Deciphering Storytelling Events: A Study of Neural and Prompt-Driven Event Detection in Short Stories

Chaitanya Kirti[1], Ayon Chattopadhyay[2], Ashish Anand[2,1] and Prithwijit Guha[3,1]
[1]Centre for Linguistic Science and Technology,
[2]Department of Computer Science and Engineering,
[3]Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati, Assam, India
Email: {ckirti, ayonchattopadhyay, anand.ashish, pguha}@iitg.ac.in

***Abstract*—Detecting events in literary narratives is relatively challenging due to the presence of both realis and non-realis events. Realis events are those that actually take place. Unlike other domains, such as biomedical and news articles, events in literary narratives are not always factual. This paper introduces "*Gatha-200*", a manually annotated dataset of 200 short stories for realis events. We employed several neural models for realis event detection. The best-performing neural model reached a macro F1 score of 92.2%. Furthermore, we also evaluated the capability of the prompt-based learning approach to detect the realis events in zero-shot and few-shot settings and got promising results.**

***Index Terms*—Natural language processing, Event detection, Short stories, Prompt-based learning**

I. Introduction

Event detection is one of the important tasks for information extraction and plays crucial role in several downstream tasks including question answering. An event, indicating a change of status, is a specific occurrence at a certain time and place [1]. Previous works primarily focused on event detection in domains such as biomedical and newswire texts [2]. However, only a few studies have focused on event detection in literary texts [2].

Detecting events in literary texts such as short stories is more challenging. The challenges stem from the following reasons. The narrative structures [2] are more intricate in nature. The language in news or biomedical articles describes real-world experiences and displays relatively direct cause-and-effect linkages between events [3]. Literature is inherently a creative endeavor. In the majority of literary narratives, events are not always factual. The presence of real/realis events along with the non-realis events, poses further challenges. Realis events belong to the real world and are portrayed as existing within the imaginative realm of the narrative, occurring at a particular location and a specific moment [2]. Figure 1 illustrates the differentiation between realis and non-realis events. The primary objective of this study is to detect realis events from a specific kind of literary work - children's short stories.

Towards this objective, this study presents a novel dataset of 200 short stories annotated for realis events. The dataset is referred to as *Gatha-200*. Although the short stories are in English, they are written in the Indian context, focusing on children. The stories are taken from various collections to cover various events, writing styles, and narrations. The description of the dataset is presented in the section III. A comparative study is performed to evaluate the performance of various baseline neural models to analyse their ability to detect realis events in short stories. Finally, the prompt-based learning approach [4] is also explored in both zero-shot and few-shot settings. Our experiments on a small subset of data indicate that the prompt-based approach in a few-shot setting can give promising results and is preferable.

S1: On **hearing** this answer, the spirit **went** again and **suspended** himself on that tree.
S2: They **lived** happily after their **marriage**.
S3: They looked **terrified**.
S4: I do not know.
S5: I will help you download pictures from the internet.

Fig. 1: Snippets from the dataset showing realis (in bold) and non-realis events (underlined). S1, S2, and S3 have realis events. S4 and S5 show non-realis events. Words marked in red, green, and blue correspond to verbs, nouns, and adjectives, respectively.

Contributions of the study are:

- A benchmark dataset *Gatha-200* of 200 short stories annotated for realis events
- A comparison of various baseline neural models for identifying realis events on *Gatha-200*
- Analysis of prompt-based learning with zero-shot and few-shot settings
- Qualitative error analysis

The remaining portions of the paper are structured as follows: The literature on event detection datasets and techniques is reviewed in section II. Section III describes the dataset used to train the event detection model. Section IV describes the methodology used in the work. In section V and VI, we provide the experiment settings and results. A short description of errors is discussed in VII. The paper's conclusion is provided in section VIII.

979-8-3503-3078-6/23/$31.00

II. Related Work

Detecting event mentions in a text is the first step toward detecting the events [5]. The subsequent pair of subsections cover the related datasets and the prevailing methods in event detection.

A. Datasets

The literature contains various approaches for creating annotated corpus that have been put forward. real [6]. The lack of an annotated corpus of biological texts results in the creation of the GENIA corpus [7]. A collection of sentences sourced from news articles, encompassing reports on events, science journalism, and finance, has been annotated by Amazon Mechanical Turk workers to categorize them as either general or specific [8]. A new dataset of gun violence articles from local newspapers and television station reports with annotations was introduced with the name Gun Violence Database (GVDB) [9]. Xiang et al. in [10] extensively delves into various datasets for event detection.

While most previous works dealt with factual data, barely any have explored the event structures in literary articles. The first attempt at event detection from literary text was by generating an annotated corpora of real events from novels [2]. Most event detection tasks have used shorter sequences (sentences or phrases) with one work on a somewhat larger sequence (document) [5] to study the function and order of events.

Our work deals with the detection of events in children's short stories. Literary texts such as short stories are usually longer than fact-based articles with different event structures. Detecting real events in such imaginative narratives remains challenging. The narrative structure takes care of the child's psychology. In these stories, inanimate objects or animals talk with each other. Such scenarios make these stories different from other fact-based or literary texts.

B. Methods

In the past, event detection models were feature-based and produced a comprehensive collection of features through diligent feature engineering [11]. Recently, various deep learning techniques have been employed to extract events, reflecting the growing popularity of such methods. Specifically, CNN [12], RNN [13] LSTM [14], Tree LSTM [15], and Graph CNN [16] have been proposed in the literature. Among the deep learning models, the earliest transformer-based approach was DMBERT [17]. In this work, we compare different contemporary neural models.

In pre-trained language models, a fresh paradigm called prompt-based learning is becoming popular [18]. Unlike the pre-training and fine-tuning approach that demands significant resources and data, prompt-based methods modify downstream tasks to align more closely with the model's pre-training tasks. In their work, [19] achieve this by creating associated prompts with blank spaces and establishing a mapping from specific filled words to corresponding target categories. This transformation effectively turns various classification problems into cloze activities. In our case, we have used Prompt-Based learning generatively, where we try to label the data with real events in both zero-shot and few-shot settings.

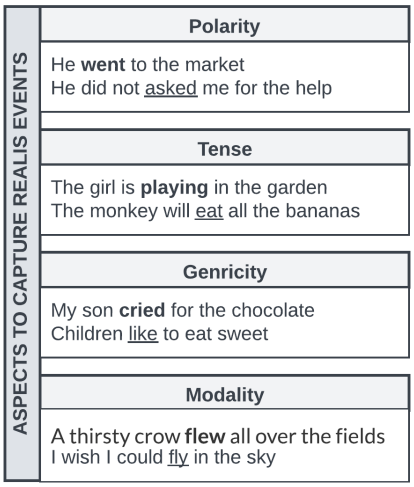

Fig. 2: Aspects to select a realis event represented with examples. Bold words represent realis events, and underlined words show non-realis events.

III. Dataset

Gatha-200 comprises 200 manually annotated short stories available in the public domain. These stories are in the Indian context written in English, mainly focusing on children. The stories are taken from various collections of interrelated fables such as Panchtantra (animal fables), Tenali-Rama, and some short stories from Hindu mythology. This variety of stories is needed to explore various events, writing styles, and narrations. The annotation guideline for our work is adapted from previous work of ACE [6] and Litbank [2]. We limited the event extent to a single word, i.e., a word represents each event. Events are usually single or phrasal verbs, but sometimes nouns, adjectives, and phrasal nouns.

In broad terms, our objective is to represent the events that are explicitly described as "actually happening" in the text. To encompass realis events, four aspects are taken into account: *Polarity*, *Tense*, *Genericity* and, *Modality* [20]. Events with positive polarity are annotated. Moreover, events in the present and past are sure events. Generic sentences do not imply the occurrence of the event. So, it needs to be specific. Also, sentences representing wishes, beliefs, etc., are not tagged as realis events. The examples of these aspects are shown in Figure 2.

For annotation purposes, two annotators who are the co-authors of this work were employed. Fifty random stories were selected, and the annotators were asked to annotate the stories following the annotation guidelines. The inter-annotator agreement (IAA) is 83.1%. IAA provides a meaningful measure of the clarity of annotation guidelines, the consistency in annotators' comprehension of these guidelines, and the overall feasibility of the annotation task [21]. One annotator annotated the remaining 150 stories, totaling the annotated stories to 200.

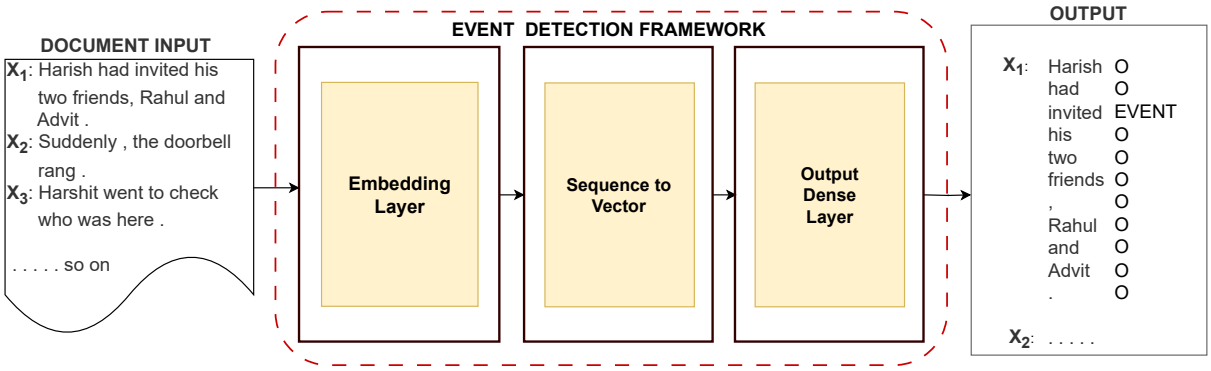

Fig. 3: Box representation of the entire event detection framework. The framework consists of three modules. (i) Embedding generation, (ii) Sequence to vector, and (iii) Output dense layer.

TABLE I: Basic statistics of the dataset.

Statistics	Count
Total words in the dataset	157,287
Total unique words in the dataset	10,558
Total sentences in the dataset	13,861
Average words per story	786
Average sentences per story	69
Average words per sentence	11
Total events in the dataset	13,171
Average events per story	66

Some basic statistics of the dataset are presented in Table I. There are 157,287 tokens in the dataset. It is evident from the table that short stories are lengthier (an average of 69 sentences per story), but the sentences in the stories are shorter (an average of 11 words per sentence). The number of events in the dataset is slightly less than the number of sentences. Several sentences have either no events or have non-realis events. Sentences with two or even three events exist in the dataset. For instance, in the sentence, *He felt so pleased with himself that he slapped his arms and shouted out.*, there are three events i.e., *"felt"*, *"slapped"*, and *"shouted"*.

IV. Methodology

The event detection task is formulated as a sequence labeling task. We perform an extensive comparative study of the performance of neural models on *Gatha-200* for the realis event detection task. In particular, a set of different variants of LSTM, BiLSTM-based neural models with different choices of pre-trained embeddings are compared. Keeping the recent advances in mind, a small set of experiments are also performed with the Prompt-based learning approach. The details of these approaches are provided in this section.

A. Neural methods

The overall framework for neural methods consists of three modules (a) *Embedding Generation*, (b) *Sequence to Vector* and (c) *Dense Layer*, as shown in Figure 3. For the embedding generation module, this study has experimented with both static and contextual embeddings. Pre-trained word [1] embeddings are used as static embedding, and BERT embeddings [22] are used as contextual embedding.

In sequence to vector module, LSTM and BiLSTM are used. Additionally, three variations of BiLSTM have been used, (a) *BiLSTM with Document Context*, (b) *BiLSTM with Sentence CNN* and (c) *BiLSTM with Subword CNN*. All the variations of BiLSTM are trained separately using Fasttext and BERT embeddings. Finally, the dense layer learns the function to predict the event status for each token of the input sentence.

1) LSTM: A 100-dimensional, single-direction LSTM model is used as a baseline.

2) BiLSTM: The accurate detection of a token as a label depends on the backward and forward context information. BiLSTM serves as a great baseline to model both backward and forward contexts. Further modification of the BiLSTM is done as described below to capture more information.

3) BiLSTM with Subword CNN: Subword character CNN is used as an intermediate layer between the embedding layer and BiLSTM. Subword character CNN captures meaningful representations of out-of-vocabulary words for learned embeddings. Each word embedding vector is represented as the result of a CNN having 100 filters, with max pooling resulting in a 100-dimensional character representation of a token. The embedding of the token at that place is subsequently updated to include character representation, and the updated representation goes as an input to the subsequent LSTM layer.

4) BiLSTM with Sentence CNN: Several works have used a CNN on the sentence level [23] for event detection. In a sentence $w = \{w_1, \ldots, w_n\}$ of n tokens, while determining the status of an event of a word at position i, the CNN (1-dimensional CNN) convolves over the sequence w with positional encodings $p = \{p_1, \ldots, p_n\}$. The positional encoding encodes the distance between the target token i and each token position $j \in [1, n]$. In this work, we have adopted the work

[1] https://fasttext.cc/docs/en/english-vectors.html

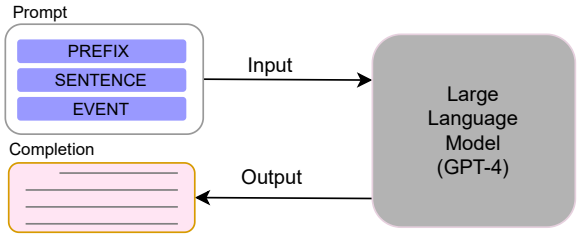

Fig. 4: The input to an LLM is a sequence of tokens sent through prompts. The output is the model's prediction based on the query.

of Nguyen et al. [23], where the output of the CNN is then sent to a max-pooling layer to build a representation c_i for the target location i that is appended to the output of the BiLSTM o_i at the time step before generating the prediction.

The CNN consists of 200 filters (Each word's bigrams and trigrams were scoped 100 times). Using signed bucketing $(\pm 1, \pm 2, \ldots, \pm 20, > 20)$, the positional arguments are encoded between the target token at position i and the token at position j. With its 5-dimensional embedding, each bucket refers to a unique choice of position.

5) BiLSTM with Document Context: The main problem with BiLSTM is that the only information available is from the set, which may not always be enough to say precisely whether a given token represents an event. Therefore, to boost the performance of BiLSTM, the document context was integrated with the sentence context already captured by the model. Based on previous research involving the global context, conclusions are drawn that the exact prediction of composite realis events requires a lot of document context across pages or documents [5]. Therefore, the entire document has been considered as a single sentence in this experiment.

B. Prompt-based methods

Large Language Models (LLMs) are, at their core, massive transformer-based models that employ deep learning techniques to process and understand natural language. By training on vast amounts of text data, these models can uncover patterns and connections between entities within the language [4].

They possess the capacity to comprehend intricate textual content, identify objects and their interrelationships, and generate grammatically accurate and coherent language. The LLM utilized for our purpose is GPT-4[2]. GPT-4 has broader general knowledge and problem-solving skills, enabling it to resolve difficult problems more accurately [24].

Even though LLMs perform very well in most tasks, we need to fine-tune them for the task we want to accomplish using them [25]. Fine-tuning an LLM requires a lot of resources that might not be readily available to everyone. Hence, we employed a different approach to model the LLM. This type of learning involving prompts to direct the decision of

[2]https://openai.com/product/gpt-4

```
[prefix]
Sentence: [sentence]
Event:
```

(a) zero-shot

```
[prefix]
Sentence: [sentence]
Event: [event]
Sentence: [sentence]
Event:
```

(b) few-shot

Fig. 5: The prompt format is shown, where the [] contains variables that are sent to the LLM. In prefix, the definition of realis event is provided. Sentences and event pairs were taken from a JSON file, which is given in Figure 6.

TABLE II: Description of hyperparameters of baseline models

Hyperparameter	Values
Epoch	1000
Batch size	32
Learning rate	2e-5
LSTM size	100
BiLSTM size	100
Word embedding dimension	100
BERT embedding dimension	3072

the LLMs according to users' specifications is also known as prompt-based learning [26]. Prompt-based learning removes the necessity of labeled data and computing resources for fine-tuning [27]. Figure 4 illustrates a basic setting for prompt-based learning.

We used the zero-shot and few-shot prompt-based approach. The zero-shot approach was adopted to see if GPT-4 can recognize realis events by providing the definition of realis event. In few-shot learning, we offered feedback to the GPT-4 if it made errors in detecting realis events.

V. Experiments

Basic Setup: The manually annotated dataset, "*Gatha-200*" has been divided into three sets. The training and development set consists of 120 and 20 stories, respectively. For testing, we have used 60 stories. We performed two sets of experiments to evaluate the performance of various baseline models on the proposed dataset, "*Gatha-200*". In the first set of experiments, we used Fasttext word embeddings. The second set of experiments is conducted to check the effect of different word embedding. Here, we kept the model as it is but changed the word embedding from Fasttext to BERT. Table II lists all the hyperparameters used across the models.

Prompt-based Methods: To achieve the task of event detection using prompt engineering, we embarked upon a series of well-considered steps, which are as follows:

1) (Prompting) Generating prompts: Prompts are paramount when communicating with and guiding the behavior of LLM. They function as inputs or questions that users can present to elicit particular responses from the model. Attaining desired results with LLM models heavily relies on well-crafted prompt design. This process is commonly known as prompt engineering.

TABLE III: Model performance of event detection. BiLSTM with document context with BERT embeddings is the best-performing model. The reported bootstrap confidence interval of all the metrics is 95%.

METHOD	WORD EMBEDDINGS			BERT EMBEDDINGS		
	PRECISION	RECALL	F1 (macro)	PRECISION	RECALL	F1 (macro)
Verbs Only	33.8[32.4-35.1]	90.2[88.9-91.5]	49.1[47.6-50.6]	-	-	-
LSTM	88.8[87.6-89.9]	83.6[82.4-84.8]	86.1[85.2-87.0]	-	-	-
BiLSTM	87.9[86.7-89.0]	85.9[84.8-87.8]	86.9[86.0-87.8]	86.8[85.7-87.9]	94.9[94.2-95.7]	90.7[89.9-91.4]
+Sentence CNN	89.9[88.8-91.0]	81.3[80.0-82.6]	85.4[84.4-86.3]	87.0[85.9-88.1]	94.5[93.8-95.3]	90.6[89.9-91.4]
+Subword CNN	87.6[86.5-88.8]	88.7[87.6-89.7]	88.2[87.3-89.0]	87.8[86.7-88.9]	95.1[94.3-95.8]	91.3[90.6-92.0]
+Document Context	87.4[86.2-88.6]	86.6[85.4-87.7]	87.0[86.1-87.8]	**89.3**[88.2-90.2]	**95.3**[94.5-96.0]	**92.2**[91.5-92.8]

{"sentence" : "One day a crocodile named Karalamukha came out of the waters and loitering on the sands came to the tree ","event" : "came, loitering, came"}

{"sentence" : "Mandy laughed and said , " I can not believe you ","event" : "laughed, said"}

.....

Fig. 6: In few-shot learning, events and their corresponding sentences are sent as a prompt to the GPT-4 as input as a context.

2) Zero-Shot Setting: Zero-shot learning involves zero training or fine-tuning the LLM to perform tasks where examples are not shown. We use the prompts to generate weakly labeled data. Referring to Figure 5a, we can see that we did not provide any example sentences in the prompt. We just provided the definition as prefixes to generate weak labels.
3) Few-Shot Setting: Few-shot learning with LLMs is an extension of zero-shot learning, where the model is trained or fine-tuned to perform tasks with very limited examples or shots. If the weakly labeled data is incorrect, we correct it and give it as an example/feedback for the next iteration, as shown in Figure 5b. We also showed examples of corrected labels in Figure 6. The corrected examples are saved in a JSON file from which we fill in the variables present in the form. We iterate the above steps till the model performs substantially well, which in our case was 30 labeled sentences.

VI. Results

The performance of various models using different word embeddings are shown in Table III. As expected, use of BERT embeddings led to improved performance for the same model while using static embeddings of Fasttext. Furthermore, we observe that the BiLSTM model is performing better in comparison with verbs only and LSTM baselines. This is due to the ability of BiLSTM to capture both forward and backward contexts. The incorporation of Subword CNN with BiLSTM further improves the performance, resulting in an F1 score of 88.2%. This is attributed to the average of sub-tokens generated by the subword CNN. However, when the word embeddings were changed to BERT, BiLSTM+Document Context performed best. This is due to the capture of better context by treating the whole document as a sentence. Moreover, the rich embedding of BERT elevates the performance of the model, achieving the F1 score of 92.2%.

In prompt-based learning methods, we predicted events for 20 stories from our dataset in a zero-shot mode and a few-shot mode. A shot means feedback for wrong predictions of GPT-4. In the zero-shot settings, a sentence and prompt defining realis events are given as input to the GPT-4 model to predict events. The zero-shot mode yields a low F1 score of 39.4% as shown in Table IV.

TABLE IV: Performance of zero-shot and few-shot learning.

LEARNING	N-SHOT	F1
ZERO-SHOT	-	39.4
FEW-SHOT	5-shot	58.9
	10-shot	72.1
	20-shot	79.8
	30-shot	84.5
	40-shot	84.7

Conversely, in few-shot mode, we give random sentences as input for prompting, correcting them, and providing that as feedback before further generation. We experimented with 5-shot, 10-shot, 20-shot, 30-shot, and 40-shot. Table IV indicates that as the number of shots increases, performance also improves, achieving 84.5% F1 score for 30-shot. However, further increments of shots improved marginal performance, indicating that 30-shot is the optimal value. The results indicate that GPT-4 possesses extensive potential for performing a wide range of NLP tasks, including event detection.

VII. Error Analysis

This section describes the errors while predicting realis events by the best-performing BiLSTM+Document context model and prompt-based few-shot method. We observed that most errors in both approaches stemmed from incorrectly classifying unreal events as real ones. In the sentence, "*He dreamt that he was flying in the air.*" the word *"flying"* was predicted as a realis event by the BiLSTM+Document context model. However, "*flying*" is happening in someones dream, which is an unreal event. Similarly, in the sentence, "*While passing by, the two bulls Cleo and Gabby would stop seeing the garbage.*" the word "*seeing*" was classified as realis event. However, it is clear that the sentence is a generic statement and cannot be treated as a realis event according to the aspects we mentioned in Figure 2. Furthermore, in the zero-shot setting of the prompt-based learning method, the GPT-4 was even

predicting proper nouns, such as the names of persons or places as events. However, after providing 30 examples of realis events as feedback (30 shots), most of the error was found to be coming from predicting the events that might happen in the future. For instance, in the sentence, "*You will be surprised to know that Hindi is now being taught in schools.*" the word "*surprised*" was wrongly predicted as a realis event. Apart from the error, it is noteworthy that the prompt-based approach has demonstrated its ability to successfully detect events even within complex sentences. For instance, in the sentence, "*Sunny was surprised to see the police at his farm.*". The GPT-4 has predicted only the word "*surprised*" as the event while neglecting the word "*see*". This was in accordance with the annotation guideline.

VIII. Conclusion

In this work, we have introduced "*Gatha-200*", a dataset for event detection in the short stories. The short stories are in the Indian context and mainly focus on children. We cleaned, pre-processed, and annotated the stories, adhering to the annotation guidelines. We evaluated baseline neural models for event detection. Results indicate that BiLSTM+Document context with BERT embeddings performs best on the proposed dataset. Furthermore, we have showcased the feasibility of employing LLM (GPT-4) for our task, yielding promising results. In the future, we intend to focus on event classification and the extraction of event arguments using GPT-4.

References

[1] N. Peinelt, D. Nguyen, and M. Liakata, "tbert: Topic models and bert joining forces for semantic similarity detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7047–7055, 2020.

[2] M. Sims, J. H. Park, and D. Bamman, "Literary event detection," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 3623–3634, Association for Computational Linguistics, July 2019.

[3] R. Sprugnoli and S. Tonelli, "One, no one and one hundred thousand events: Defining and processing events in an inter-disciplinary perspective," *Natural language engineering*, vol. 23, no. 4, pp. 485–506, 2017.

[4] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[5] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 789–797, 2010.

[6] L. D. Consortium *et al.*, "Ace (automatic content extraction) english annotation guidelines for events. version 5.4.3," *ACE*, 2005.

[7] J. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of bionlp shared task 2011. in proceedings of the bionlp shared task 2011 workshop," *Association for Computational Linguistics, Portland, OR*, pp. 1–6, 2011.

[8] A. Louis and A. Nenkova, "A corpus of general and specific sentences from news.," in *LREC*, vol. 1818, p. 10, Citeseer, 2012.

[9] E. Pavlick, H. Ji, X. Pan, and C. Callison-Burch, "The gun violence database: A new task and data set for nlp," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1018–1024, 2016.

[10] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173111–173137, 2019.

[11] J. Araki and T. Mitamura, "Joint event trigger identification and event coreference resolution with structured perceptron," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2074–2080, 2015.

[12] H. Lin, Y. Lu, X. Han, and L. Sun, "Nugget proposal networks for Chinese event detection," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 1565–1574, Association for Computational Linguistics, July 2018.

[13] R. V S S Patchigolla, S. Sahu, and A. Anand, "Biomedical event trigger identification using bidirectional recurrent neural network based models," in *BioNLP 2017*, (Vancouver, Canada,), pp. 316–321, Association for Computational Linguistics, Aug. 2017.

[14] X. Feng, B. Qin, and T. Liu, "A language-independent neural network for event detection," *Science China Information Sciences*, vol. 61, pp. 1–12, 2018.

[15] W. Yu, M. Yi, X. Huang, X. Yi, and Q. Yuan, "Make it directly: event extraction based on tree-lstm and bi-gru," *IEEE Access*, vol. 8, pp. 14344–14354, 2020.

[16] T. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[17] X. Wang, X. Han, Z. Liu, M. Sun, and P. Li, "Adversarial training for weakly supervised event detection," 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)*, pp. 998–1008, 2019.

[18] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[19] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, (Online), pp. 255–269, Association for Computational Linguistics, Apr. 2021.

[20] J. Aguilar, C. Beller, P. McNamee, B. Van Durme, S. Strassel, Z. Song, and J. Ellis, "A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards," in *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, (Baltimore, Maryland, USA), pp. 45–53, Association for Computational Linguistics, June 2014.

[21] M. Lee, L.-K. Soon, E. G. Siew, and L. F. Sugianto, "CrudeOilNews: An annotated crude oil news corpus for event extraction," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, (Marseille, France), pp. 465–479, European Language Resources Association, June 2022.

[22] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013.

[23] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 365–371, 2015.

[24] A. Bahrini, M. Khamoshifar, H. Abbasimehr, R. J. Riggs, M. Esmaeili, R. M. Majdabadkohne, and M. Pasehvar, "Chatgpt: Applications, opportunities, and threats," in *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pp. 274–279, IEEE, 2023.

[25] R. Behnia, M. R. Ebrahimi, J. Pacheco, and B. Padmanabhan, "Ew-tune: A framework for privately fine-tuning large language models with differential privacy," in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 560–566, IEEE, 2022.

[26] H. Lang, M. N. Agrawal, Y. Kim, and D. Sontag, "Co-training improves prompt-based learning for large language models," in *International Conference on Machine Learning*, pp. 11985–12003, PMLR, 2022.

[27] I. Beltagy, A. Cohan, R. Logan IV, S. Min, and S. Singh, "Zero-and few-shot nlp with pretrained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 32–37, 2022.