASER/

[12] https://github.com/allenai/allennlp

[13] https://stanfordnlp.github.io/CoreNLP/

[14] The evaluation is performed in September 2023.

[15] https://competitions.codalab.org/competitions/15333

Method	Size	SCT-v1.0	SCT-v1.5	MCNC
(Lv et al., 2020)	125M	-	-	58.66
(Zhou et al., 2021)	469M	-	-	63.62
CoCoLM (Yu et al., 2020)	355M	97.70	-	-
TransBERT (Li et al., 2019)	355M	91.80	90.30	-
EventBERT (Zhou et al., 2022a)	355M	-	91.33	63.50
ClarET (Zhou et al., 2022b)	400M	-	91.18	64.61
RoBERTa-base (Liu et al., 2019)	125M	92.75±0.24	87.14±0.39	61.28±0.14
RoBERTa-large (Liu et al., 2019)	355M	96.74±0.08	92.34±0.06	63.01±0.12
DeBERTa-large (He et al., 2021)	354M	98.13±0.34	94.67±0.25	65.67±0.13
EventGround-RoBERTa-base	126M	93.30±0.11	87.65±0.13	62.11±0.07
EventGround-RoBERTa-large	358M	97.10±0.13	92.86±0.05	63.96±0.15
EventGround-DeBERTa-large	358M	**98.29±0.16**	**95.01±0.32**	**66.05±0.12**

Table 2: Main results on the benchmarks. Numbers are mean and standard deviation of accuracy (%) over three runs. Underlined results are the previous state-of-the-art performance.

Model	SCT-v1.0	SCT-v1.5
Random	50.00	50.00
ChatGPT$_{\text{Vanilla}}$	77.80	77.00
ChatGPT$_{\text{DOT}}$	67.80	69.00
ChatGPT$_{\text{Node}}$	72.00	**78.00**
ChatGPT$_{\text{Node \& Edge}}$	**79.60**	**78.00**

Table 3: ChatGPT evaluation results (accuracy %). We report the model performance when (1) ChatGPT$_{\text{Vanilla}}$: no knowledge is provided; (2) ChatGPT$_{\text{DOT}}$, ChatGPT$_{\text{Node}}$, and ChatGPT$_{\text{Node \& Edge}}$: the knowledge subgraphs are transformed into sequences as part of the inputs.

	EventGround$_{\text{RB}}$	EventGround$_{\text{BB}}$
w/o know.	92.75±0.24	83.63±1.16
w/o extract.	91.86±0.21	83.74±0.38
w/o norm.	92.43±0.46	83.98±0.87
w/o PIE	92.81±0.32	83.88±1.40
- ARGM	93.17±0.25	84.79±1.37
- ARG2,3,4	93.03±0.49	84.53±0.60
- ARG1	**93.30±0.11**	**85.78±0.74**

Table 4: Effect of event extraction, normalization and partial information extraction (PIE). The mean and standard deviation of accuracies on SCT-v1.0 are reported, where "RB" and "BB" refer to RoBERTa-base and BERT-base versions.

4.5.1. Effect of event extraction, normalization, and partial information extraction

As shown in Table 4, we ablate the event extraction ("w/o extract."), the event normalization ("w/o norm.") and the partial information extraction ("w/o PIE" and "- ARGX") respectively. Specifically, when ablating the event extraction module, we instead use the whole sentence for event grounding. When ablating the event normalization part, we skip the normalization step, and use the raw events for grounding. For partial information extraction, we drop event arguments in the order described in § 3.1.3, where the highest level ("- ARG1") contains all the partial events in the previous levels. The baseline ("w/o know.") shows the results of vanilla language models, which do not leverage any external knowledge.

We have several observations. First, the event extraction and normalization steps are necessary. When removed, the performance relative to the baseline does not improve, or even drops. Second, the partial information extraction step is crucial. By only taking the first level of partial events (removing modifier arguments), we have seen considerable performance gain. The model reaches its best performance after dropping ARG1.

In § 3, we discuss the *sparsity* of events. Here, we conduct both automatic and human evaluation to discuss how our method contribute to the alleviation of sparsity.

• **Automatic Evaluation** (Figure 3) We analyze by automatic measures: (1) the average L2 distance $\bar{d}$ in event matching (§ 3.2.1), and (2) the percentage of events considered as successful match, i.e. with L2 distance below $l = 0.65$ (hit rate).

• **Human Evaluation** (Table 5, Figure 4) We evaluate the matching results by human annotation. Three domain experts are asked to annotate whether event matching is successful for 50 stories (∼500 events) randomly sampled from the validation set of SCT v1.0. The Fleiss's Kappa value is 0.7414. We obtain ground-truth labels by majority vote, and present the accuracy of different event matching methods in Table 5. To investigate the effect of the threshold l used in § 3.2.1, we visualize F1 scores under different threshold values in Figure 4.

We can observe that: 1) Directly matching sentences to KGs (w/o extract.) has rather low performance, which necessitates the event extraction stage. 2) The event normalization step drastically improves the matching performance. Removing normalization step can decrease the accuracy by up to 76.7%. 3) In general, the matching perfor-

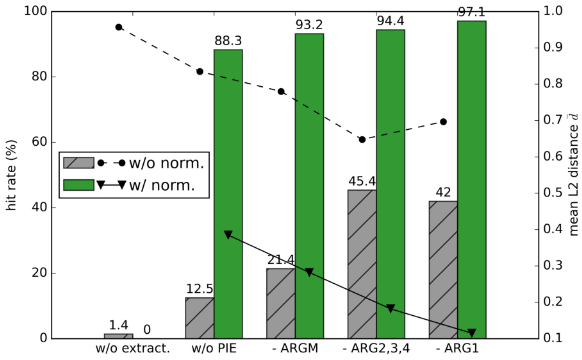


Figure 3: A comparison on the event grounding performance under different settings. The bar plot (with y-axis on the left) shows the percentage hit rate of event matching. The lines show the average L2 distance $\bar{d}$. We do not conduct normalization for “w/o extract.”.

	w/o norm.	w/ norm.
w/o extract.	4.7	-
w/o PIE	7.5	37.5
- ARGM	10.0	56.2
- ARG2,3,4	14.6	73.4
- ARG1	9.9	86.6

Table 5: Human evaluation for the accuracy of event matching (%).

mance gradually increases as the abstract level increases. 4) The Pearson’s r between automatic and human evaluation results is 0.8977, indicating thresholding on $L2$ distance is a reasonable way to automatically filter out poorly matched events. Moreover, from Figure 4, we learn that event extraction, normalization, and partial information extraction improve not only performance but also robustness of event matching. Notably, our main model (w/ norm. -ARG1) has much higher success rate than the other models, and it is meanwhile insensitive to the tuning of threshold l.

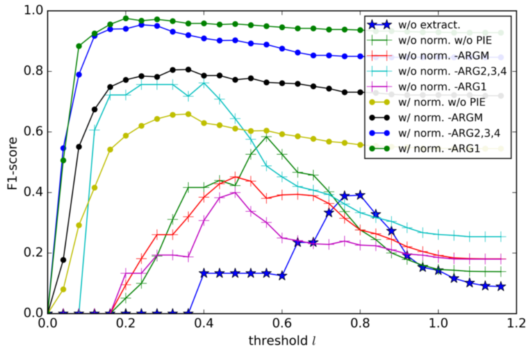


Figure 4: The F1-score to threshold curves. They reflect the event matching performance under different threshold l.

Model	Type	w/o know.	w/ know.
BERT	base	83.63±1.16	85.78±0.74
	large	88.85±0.23	90.49±0.41
RoBERTa	base	92.75±0.24	93.30±0.11
	large	96.74±0.08	97.10±0.13
DeBERTa	base	96.03±0.17	96.38±0.14
	large	98.13±0.24	98.29±0.16

Table 6: Effect of different text encoders. Three backbone language models BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2021) are tested on SCT-v1.0.

		L-layer	
n-hidden	conv.	2	3
128	RGCN	93.30±0.11	92.97±0.17
	GIN	92.93±0.37	92.57±0.24
	GCN	92.95±0.10	93.16±0.22
256	RGCN	93.14±0.20	93.12±0.17
	GIN	93.05±0.42	92.41±0.31
	GCN	92.94±0.13	92.86±0.21

Table 7: Effect of different GNN settings on SCT-v1.0.

4.5.2. Effect of model structure

We test the GNN-based reasoning model performance with different backbone text encoders (Table 6). Compared with the baselines (“w/o know.”), our framework consistently improves performance across different versions of LMs.

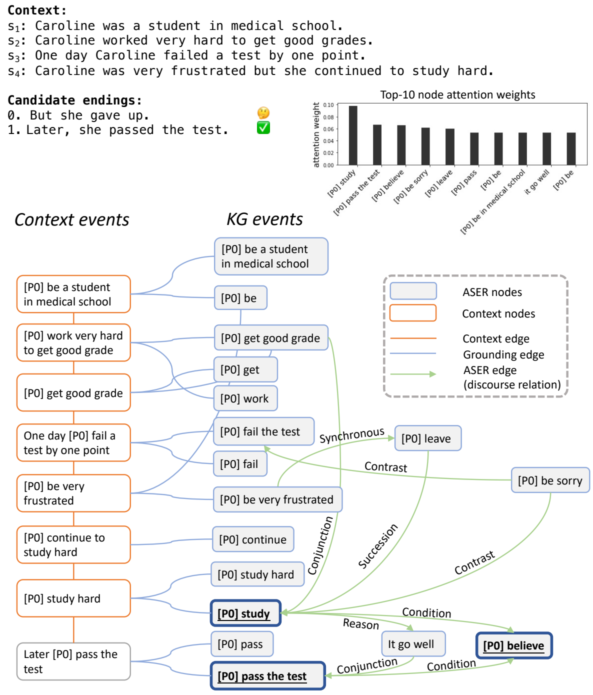
We also investigate the effect of different GNN configurations in Table 7. Apart from the relational convolutional layers (RGCN (Schlichtkrull et al., 2018)), we additionally test GIN (Xu et al., 2018) and GCN (Kipf and Welling, 2016), which do not model the edge type information. We can observe that RGCN outperforms GIN and GCN under the same settings. This indicates the discourse relation knowledge in ASER is beneficial for narrative reasoning.

We evaluate the LLM-based reasoning model under different graph sequentialization settings (Table 12). It is noteworthy that ChatGPT faces difficulties in understanding the knowledge represented in DOT language, resulting in a performance drop of approximately 10%. One possible reason for this is that the model was not trained to comprehend such structured representations. Additionally, providing only node information to the model does not yield significant benefits. The model demonstrates improved performance when using the "Node & Edge" representation of graphs.

4.6. Case study

A running example is presented in Figure 5. The top three nodes that our model focuses on are “[P0] study,” “[P0] pass the test,” and “[P0] believe.”

They are highly related to the correct candidate ending 1. Also note the path ("[P0] study," *Reason*, "it go well," *Conjunction*, "[P0] pass the test") could be explained as the causal story: Someone studies hard, so it (the learning, or the exam) goes well, and he/she passes the test.



```
Context:
s1: Caroline was a student in medical school.
s2: Caroline worked very hard to get good grades.
s3: One day Caroline failed a test by one point.
s4: Caroline was very frustrated but she continued to study hard.

Candidate endings:
0. But she gave up.                    🤔
1. Later, she passed the test.         ✅
```

Figure 5: An example from SCT-v1.0. The top-10 node attention weights are shown in the barplot. The top-3 nodes are **bolded and underlined** .

5. Conclusion

We point out two critical problems on grounding free-texts to eventuality-centric KGs, namely the *event representation* and *event sparsity* problems. We propose a simple while effective approach, EventGround, to address these problems and to leverage the retrieved graph knowledge for narrative reasoning. Empirical results demonstrate its consistent performance improvement. Further investigation reveals that the normalization and partial information extraction components drastically improve the grounding performance by alleviating event sparsity.

Limitations

In event normalization, we only normalize personal words in event as it is the most common spans that worth normalization, normalization of other type of information are not considered, which we leave for future work. When grounding to event-centric KGs, we consider finding the shortest paths to retrieve the knowledge subgraph due to high computational complexity of solving the Steiner tree problem. Other retrieval methods (e.g. reinforcement learning based) could also be considered.

Acknowledgements

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong. We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

Emmon Bach. 1986. The algebra of events. *Linguistics and philosophy*, pages 5–16.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4762–4779. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797.

Chunkit Chan and Tsz Ho Chan. 2023. Discourse-aware prompt for argument impact classification. In *Proceedings of the 15th International Conference on Machine Learning and Computing, ICMLC 2023, Zhuhai, China, February 17-20, 2023*, pages 165–171. ACM.

Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song.

2023a. Chatgpt evaluation on sentence level relations: A focus on temporal, causal, and discourse relations. *CoRR*, abs/2304.14827.

Chunkit Chan, Xin Liu, Tsz Ho Chan, Jiayang Cheng, Yangqiu Song, Ginny Y. Wong, and Simon See. 2023b. Self-consistent narrative prompts on abductive natural language inference. *CoRR*, abs/2309.08303.

Chunkit Chan, Xin Liu, Jiayang Cheng, Zihan Li, Yangqiu Song, Ginny Y. Wong, and Simon See. 2023c. Discoprompt: Path prediction prompt tuning for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 35–57. Association for Computational Linguistics.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614.

Yi Chen, Jiayang Cheng, Haiyun Jiang, Lemao Liu, Haisong Zhang, Shuming Shi, and Ruifeng Xu. 2022. Learning from sibling mentions with scalable graph inference in fine-grained entity typing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2076–2087.

Jiayang Cheng, Haiyun Jiang, Deqing Yang, and Yanghua Xiao. 2021. A question-answering based framework for relation extraction validation. *arXiv preprint arXiv:2104.02934*.

Li Cui, Deqing Yang, Jiayang Cheng, and Yanghua Xiao. 2021a. Incorporating syntactic information into relation representations for enhanced relation extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 416–428. Springer.

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021b. Refining sample embeddings with relation prototypes to enhance continual relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 232–243.

Richard R Day, Julian Bamford, Willy A Renandya, George M Jacobs, and Vivienne Wai-Sze Yu. 1998. Extensive reading in the second language classroom. *RELC Journal*, 29(2):187–191.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, and Junwen Duan. 2019. Event representation learning enhanced with external commonsense knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4894–4903.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021. Discos: Bridging the gap between discourse knowledge and commonsense knowledge. In *Proceedings of the Web Conference 2021*, pages 2648–2659.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4937–4951. Association for Computational Linguistics.

Emden R Gansner, Eleftherios Koutsofios, Stephen C North, and K-P Vo. 1993. A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230.

Michael R Garey and David S. Johnson. 1977. The rectilinear steiner tree problem is np-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of closed-source large language model. *CoRR*, abs/2305.12870.

Cheng Jiayang, Lin Qiu, Tsz Chan, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, et al. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11518–11537.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Michael P Jordan. 1998. The power of negation in english: Text, context and relevance. *Journal of pragmatics*, 29(6):705–752.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Jan Kocon, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydlo, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocon, Bartlomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Milkowski, Marcin Oleksy, Maciej Piasecki, Lukasz Radlinski, Konrad Wojtasik, Stanislaw Wozniak, and Przemyslaw Kazienko. 2023. Chatgpt: Jack of all trades, master of none. *CoRR*, abs/2302.10724.

Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. 2016. Event linking with sentential features from convolutional neural networks. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 239–249.

I-Ta Lee and Dan Goldwasser. 2019. Multi-relational script learning for discourse relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4214–4226.

I-Ta Lee, Maria Leonor Pacheco, and Dan Goldwasser. 2020. Weakly-supervised modeling of contextualized event embedding for discourse relations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4962–4972.

I-Ta Lee, Maria Leonor Pacheco, and Dan Goldwasser. 2021. Modeling human mental states with an entity-based narrative graph. *arXiv preprint arXiv:2104.07079*.

Haoran Li, Yulin Chen, Jinglong Luo, Yan Kang, Xiaojin Zhang, Qi Hu, Chunkit Chan, and Yangqiu Song. 2023a. Privacy in large language models: Attacks, defenses and future directions. *CoRR*, abs/2310.10383.

Haoran Li, Dadi Guo, Donghao Li, Wei Fan, Qi Hu, Xin Liu, Chunkit Chan, Duanyi Yao, and Yangqiu Song. 2023b. P-bench: A multi-level privacy evaluation benchmark for language models. *CoRR*, abs/2311.04044.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Zhongyang Li, Xiao Ding, and Ting Liu. 2019. Story ending prediction by transferable bert. *arXiv preprint arXiv:1905.07504*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839.

Xin Liu, Jiayang Cheng, Yangqiu Song, and Xin Jiang. 2022. Boosting graph structure learning with dummy nodes. In *International Conference on Machine Learning*, pages 13704–13716. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shangwen Lv, Fuqing Zhu, and Songlin Hu. 2020. Integrating external event knowledge for script learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 306–315.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER.

In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5721–5732. Association for Computational Linguistics.

Aman Madaan and Yiming Yang. 2021. Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. 2020. Finding and generating a missing part for story completion. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 156–166.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. *arXiv preprint arXiv:2009.07758*.

Alexander PD Mourelatos. 1978. Events, processes, and states. *Linguistics and philosophy*, 2:415–434.

Gregory Murphy. 2004. *The big book of concepts*. MIT press.

Joel Nothman, Matthew Honnibal, Ben Hachey, and James R Curran. 2012. Event linking: Grounding event reference in a news archive. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 671–688. Association for Computational Linguistics.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah

Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Siddarth Srinivasan, Richa Arora, and Mark Riedl. 2018. A simple and effective approach to the story cloze test. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 92–96.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019a. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019b. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *CoRR*, abs/2007.00849.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *CoRR*, abs/2110.07178.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete kbs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qagnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546.

Changlong Yu, Hongming Zhang, Yangqiu Song, and Wilfred Ng. 2020. Cocolm: Complex commonsense enhanced language model. *arXiv preprint arXiv:2012.15643*.

Xiaodong Yu, Wenpeng Yin, Nitish Gupta, and Dan Roth. 2021. Event linking: Grounding event mentions to wikipedia. *arXiv preprint arXiv:2112.07888*.

Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. 2022. Aser: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities. *Artificial Intelligence*, page 103740.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In *Proceedings of the web conference 2020*, pages 201–211.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Suyu Ge, Yuning Mao, Yizhu Jiao, Xingxing Zhang, Yichong Xu, Chenguang Zhu, Michael Zeng, and Jiawei Han. 2022. Unsupervised summarization with customized granularities. *arXiv preprint arXiv:2201.12502*.

Yucheng Zhou, Xiubo Geng, Tao Shen, Guodong Long, and Daxin Jiang. 2022a. Eventbert: A pretrained model for event correlation reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 850–859.

Yucheng Zhou, Xiubo Geng, Tao Shen, Jian Pei, Wenqiang Zhang, and Daxin Jiang. 2021. Modeling event-pair relations in external knowledge graphs for script reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4586–4596.

Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022b. Claret: Pretraining a correlation-aware context-to-event transformer for event-centric generation and classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2559–2575.

A. Detailed experimental results

We present the detail performance comparison for SCT-v1.0 and SCT-v1.5 (in Table 8), as well as MCNC (in Table 9). Performance of the significant baselines in the corresponding tasks is presented.

Method	SCT-v1.0	SCT-v1.5
Random	50.00	50.00
(Chaturvedi et al., 2017)	77.60	-
(Mostafazadeh et al., 2016)	58.50	-
(Srinivasan et al., 2018)	76.50	-
(Yu et al., 2020)	97.70	-
(Zhou et al., 2022a)	-	91.33
(Zhou et al., 2022b)	-	91.18
(Li et al., 2019)	91.80	90.30
RoBERTa-base	92.75±0.24	87.14±0.39
RoBERTa-large	96.74±0.08	92.34±0.06
DeBERTa-large	98.13±0.34	94.67±0.25
EventGround-RB	93.30±0.11	87.65±0.13
EventGround-RL	97.10±0.13	92.86±0.05
EventGround-DL	98.29±0.16	95.01±0.32

Table 8: Results on SCT v1.0 and v1.5. Numbers are the mean and standard deviation of accuracy (%) over three runs.

Method	MCNC
Random	20.00
(Chambers and Jurafsky, 2008)	30.52
(Granroth-Wilding and Clark, 2016)	49.57
(Li et al., 2018)	52.45
(Ding et al., 2019)	56.03
(Lv et al., 2020)	58.66
(Zhou et al., 2021)	63.62
(Zhou et al., 2022a)	63.50
(Lee et al., 2020)	63.59
(Lee and Goldwasser, 2019)	63.67
(Zhou et al., 2022b)	64.61
RoBERTa-base	61.28±0.14
RoBERTa-large	63.01±0.12
DeBERTa-large	65.67±0.13
EventGround-RB	62.11±0.07
EventGround-RL	63.96±0.15
EventGround-DL	66.05±0.12

Table 9: Results on MCNC. Numbers are the mean and standard deviation of accuracy (%) over three runs.

B. Results and statistics of event extraction and grounding

Table 11 shows the detailed statistics of the event grounding and subgraph retrieval stage. It is clear that our proposed event extraction, normalization and multi-level extraction method help alleviate the event sparsity to a large extent. This not only reflects on the hit rate and mean L-2 distance during event grounding stage, but also in their retrieved graphs statistics.

Table 10 shows the performance comparison between semantic similarity based matching (which we used) and the token-level similarity matching. It is clear from the table that the token-level based similarity matching, such as tf-idf, fails to perform as good as the semantic based matching.

Note that, the information extraction here is fundamentally different from the entity-centric line of work (Cui et al., 2021b,a; Chen et al., 2022), as our setting involves decomposition and semantic similarity computations over text snippets.

	RoBERTa	BERT
Baseline (w/o know.)	92.75±0.24	83.63±1.16
Token-level similarity (tf-idf)	92.84±0.27	84.27±0.73
Semantic similarity (SBERT)	93.30±0.11	85.78±0.74

Table 10: Performance comparison between baseline, token-level similarity based event matching, and semantic similarity based event matching.

C. Supplementary case studies

Apart from the case study provided in Section 4.6, we additionally provide another two examples in Figure 10 and 11.

D. Annotation details

We show the annotation interface presented to the expert annotators in 12. Users are prompted to compare the event and its matched anchor, and then to give an evaluation of the quality (Successful-1 or Not-0). Since the annotation requires domain-specific knowledge, we recruited 3 student researchers within our area who volunteered to help us conduct the evaluation. The payment to annotators is higher than the local minimum wage.

E. Obtaining ChatGPT Performance

In addition to GNNs (Kipf and Welling, 2016; Xu et al., 2018; Schlichtkrull et al., 2018; Liu et al., 2022), we also evaluated large language models as graph reasoning modules. Recently, large language models (e.g., ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023)) have shown promising performance on various tasks, and have raised concerns and discussions on topics such as factuality and privacy (Wang et al., 2023; Bubeck et al., 2023; Kocon et al., 2023; Chan et al., 2023a; Jiang et al., 2023; Li et al., 2023a,b). In this paper, we

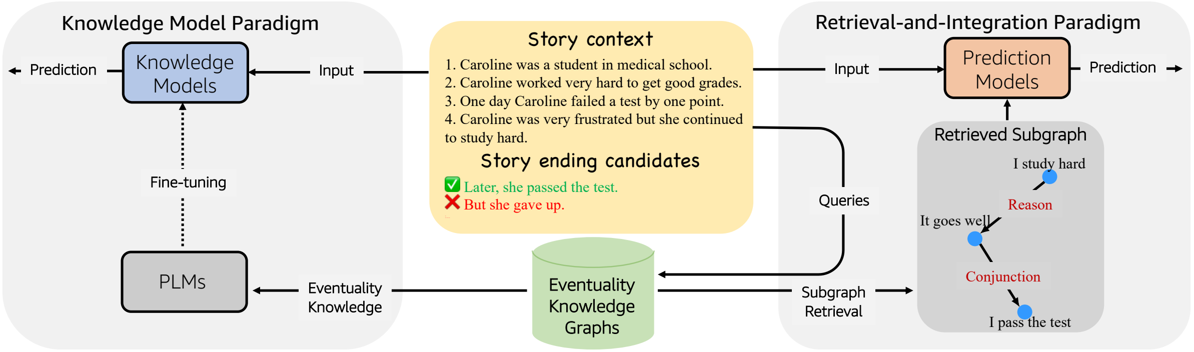

Knowledge Model Paradigm
Prediction
Knowledge Models
Input
Fine-tuning
PLMs
Eventuality Knowledge
Story context
1. Caroline was a student in medical school.
2. Caroline worked very hard to get good grades.
3. One day Caroline failed a test by one point.
4. Caroline was very frustrated but she continued to study hard.
Story ending candidates
✅ Later, she passed the test.
❌ But she gave up.
Eventuality Knowledge Graphs
Retrieval-and-Integration Paradigm
Input
Prediction Models
Prediction
Queries
Retrieved Subgraph
I study hard
Reason
It goes well
Conjunction
I pass the test
Subgraph Retrieval

Figure 6: Overview of the knowledge model paradigm (left) and the retrieval-and-integration paradigm (right). The knowledge model paradigm pretrains LMs with specially designed objectives, and then further finetunes them to adapt to downstream tasks for prediction. The retrieval-and-integration paradigm retrieves relevant subgraphs of the story context and then makes predictions according to the retrieved subgraphs.

	Event grounding		Subgraph retrieval			
	hit rate (%)	mean L2 distance $\bar{d}$	$\lvert\mathcal{V}_{sub}\rvert$	$\lvert\mathcal{E}_{sub}\rvert$	$\lvert\mathcal{V}_{joint}\rvert$	$\lvert\mathcal{E}_{joint}\rvert$
w/o extract.	1.43	0.9566	0.1235	0.1951	5.12	8.35
w/o PIE	88.28 12.50	0.3853 0.8351	13.37	36.33	21.60	67.17
- ARGM	93.22 21.43	0.2819 0.7801	22.34	74.12	30.53	109.64
- ARG2,3,4	94.38 45.44	0.1818 0.6477	28.03	93.94	36.20	134.09
- ARG1	97.12 41.97	0.1150 0.6968	63.27	281.32	71.41	330.73

Table 11: Results and statistics of event grounding and subgraph retrieval. The gray numbers are the statistics for “w/o norm.” experiments.

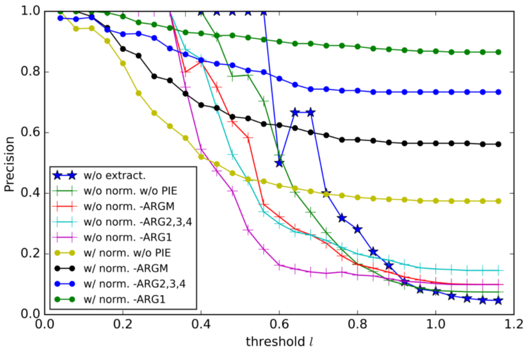


Figure 7: The Precision to threshold curves.

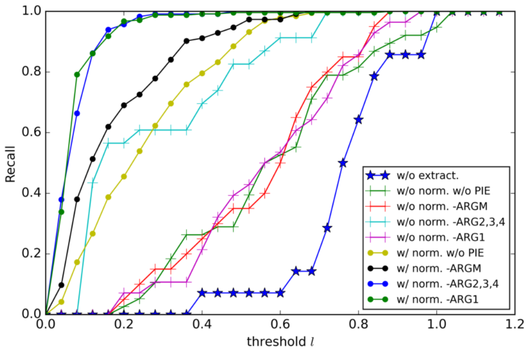


Figure 8: The Recall to threshold curves.

test ChatGPT [16] in narrative reasoning tasks with additional grounded knowledge. The zero-shot performance of large language models, which relies on the sophisticated design of templates, has shown variance across various tasks (Ma et al., 2022; Chan et al., 2023b,c; Chan and Chan, 2023). To obtain replicable and representative results, we follow Robinson and Wingate (2023); Cheng et al. (2021) to formulate the task as a multiple choice question answering problem.

[16] The evaluation is performed in September 2023 by calling ChatGPT Model (*gpt-3.5-turbo*) API .

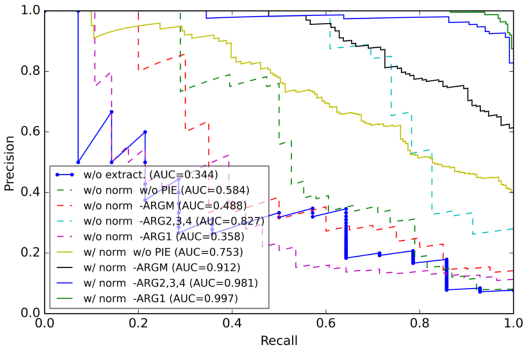


Figure 9: The Precision-Recall curve.

Model	SCT-v1.0 (%)	SCT-v1.5 (%)
Random	50.00	50.00
ChatGPT$_{\text{Prompt}}$	77.80	77.00
ChatGPT$_{\text{w/ proscript DOT}}$	67.80	69.00
ChatGPT$_{\text{w/ node}}$	72.00	**78.00**
ChatGPT$_{\text{w/ node \& edge}}$	**79.60**	**78.00**

Table 12: The performance of ChatGPT performs on the SCT-v1.0 test set (sampled 500 instances) and the SCT-v1.5 validation set. The submission upload for the SCT-v1.5 leaderboard (https://competitions.codalab.org/competitions/15333) is no longer available. Therefore, we test ChatGPT performance on the validation set. The ChatGPT template is displayed in Figure 13.

Context:
s_1: Ava needed to go shopping with her two-year old.
s_2: But she couldn’t find his shoes even after looking everywhere!
s_3: She decided she had no choice but to buy him new shoes.
s_4: She carried him into the store in order to select a new pair.

Candidate endings:
0. Ava was a neglectful mother. 🤔
1. Ava took good care of her son. ✅

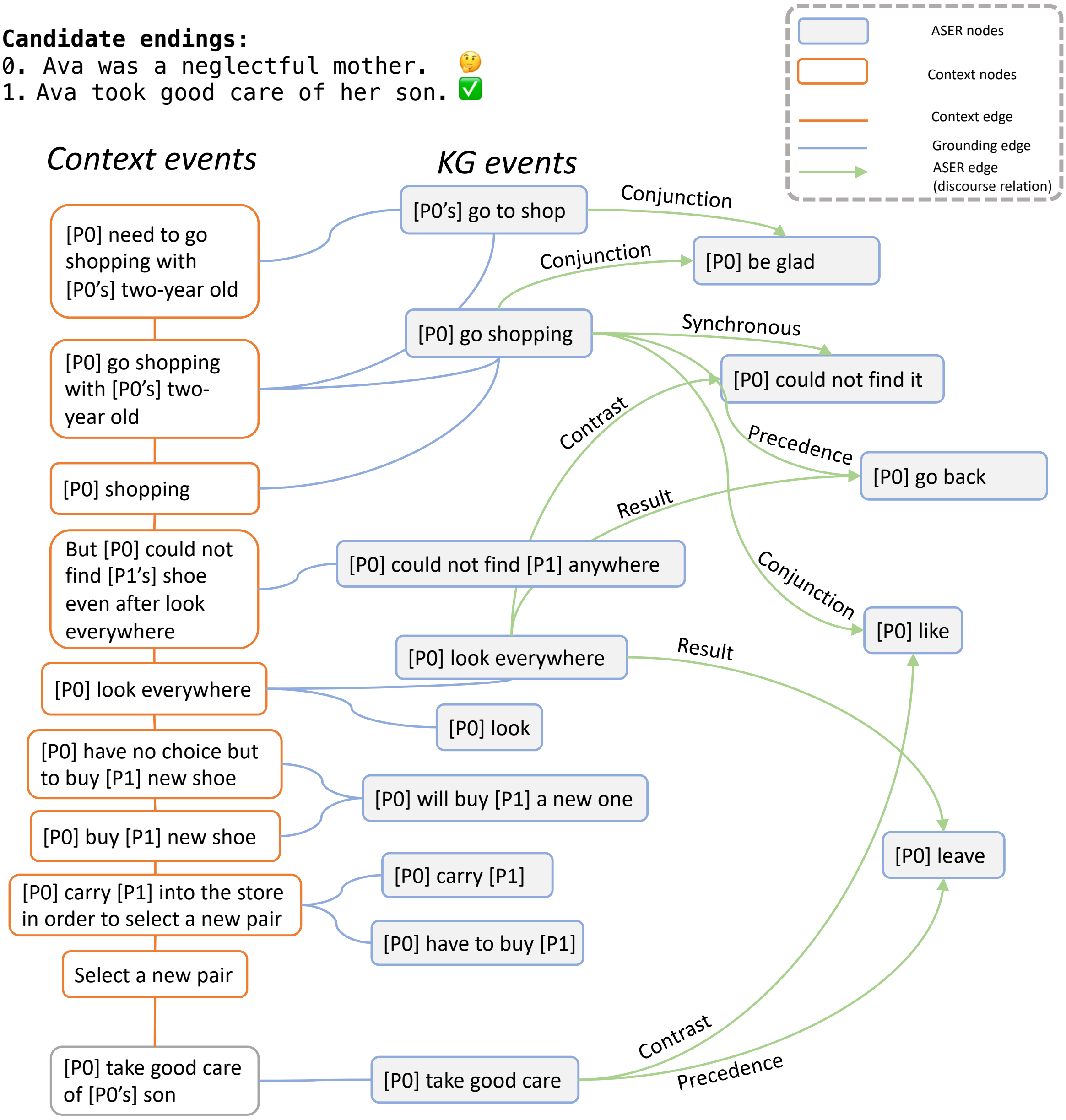
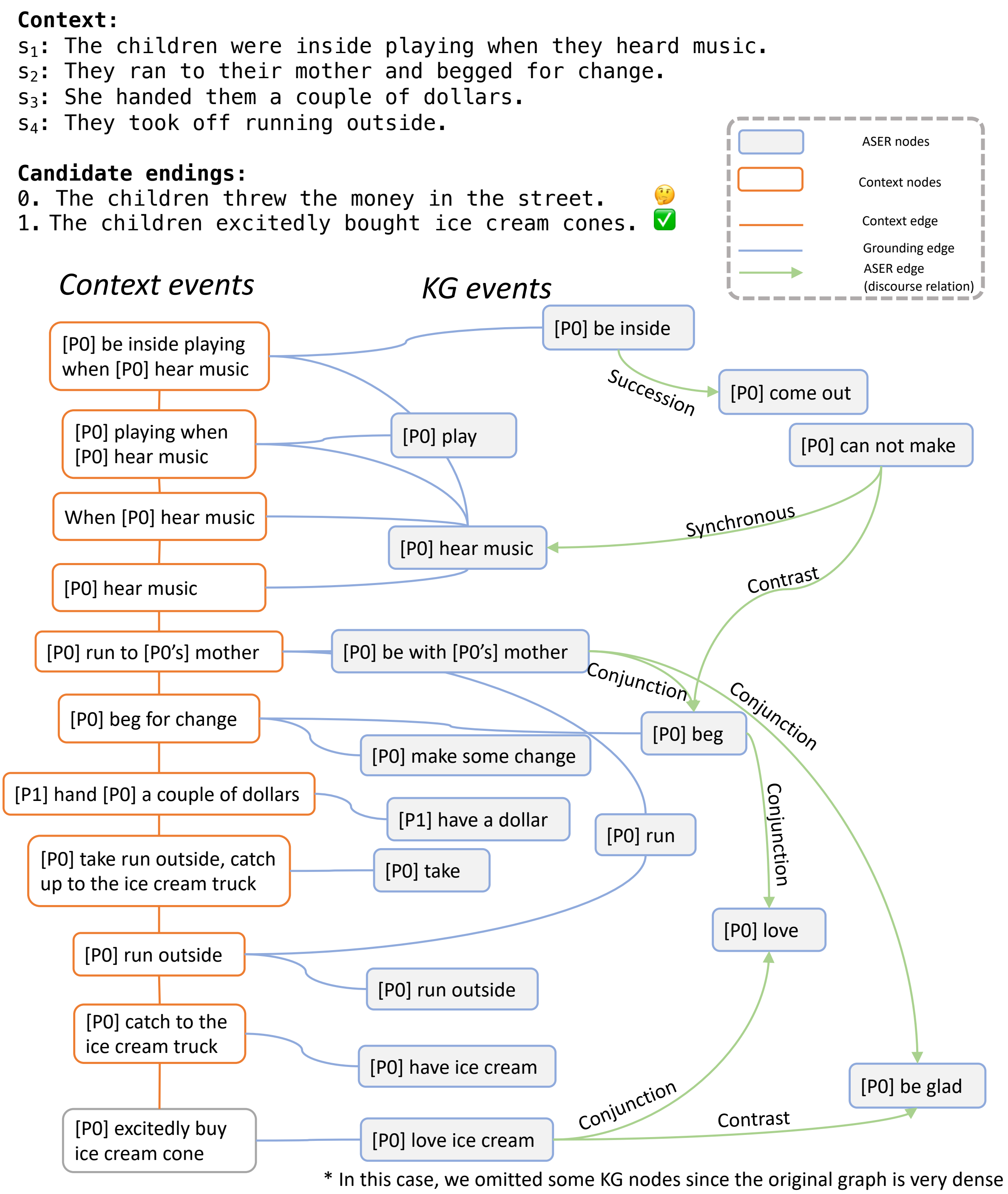


Figure 10: Supplementary case 1.



Context:

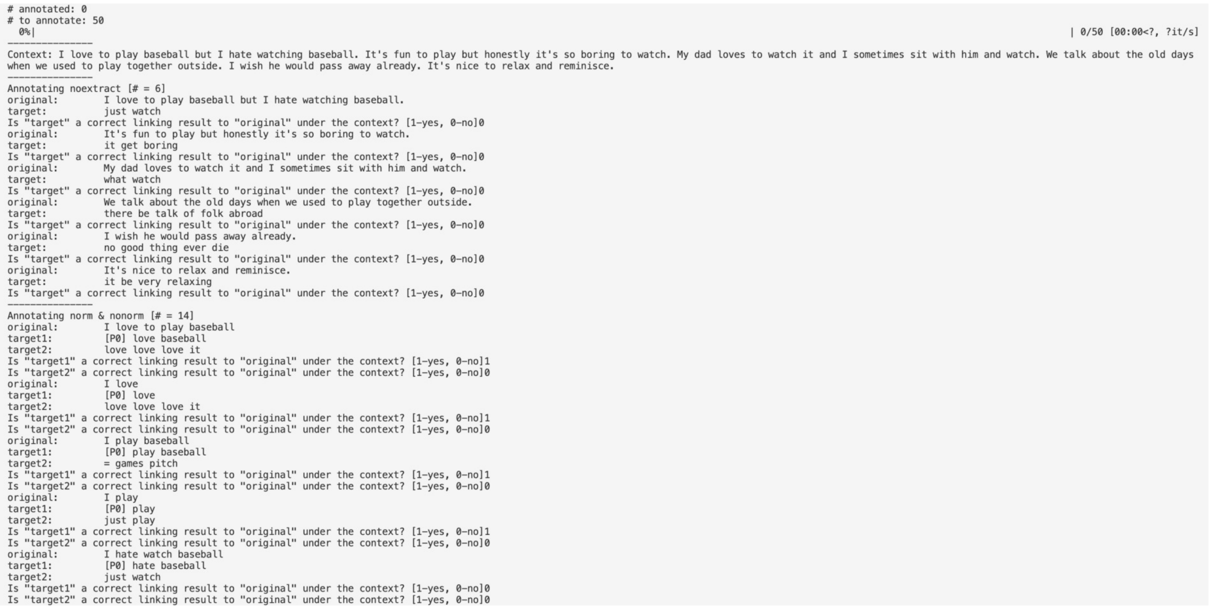
s_1: The children were inside playing when they heard music.
s_2: They ran to their mother and begged for change.
s_3: She handed them a couple of dollars.
s_4: They took off running outside.

Candidate endings:

0. The children threw the money in the street. 🤔
1. The children excitedly bought ice cream cones. ✅

* In this case, we omitted some KG nodes since the original graph is very dense

Figure 11: Supplementary case 2.



```
# annotated: 0
# to annotate: 50
  0%|                                                                                                              | 0/50 [00:00<?, ?it/s]
---------------
Context: I love to play baseball but I hate watching baseball. It's fun to play but honestly it's so boring to watch. My dad loves to watch it and I sometimes sit with him and watch. We talk about the old days
when we used to play together outside. I wish he would pass away already. It's nice to relax and reminisce.
---------------
Annotating noextract [# = 6]
original:        I love to play baseball but I hate watching baseball.
target:          just watch
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        It's fun to play but honestly it's so boring to watch.
target:          it get boring
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        My dad loves to watch it and I sometimes sit with him and watch.
target:          what watch
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        We talk about the old days when we used to play together outside.
target:          there be talk of folk abroad
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        I wish he would pass away already.
target:          no good thing ever die
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        It's nice to relax and reminisce.
target:          it be very relaxing
Is "target" a correct linking result to "original" under the context? [1-yes, 0-no]0
---------------
Annotating norm & nonorm [# = 14]
original:        I love to play baseball
target1:         [P0] love baseball
target2:         love love love it
Is "target1" a correct linking result to "original" under the context? [1-yes, 0-no]1
Is "target2" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        I love
target1:         [P0] love
target2:         love love love it
Is "target1" a correct linking result to "original" under the context? [1-yes, 0-no]1
Is "target2" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        I play baseball
target1:         [P0] play baseball
target2:         = games pitch
Is "target1" a correct linking result to "original" under the context? [1-yes, 0-no]1
Is "target2" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        I play
target1:         [P0] play
target2:         just play
Is "target1" a correct linking result to "original" under the context? [1-yes, 0-no]1
Is "target2" a correct linking result to "original" under the context? [1-yes, 0-no]0
original:        I hate watch baseball
target1:         [P0] hate baseball
target2:         just watch
Is "target1" a correct linking result to "original" under the context? [1-yes, 0-no]0
Is "target2" a correct linking result to "original" under the context? [1-yes, 0-no]0
```

Figure 12: Annotation interface in command line.

	Templates
$\text{ChatGPT}_{\text{Prompt}}$	**Question: Which choice of narrative is more reasonable? Only answer \"A\" or \"B\" only without any other words or explanations.\nA.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. Danny decided to go to sleep.**\nB.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. They prepared for the start of the race.**\nAnswer:**
$\text{ChatGPT}_{\text{Proscript DOT}}$	**Event knowledge on narrative choice A:** 0: \'[P0] buy a boat\’; … \n 12: \'[P0] go\'**\nEvent knowledge Edges for narrative choice A:** 0-->1; … 12-->6; **\nEvent knowledge on narrative choice B:** 0: \'[P0] buy a boat\’; … \n 12: \'[P2] prepare\'**Event knowledge Edges for narrative choice B:** 0-->1; … 12-->5; **\nQuestion: Which choice of narrative is more reasonable based on the event knowledge, knowledge edge and the choices? Only answer "A" or "B" only without any other words or explanations. All [P0], [P1]...etc are the people mentioned in the passage.\nA.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. Danny decided to go to sleep.**\nB.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. They prepared for the start of the race.**\nAnswer:**
$\text{ChatGPT}_{\text{Node}}$	**Event knowledge on narrative choice A:** [P0] buy a boat. … \n [P0] go**\nEvent knowledge on narrative choice B:** [P0] buy a boat. … \n [P2] prepare**\n\nQuestion: Which choice of narrative is more reasonable based on the event knowledge and the choices? Only answer "A" or "B" only without any other words or explanations. All [P0], [P1]...etc are the people mentioned in the passage.\nA.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. Danny decided to go to sleep.**\nB.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. They prepared for the start of the race.**\nAnswer:**
$\text{ChatGPT}_{\text{Node \& Edge}}$	**Event knowledge on narrative choice A:** [P0] buy a boat-->[P0\'s] nearby marina have a race; … [P0] go-->[P0] go to sleep; **\nEvent knowledge on narrative choice B:** [P0] buy a boat-->[P0\'s] nearby marina have a race; … [P2] prepare-->[P2] prepare for the start of the race; **\n\nQuestion: Which choice of narrative is more reasonable based on the event knowledge and the choices? Only answer "A" or "B" only without any other words or explanations. All [P0], [P1]...etc are the people mentioned in the passage.\nA.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. Danny decided to go to sleep.**\nB.** Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat. They prepared for the start of the race.**\nAnswer:**

Figure 13: ChatGPT Template