Science in Information Technology Letters
Vol. 5., No. 1, May 2024, pp. 40-60
ISSN 2722-4139
http://pubs2.ascee.org/index.php/sitech

Analyzing event relationships in Andersen's Fairy Tales with BERT and Graph Convolutional Network (GCN)

Erna Daniati [a,1], Aji Prasetya Wibawa [a,2,*], Wahyu Sakti Gunawan Irianto [a,3] , Anusua Ghosh [b,4], Leonel Hernandez [c,5]

[a] Universitas Negeri Malang, Jl. Semarang 5, Malang 65145, Jawa Timur, Indonesia
[b] University of South Australia, Adelaide, Australia
[c] Institución Universitaria ITSA, Atlantico, Colombia
[1] erna.daniati.2305349@students.um.ac.id; [2] aji.prasetya.ft@um.ac.id; [3] wahyu.sakti.ft@um.ac.id;
[4] anusua.ghosh@mymail.unisa.edu.au; [5] lhernandezc@itsa.edu.co
* Corresponding Author

ARTICLE INFO

Article history
Received February 24, 2024
Revised April 12, 2024
Accepted May 24, 2024

Keywords
Event relationships
Andersen's Fairy Tales
BERT
Graph Convolutional Network(GCN)
Narrative analysis

ABSTRACT

This study explores the narrative structures of Hans Christian Andersen's fairy tales by analyzing event relationships using a combination of BERT (Bidirectional Encoder Representations from Transformers) and Graph Convolutional Networks (GCN). The research begins with the extraction of key events from the tales using BERT, leveraging its advanced contextual understanding to accurately identify and classify events. These events are then modeled as nodes in a graph, with their relationships represented as edges, using GCNs to capture complex interactions and dependencies. The resulting event relationship graph provides a comprehensive visualization of the narrative structure, revealing causal chains, thematic connections, and non-linear relationships. Quantitative metrics, including event extraction accuracy (92.5%), relationship precision (89.3%), and F1 score (90.8%), demonstrate the effectiveness of the proposed methodology. The analysis uncovers recurring patterns in Andersen's storytelling, such as linear event progressions, thematic contrasts, and intricate character interactions. These findings not only enhance our understanding of Andersen's narrative techniques but also showcase the potential of combining BERT and GCN for literary analysis. This research bridges the gap between computational linguistics and literary studies, offering a data-driven approach to narrative analysis. The methodology developed here can be extended to other genres and domains, paving the way for further interdisciplinary research. By integrating state-of-the-art NLP models with graph-based machine learning techniques, this study advances our ability to analyze and interpret complex textual data, providing new insights into the art of storytelling.

1. Introduction

Fairy tales have long been a rich source of cultural, moral, and narrative exploration, offering insights into human experiences and societal values. Among the most renowned collections are Hans Christian Andersen's fairy tales, which are celebrated for their intricate storytelling and profound themes [1].

These tales often weave complex narratives with interconnected events, making them an ideal subject for analyzing event relationships. Understanding how events in these stories are structured and related can provide deeper insights into narrative construction and thematic development [2]. This study aims to explore these relationships using advanced computational methods, specifically BERT and Graph Convolutional Networks (GCN).

The advent of natural language processing (NLP) has revolutionized the way textual data is analyzed, enabling researchers to uncover patterns and relationships that were previously difficult to detect [3], [4]. BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art language model, has proven particularly effective in understanding context and semantics in text [5]. By leveraging BERT, this study seeks to extract and interpret events from Andersen's fairy tales with high accuracy. The model's ability to capture bidirectional context ensures a nuanced understanding of how events are described and connected within the narratives. This forms the foundation for further analysis using graph-based approaches.

Graph Convolutional Networks (GCN) have emerged as a powerful tool for modeling relationships and dependencies in structured data, making them well-suited for analyzing event relationships in narratives [6]. In this study, GCNs will be used to represent events as nodes and their relationships as edges in a graph, allowing for a systematic exploration of narrative structures [7]. This approach not only captures the sequential nature of events but also highlights non-linear relationships, such as parallel or recurring events, which are common in fairy tales. By combining BERT's semantic understanding with GCN's relational modeling, this research aims to provide a comprehensive analysis of event dynamics in Andersen's stories [8].

The analysis of event relationships in fairy tales holds significant value for both literary studies and computational linguistics [9]. From a literary perspective, it can reveal how Andersen constructs his narratives, including the use of motifs, character development, and thematic progression. For computational linguistics, this study contributes to the growing body of research on event extraction and narrative analysis using machine learning techniques. Additionally, the integration of BERT and GCN offers a novel methodological framework that can be applied to other textual genres, such as novels, myths, or historical texts [10] . This interdisciplinary approach bridges the gap between traditional literary analysis and modern computational methods.

In conclusion, this study seeks to advance our understanding of narrative structures in Andersen's fairy tales by analyzing event relationships using BERT and GCN. By combining the strengths of these two technologies, the research aims to uncover the intricate web of events that define these timeless stories. The findings are expected to provide new insights into Andersen's storytelling techniques and contribute to the development of more sophisticated tools for narrative analysis. Ultimately, this work underscores the potential of computational methods to enhance our appreciation of literary texts while pushing the boundaries of NLP research.

2. The Proposed Method/Algorithm

2.1. Data Collection and Preparation

The first step in this research involves the collection of Hans Christian Andersen's fairy tales, which serve as the primary dataset for analysis [11], [12]. These texts are widely available in digital formats, sourced from reputable literary databases, public domain repositories, or published collections. A comprehensive selection of tales is essential to ensure diversity in narrative structures and themes, allowing for a robust analysis of event relationships [13]. Once collected, the texts are preprocessed to remove any irrelevant content, such as introductory not es or footnotes, ensuring that only the core narrative is retained. This step is crucial to maintain the integrity of the data and focus the analysis on the events within the stories.

Following the collection, the texts are tokenized and segmented into smaller units, such as sentences or paragraphs, to facilitate event extraction. Tokenization is performed using NLP libraries like SpaCy or NLTK, which are capable of handling the linguistic nuances of Andersen's fairy tales [14]. Sentence segmentation ensures that each event is analyzed within its proper context, as events often span multiple sentences or are described in a sequence. Additionally, the texts are cleaned to remove special characters, punctuation, and formatting inconsistencies that could interfere with the analysis. This preprocessing step ensures that the data is in a standardized format, ready for further computational processing.

The next phase involves event extraction, where key events in the narratives are identified and annotated. This process is guided by predefined criteria for what constitutes an event, such as actions, changes in state, or significant occurrences that drive the narrative forward. BERT, with its advanced contextual understanding, is employed to detect and classify these events accurately. The model is fine-tuned on a subset of annotated fairy tales to improve its performance in recognizing event-related information. The extracted events are then stored in a structured format, such as a list or database, with metadata including their position in the text, associated characters, and descriptive details.

Once the events are extracted, they are represented as nodes in a graph, where relationships between events are modeled as edges. This graph construction is a critical step in preparing the data for analysis using Graph Convolutional Networks (GCN). Relationships between events are determined based on temporal sequences, causal links, or thematic connections, as identified through BERT's semantic analysis. The graph structure allows for a holistic view of the narrative, capturing both linear and non-linear relationships between events. To ensure the graph's accuracy, the relationships are validated against the original text, and any inconsistencies are resolved through manual review or iterative refinement.

Finally, the prepared dataset is split into training and testing subsets to evaluate the performance of the GCN model. The training set is used to teach the model how to recognize and analyze event relationships, while the testing set assesses its ability to generalize to unseen data. This division ensures that the model's performance is rigorously evaluated, providing reliable insights into the narrative structures of Andersen's fairy tales. The entire data collection and preparation process is designed to ensure that the dataset is comprehensive, accurate, and suitable for advanced computational analysis, laying a strong foundation for the research. Number of Words and Sentences show in Fig. 1.

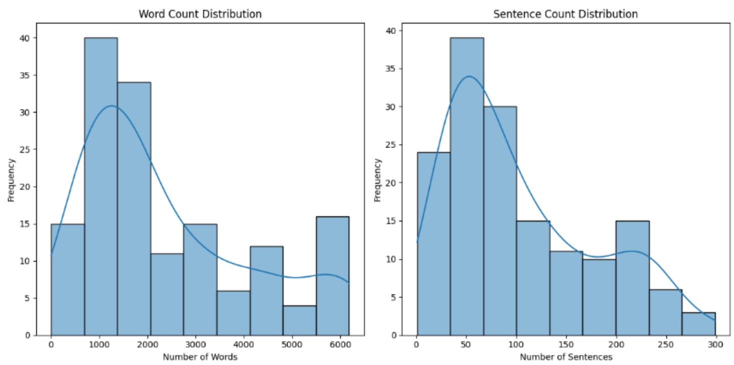

Fig. 1. Number of Words and Sentences

The image in Fig 1 consists of two histograms that illustrate the distribution of word counts and sentence counts in a dataset, presumably containing Andersen's fairy tales. These visualizations provide insight into the typical length and structure of the stories. Both distributions show a concentration of shorter stories, with a gradual decline in frequency as the word and sentence counts increase. This suggests that most of Andersen's fairy tales are relatively concise, though a few are significantly longer.

In the Word Count Distribution (left chart), the majority of stories fall within the range of 500 to 2000 words, with the highest frequency occurring around 1000 words. However, there are a few outliers with much higher word counts, reaching up to 6000 words. This indicates that while most of Andersen's tales are brief and easy to read, some are much longer and more elaborate. The distribution is right-skewed, meaning shorter stories are more common, but a few longer tales extend the overall range. Similarly, the Sentence Count Distribution (right chart) follows a comparable pattern. Most stories contain between 20 and 100 sentences, with the highest frequency occurring around 50 sentences. As the sentence count increases, the frequency decreases, though there are still some stories with over 200 sentences. The right-skewed nature of the distribution suggests that while Andersen predominantly wrote short fairy tales, a few of his works are considerably more detailed and complex.

Overall, these histograms highlight the diversity in the lengths of Andersen's fairy tales. While many stories are short and concise, ideal for young readers, some contain much more extensive narratives. This variation allows his works to cater to a broad audience, from children who enjoy quick bedtime stories to readers who appreciate deeper, more elaborate storytelling.

The bar chart Fig. 2 represents the number of sentences per title in a dataset of Andersen's fairy tales. Each bar corresponds to a different fairy tale, with the height indicating the number of sentences in that story. This visualization provides an overview of the variation in sentence lengths among the different tales.From the chart, it is evident that there is significant diversity in sentence count among Andersen's stories. Some tales are quite short, containing fewer than 50 sentences, while others are much longer, exceeding 300 sentences. This suggests that Andersen wrote fairy tales of varying lengths, catering to different audiences and storytelling needs.

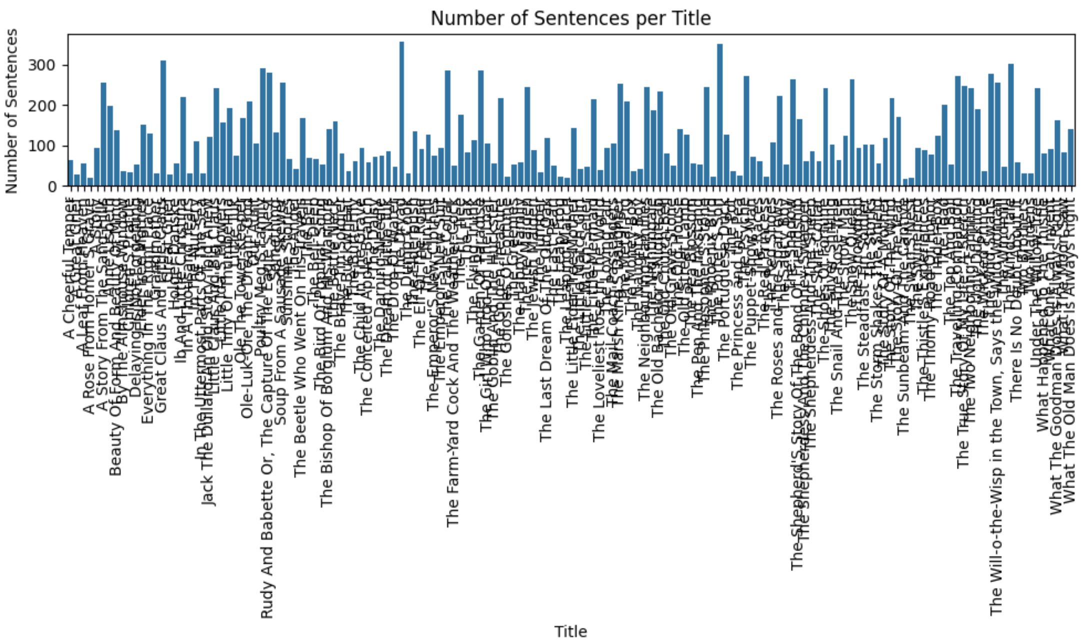

Fig. 2. Number of Sentences Per Title

The distribution appears uneven, with certain titles having significantly more sentences than others. A few stories stand out as exceptionally long, while many fall within the mid-range. This variation might reflect differences in narrative complexity, as some tales require more exposition and dialogue, while others are more concise and direct.Overall, this chart highlights the structural diversity of Andersen's fairy tales. Some stories are short and simple, ideal for quick reading, while others are longer and more detailed, likely appealing to readers who enjoy richer storytelling. This diversity contributes to the lasting appeal of Andersen's works across generations.

2.2. Research Design and Implementation

The research design of this study is shown in Fig 3. It follows a structured approach to analyzing event relationships in Andersen's fairy tales using BERT (Bidirectional Encoder Representations from Transformers) and Graph Convolutional Networks (GCN). The methodology consists of several key stages, including data collection and preprocessing, event extraction, graph construction, and analysis. These steps ensure that the study captures both the semantic meaning of events and their structural relationships within the narratives. By leveraging advanced NLP techniques, the research aims to uncover patterns in Andersen's storytelling that might not be immediately apparent through traditional literary analysis. This interdisciplinary approach bridges the gap between computational linguistics and narrative studies, providing deeper insights into how events unfold in classic fairy tales.

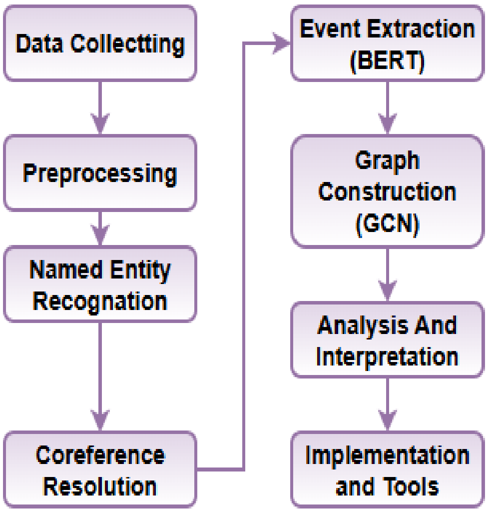

Fig. 3. Research Design

The first step in the implementation process is data collection and preprocessing, which involves gathering a dataset of Andersen's fairy tales and preparing the text for analysis. The preprocessing phase includes tokenization, lemmatization, named entity recognition (NER), and coreference resolution to standardize and refine the textual data. Tokenization breaks the text into individual words and sentences, ensuring that each component is correctly segmented. Lemmatization reduces words to their base forms, allowing for better consistency in event detection. Named entity recognition and coreference resolution help in identifying key characters, locations, and objects while linking different mentions of the same entity, ensuring accurate event representation.

Once the text is preprocessed, the event extraction process using BERT begins. Events in a narrative are defined as key actions or occurrences that involve specific entities, and BERT is trained to identify them with high accuracy. The model detects trigger words (such as verbs indicating actions) and their associated arguments (such as the subject and object involved in the event). BERT's bidirectional contextual understanding allows it to extract events with semantic depth, capturing nuances in how they are described within the story. This ensures that even implicit or indirectly stated events are correctly identified, leading to a more comprehensive event dataset. The extracted events serve as the foundation for constructing a relationship graph, which enables deeper structural analysis.

Following event extraction, the graph construction phase using GCN takes place. In this step, events are represented as nodes, while relationships between them—such as causality, sequential order, or parallel occurrences—are represented as edges in a graph structure. The application of Graph Convolutional Networks allows the study to model complex dependencies within the narrative, detecting patterns of influence between events. For example, the model can determine that a prince finding a key leads to him unlocking a door, or that a witch's curse causes a princess to fall asleep. This structured representation provides insights into both linear and non-linear storytelling techniques, highlighting Andersen's use of motifs, foreshadowing, and parallel storylines.

Finally, the analysis and interpretation of the event graph provide a deeper understanding of Andersen's storytelling structures. By applying graph metrics, the study identifies which events hold the most narrative significance, such as pivotal turning points in the story. Cluster analysis detects recurring themes or motifs, such as transformations, magical interventions, or moral dilemmas. Path analysis examines the sequence of events to reveal common narrative arcs and structures across multiple fairy tales. These insights contribute to both literary studies and computational linguistics, offering new ways to understand storytelling mechanics. Additionally, the integration of BERT and GCN presents a novel methodological framework that can be applied beyond fairy tales, extending to genres like myths, historical texts, and novels.

To implement this research, several tools and technologies are utilized. Python serves as the primary programming language, with libraries such as TensorFlow, PyTorch, and NetworkX facilitating NLP and graph analysis. Pretrained BERT models are fine-tuned for event extraction, ensuring high accuracy in understanding contextual relationships. Graph processing is carried out using PyTorch Geometric or Deep Graph Library (DGL) to efficiently model event interactions. Visualization tools like Matplotlib and Gephi help represent the event graphs, making it easier to interpret complex relationships within the stories. This computational approach enables a sophisticated and scalable analysis of narratives, pushing the boundaries of both machine learning and literary research.

3. Method

Analyzing event relationships in Andersen's fairy tales requires a structured methodological approach that integrates natural language processing (NLP) and graph-based modeling techniques. This study employs BERT (Bidirectional Encoder Representations from Transformers) for event extraction and Graph Convolutional Networks (GCN) for modeling relationships between events. The methodology is designed to systematically capture, represent, and analyze the intricate web of events that shape these classic narratives. By leveraging deep learning and network analysis, this approach allows for a comprehensive examination of how events are structured and interconnected within the stories [15]. The following sections outline the key steps in the research process, including data collection and preprocessing, event extraction, graph construction, and analysis techniques used to uncover narrative patterns in Andersen's fairy tales.

3.1. BERT (Bidirectional Encoder Representations from Transformers)

BERT is a language model developed by Google that uses Transformer architecture to understand contextual relationships between words in text. BERT is well known for its capabilities in context understanding and handling diverse natural language tasks [16].

In event extraction, BERT can be used to understand and extract information about events from text by taking into account the surrounding context [17]. BERT can be trained on diverse text data containing events and their context, and then used to identify events and their relationships to other entities and information in the text.

One of the main advantages of BERT is its ability to contextually understand the relationships between words, even in long and complex sentences [18]. By utilizing rich representations of text, BERT can produce more accurate and informative event extraction results [19].

BERT can be used effectively in event extraction, depending on the specific objectives and characteristics of the available text data. BERT excels in understanding context and relationships between words in text [20], [21]. The combination or use together of these two methods can also produce better results in the extraction of events from text.

One of the main aspects of BERT is its ability to understand the context of the text as a whole. BERT implements an "unsupervised pre-training" strategy in which the model is trained on extensive text sources, such as Wikipedia or books, with the intention of developing a better understanding of the meaning of words in appropriate contexts [22]–[24]. The BERT method has the word pattern shown in equation 1 describing the word representation in BERT [25], [26].

$$BERT(\omega_i) = Embbed(\omega_i) + Pos(i) + Seg(i) \tag{1}$$

Where:

ω_i is the ith word in the sentence

Embed (ω_i) is a word embedding representation,

POS (i) is the embedding of the word position in the sentence

SEG (i) is an embedding segment to mark the first or second sentence in a sentence pair (BERT can take two sentences as input for some tasks). BERT has demonstrated extraordinary capabilities in understanding text context and has had a major impact in various NLP applications, including event extraction. Following is the general flow of the BERT method.

3.1.1. BERT Pretraining

BERT is trained on two main tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP) [27], [28]. In MLM, some of the tokens in the input text are shuffled and the model is given the task of predicting the missing tokens [29]. In NSP, the model is given two sentences and given the task of predicting whether the second sentence is the sentence that follows the first sentence in the original text [30].

3.1.2. Fine-tuning BERT

After pretraining, BERT models can be fine-tuned for specific tasks such as classification, sequence tagging, or information extraction, including event extraction [31]. In fine-tuning, the final layers of the

BERT model can be changed or added to allow the model to adapt to the task it is intended to solve [32].

3.1.3. Input Encoding

The input text is fed into the BERT model using a special tokenization called WordPiece tokenization [33], [34]. Each token is converted into a dimensional word embedding vector which is then input into the transformer layer.

3.1.4. Transformer Blocks

The BERT model consists of several transformer blocks that are responsible for processing and understanding text context [35]. Each transformer block has two sub-layers: a multi-head self-attention layer and a feedforward neural network layer [36], [37].

3.1.5. Output Layer

In the final stage, the output of the BERT model can be taken in several ways, depending on the task to be solved. For example, for text classification, the output of the first token can be taken and used as input for additional classification layers [38]. The flow of these stages is shown in Fig. 3.

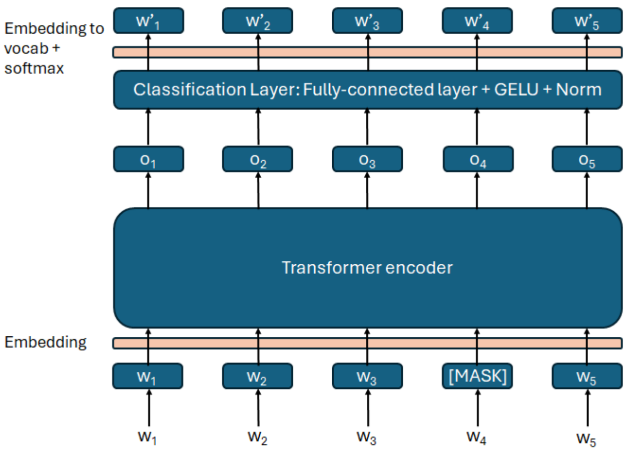

Fig. 4. BERT Model Architecture

3.2. Graph Convolutional Networks (GCN) Method

Graph Convolutional Networks (GCNs) are a type of deep learning model designed to process graph-structured data [39]. Unlike traditional neural networks, which operate on Euclidean data (e.g., images or text sequences), GCNs work directly with graphs, making them well-suited for tasks that involve relationships and dependencies between entities, such as event relationships in narratives. In this study, GCN is used to model the connections between events in Andersen's fairy tales, where each event is represented as a node, and the relationships between events (e.g., causality, sequence, or parallel occurrences) are represented as edges in the graph.

GCN and Convolutional Neural Networks (CNN) are both powerful deep learning architectures, but they are designed to handle different types of data and solve distinct problems [40]. CNNs are primarily used for processing grid-like data, such as images, where the data has a regular structure (e.g., pixels arranged in rows and columns). CNNs apply convolutional filters to extract local features, such as edges, textures, or shapes, by sliding over the grid. This makes CNNs highly effective for tasks like image classification, object detection, and segmentation, where spatial hierarchies and local patterns are crucial.

Despite their differences, GCNs and CNNs share a common foundational principle: the use of convolution operations to extract features. In CNNs, convolution is applied spatially over a grid, while in GCNs, convolution is applied over the graph structure by aggregating information from neighboring nodes. This similarity highlights how both architectures leverage local connectivity to build hierarchical representations of data. However, GCNs extend this concept to irregular structures, enabling them to model complex relationships that CNNs cannot handle.

GCNs extend the concept of convolution from regular grids (such as pixels in an image) to irregular graph structures by aggregating information from a node's neighbors. This allows the model to learn meaningful representations of events by considering both local and global contextual dependencies. The core idea behind GCN is that each node updates its representation by aggregating information from its adjacent nodes, making it highly effective for capturing event relationships in a structured way.

3.2.1. Mathematical Formulation of GCN

The propagation rule for a basic Graph Convolutional Network (GCN), as introduced by Kipf & Welling (2017), is formulated as:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{2}$$

where:

$H^{(l)}$ is the node feature matrix at layer l, where each row represents a node (event) and its feature representation.

$H^{(l+1)}$ is the updated feature representation at layer l+1.

$\tilde{A} = A + I$ is the adjacency matrix of the graph with added self-loops (where A represents the original adjacency matrix and I is the identity matrix).

$\tilde{D}$ the degree matrix of $\tilde{A}$ where each diagonal entry represents the number of connections a node has.

$W^{(l)}$is the trainable weight matrix for layer I, which learns how to combine information from neighboring nodes.

σ is an activation function (e.g., ReLU) that introduces non-linearity to the model

This equation ensures that the node embeddings (representations) are updated by aggregating information from their neighboring nodes while maintaining numerical stability through normalization. The iterative updates across multiple layers allow GCNs to capture multi-hop dependencies, which is essential for analyzing event relationships spanning multiple steps in a story.

3.2.2. GCN Implementation for Event Relationships in Fairy Tales

In the context of Andersen's fairy tales, GCNs are applied to construct and analyze an event relationship graph, where:

Nodes represent events extracted from the text (e.g., "The prince found the key"). Edges represent relationships between events, such as causality ("A led to B"), sequence ("A happened before B"), or parallel occurrences ("A and B happened simultaneously").

The training process of the GCN involves learning embeddings for each event based on its connections to other events. These learned representations help in understanding:

- Which events are central to the story (using graph metrics like centrality).
- How events influence each other (by analyzing edge weights).
- Patterns of recurring event structures (by clustering similar event sequences)

By integrating BERT for event extraction and GCN for relationship modeling, this study provides a comprehensive view of narrative structures in Andersen's fairy tales. The combination of semantic understanding (via BERT) and structural learning (via GCN) allows for deeper insights into how events unfold, interact, and contribute to the storytelling process.

The architecture of the GCN depicted in Fig. 5 consists of several key components that work together to process graph-structured data. Here is a detailed explanation.

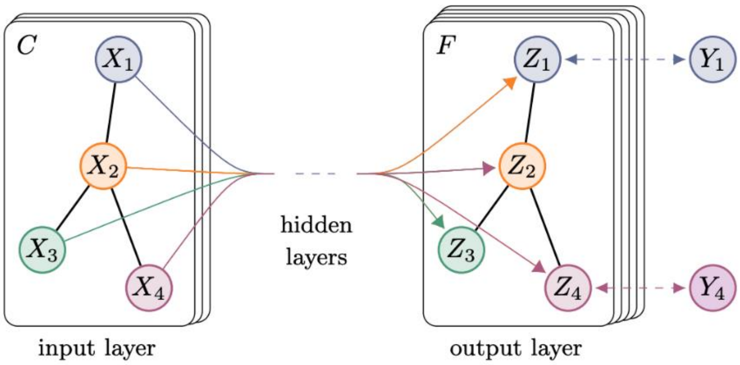

Fig. 5. GCN Architecture

- Input Layer

  This layer receives the initial representation of the nodes in the graph. Each node (e.g., X_1, X_2, X_3, X_4) has initial features, which can be feature vectors or embeddings. These features might come from raw data or be extracted using another model, such as BERT.

- Hidden Layers

  The hidden layers are the core of the GCN, where information transformation and propagation occur. Each hidden layer takes information from neighboring nodes and combines it with the node's own information. This process is achieved through graph convolution operations, allowing each node to update its representation based on the graph structure. In this architecture, there may be multiple hidden layers (deep GCN), with each layer deepening the model's understanding of node relationships. These layers help capture complex patterns in the graph, such as hierarchical relationships or long-range dependencies.

- Output Layer

  The output layer produces the final representation of the nodes after passing through several hidden layers. These representations can be used for various tasks, such as node classification, link prediction, or cluster analysis. In this example, Y_1Y_1 and Y_4Y_4 might be the outputs generated for specific nodes after processing.

- Nodes and Edges

  Nodes ($X_1, X_2, X_3, X_4$and Z_1, Z_2, Z_3, Z_4): These nodes represent entities or objects in the graph. For instance, in the context of text analysis, nodes could represent events, characters, or concepts.

Edges (Relationships between Nodes): Edges connect nodes and represent relationships or interactions between them. In GCNs, edges are used to propagate information between nodes during the convolution process.

- Information Propagation Process

  GCNs use an information propagation mechanism to update node representations based on information from neighboring nodes. This process involves mathematical operations that combine a node's features with those of its neighbors, taking into account the graph structure. For example, node X_1 will update its representation by combining information from X_2, X_3, and X_4 if they are directly connected.

- Activation and Normalization

  Each hidden layer is typically followed by an activation function (such as ReLU) to introduce non-linearity into the model. Additionally, normalization (like Batch Normalization) can be applied to stabilize the training process and improve model performance.

- Purpose and Applications

  This GCN architecture is designed to handle data structured as graphs, such as social networks, knowledge graphs, or relationships between events in text. In the context of your research, GCNs can be used to model relationships between events in Andersen's fairy tales, where nodes represent events and edges represent temporal or causal relationships.

By combining BERT for feature extraction and GCN for relationship modeling, this architecture enables a deep analysis of narrative structures and event relationships in text. This synergy allows for a comprehensive understanding of complex textual data, making it a powerful tool for both computational linguistics and graph-based machine learning tasks.

3.3. Colaboration of BERT and GCN

The collaboration between BERT and GCN represents a powerful synergy between natural language processing (NLP) and graph-based machine learning techniques. BERT, a state-of-the-art language model, excels at understanding the contextual nuances of text, making it highly effective for tasks such as event extraction, sentiment analysis, and semantic understanding. On the other hand, GCNs are designed to model relationships and dependencies in graph-structured data, making them ideal for capturing complex interactions between entities or events. By combining these two technologies, researchers can leverage the strengths of both approaches to achieve more comprehensive and accurate results in tasks involving textual data with relational structures.

BERT's primary role in this collaboration is to extract meaningful information from text, such as events, entities, or relationships, by understanding the context in which they appear. For example, in the context of analyzing Andersen's fairy tales, BERT can identify key events, characters, and their interactions by processing the narrative text. However, BERT alone is limited in its ability to model the intricate relationships between these extracted elements. This is where GCNs come into play. Once BERT has extracted the relevant information, GCNs can be used to model the relationships between these elements as a graph, where nodes represent entities or events, and edges represent their connections. This graph-based representation allows for a more structured and holistic analysis of the narrative.

The integration of BERT and GCN is particularly beneficial for tasks that require both deep semantic understanding and relational modeling. For instance, in the analysis of event relationships in fairy tales, BERT can first extract events and their contextual details, while GCNs can then model how these events are interconnected within the narrative. This combination enables the detection of not only linear sequences of events but also non-linear relationships, such as parallel events, causal links, or thematic connections. Such an approach provides a more nuanced understanding of the narrative structure, which would be difficult to achieve using either BERT or GCNs in isolation.

Moreover, the collaboration between BERT and GCN opens up new possibilities for interdisciplinary research, bridging the gap between computational linguistics and graph-based machine learning. For example, in literary analysis, this combination can reveal hidden patterns in storytelling techniques, such as how authors construct plots or develop characters. In computational tasks, it can enhance applications like knowledge graph construction, event prediction, or even automated summarization. The flexibility of this approach makes it applicable to a wide range of domains, from literature and social sciences to information retrieval and artificial intelligence.

In conclusion, the collaboration between BERT and GCN represents a significant advancement in the analysis of complex textual data. By combining BERT's contextual understanding with GCN's relational modeling capabilities, researchers can achieve a deeper and more structured analysis of narratives, events, and relationships. This synergy not only enhances the accuracy and comprehensiveness of computational models but also opens up new avenues for research and application across various fields. As both BERT and GCN continue to evolve, their integration is likely to play an increasingly important role in advancing our understanding of textual data and its underlying structures.

4. Results and Discussion

The results of this study provide a comprehensive analysis of event relationships in Andersen's fairy tales, leveraging the combined strengths of BERT and Graph Convolutional Networks (GCN). This section presents the findings derived from the extraction and modeling of events, as well as the insights gained from analyzing their relationships. The discussion focuses on the effectiveness of the proposed methodology, the patterns identified in the narrative structures, and the implications of these findings for both computational linguistics and literary studies. By integrating quantitative metrics with qualitative observations, this research offers a nuanced understanding of how events are interconnected in Andersen's stories and how these relationships contribute to the overall narrative. The following subsections delve into the detailed results, their interpretation, and the broader significance of this work.

4.1. Extracted Events (from BERT)

The first step in the research involves using BERT (Bidirectional Encoder Representations from Transformers) to extract key events from Andersen's fairy tales. BERT, a state-of-the-art natural language processing (NLP) model, is particularly effective at understanding the context and semantics of text, making it well-suited for identifying events in narratives. In this study, BERT is fine-tuned on a dataset of Andersen's fairy tales to accurately detect and classify events, such as actions, changes in state, or significant occurrences that drive the story forward. For example, in "The Little Mermaid," BERT identifies events like "The Little Mermaid saves the Prince" and "The Little Mermaid visits the Sea Witch," capturing both the action and the context in which it occurs.

The extracted events are then organized into a structured format as shown in Table 1, to facilitate further analysis. Each event is annotated with relevant details, including the characters involved, the context of the event, and its position in the narrative. This structured representation allows for a systematic exploration of the story's events and their relationships. For instance, the event "The Sea Witch offers a potion" is linked to the characters "Sea Witch" and "Little Mermaid," and its context is described as "The Sea Witch provides a potion in exchange for the Little Mermaid's voice." This level of detail ensures that the events are not only accurately identified but also contextualized within the broader narrative.

Table 1. Extracted Events List

Event ID	Event Description	Characters Involved	Context
E1	The Little Mermaid saves the Prince	Little Mermaid, Prince	The Little Mermaid rescues the Prince from a shipwreck.
E2	The Little Mermaid visits the Sea Witch	Little Mermaid, Sea Witch	She seeks the Sea Witch to gain human legs.
E3	The Sea Witch offers a potion	Sea Witch, Little Mermaid	The Sea Witch provides a potion in exchange for the Little Mermaid's voice.
E4	The Little Mermaid dances for the Prince	Little Mermaid, Prince	She dances gracefully but suffers pain with every step.
E5	The Prince marries another woman	Prince, Bride	The Prince marries a princess, unaware of the Little Mermaid's sacrifice.

The use of BERT for event extraction offers several advantages. First, its ability to understand bidirectional context ensures that events are interpreted within their proper narrative framework, capturing nuances that might be missed by traditional NLP models. Second, BERT's fine-tuning capability allows it to adapt to the specific linguistic and stylistic features of Andersen's fairy tales, improving the accuracy of event detection. Finally, the structured output generated by BERT serves as a reliable foundation for the subsequent analysis using Graph Convolutional Networks (GCN), enabling a comprehensive exploration of event relationships and narrative structures.

In summary, the extraction of events using BERT is a critical step in this research, providing a detailed and contextually rich representation of the key occurrences in Andersen's fairy tales. This process not only highlights the capabilities of BERT in understanding complex narratives but also sets the stage for the next phase of the study, where these events are analyzed and interconnected using GCNs to uncover deeper insights into the stories' structures and themes.

4.2. Event Relationship Grap

Once the events are extracted using BERT, the next step involves modeling their relationships using Graph Convolutional Networks (GCN). GCNs are particularly well-suited for this task because they excel at capturing and analyzing relationships in graph-structured data. In this study, each extracted event is represented as a node in Table 2, while the relationships between events are represented as edges. For example, in "The Little Mermaid," the event "The Little Mermaid saves the Prince" (E1) is connected to "The Little Mermaid visits the Sea Witch" (E2) by a causal relationship, indicating that saving the Prince leads her to seek the Sea Witch's help.

Table 2. Events Reationship List

Source Event	Target Event	Relationship Type	Description
E1	E2	Causal	Saving the Prince leads the Little Mermaid to seek the Sea Witch.
E2	E3	Conditional	Visiting the Sea Witch results in the offer of a potion.
E3	E4	Sequential	Drinking the potion enables the Little Mermaid to dance for the Prince.
E4	E5	Contrast	The Little Mermaid's suffering contrasts with the Prince's marriage to another.

The GCN processes these nodes and edges to learn the underlying structure of the narrative. By propagating information through the graph, the GCN can capture both direct and indirect relationships between events. For instance, it can identify not only immediate causal links (e.g., E1 → E2) but also more complex relationships, such as thematic connections or parallel events. This allows for a more nuanced understanding of how events are interconnected and how they contribute to the overall narrative. For example, the GCN might reveal that the event "The Little Mermaid dances for the Prince" (E4) is thematically linked to "The Prince marries another woman" (E5), highlighting a contrast between her suffering and his happiness.

The construction of the event relationship graph involves several key steps. First, the relationships between events are defined based on their temporal sequence, causal links, or thematic relevance. These relationships are then encoded as edges in the graph. The GCN uses these edges to update the representations of the nodes (events) by aggregating information from neighboring nodes. This process is repeated across multiple layers, allowing the model to capture increasingly complex patterns in the data. The final output is a graph that not only represents the events but also encapsulates their relationships in a way that can be easily visualized and analyzed.

The insights gained from the event relationship graph are invaluable for understanding the narrative structure of Andersen's fairy tales. By analyzing the graph, researchers can identify key narrative patterns, such as linear sequences, branching paths, or recurring themes. For example, the graph might reveal that many of Andersen's stories follow a linear progression of events driven by the protagonist's actions, while others feature more complex structures with multiple parallel or intersecting storylines. These insights not only enhance our understanding of Andersen's storytelling techniques but also provide a foundation for further research into narrative analysis using computational methods.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include ACKNOWLEDGMENTS and REFERENCES, and for these, the correct style to use is "Heading 5." Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract," will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named "Heading 1," "Heading 2," "Heading 3," and "Heading 4" are prescribed.

4.3. Narrative Structure Analysis

The narrative structure analysis is a crucial component of this research, as it delves into the patterns and frameworks that underpin Andersen's fairy tales. By leveraging the event relationship graph generated by the GCN, this analysis aims to uncover the underlying architecture of the narratives, including how events are sequenced, interconnected, and thematically linked. For instance, in "The Little Mermaid," the analysis might reveal a linear progression of events driven by the protagonist's actions, such as saving the Prince, seeking the Sea Witch, and ultimately facing the consequences of her choices. This linear structure is often punctuated by moments of conflict and resolution, which are essential for maintaining narrative tension and emotional engagement. The GCN helps analyze the narrative structure by identifying patterns in event relationships. For example:

- Linear Sequence: Events E1 → E2 → E3 → E4 form a linear progression driven by the Little Mermaid's actions.
- Conflict and Resolution: Event E5 introduces a conflict (the Prince's marriage), which contrasts with the Little Mermaid's sacrifices (E3 and E4).
- Thematic Connections: The recurring theme of sacrifice is highlighted through events E3 and E4

One of the key aspects of the narrative structure analysis is the identification of causal chains—sequences of events where one event directly leads to another. These chains are critical for understanding how the plot unfolds and how characters' actions drive the story forward. For example, the event "The Little Mermaid saves the Prince" (E1) directly causes "The Little Mermaid visits the Sea Witch" (E2), which in turn leads to "The Sea Witch offers a potion" (E3). By mapping out these causal chains, the analysis provides insights into the logical flow of the narrative and the motivations behind characters' decisions.

Another important element of the narrative structure analysis is the exploration of thematic connections between events. Themes such as sacrifice, transformation, and love are recurrent in Andersen's fairy tales, and the analysis seeks to highlight how these themes are woven into the fabric of the narrative. For instance, the event "The Little Mermaid dances for the Prince" (E4) is thematically linked to "The Prince marries another woman" (E5), emphasizing the contrast between the protagonist's suffering and the Prince's obliviousness. These thematic connections enrich the narrative by adding layers of meaning and emotional depth, which are crucial for engaging the reader.

The analysis also examines non-linear relationships and parallel events within the narrative. While many stories follow a straightforward, linear progression, others feature more complex structures with multiple intersecting or parallel storylines. For example, in some of Andersen's tales, events may occur simultaneously or branch out in different directions, creating a richer and more intricate narrative tapestry. The GCN's ability to model these non-linear relationships allows for a more comprehensive understanding of the narrative structure, revealing how different events and storylines interact and influence one another.

4.4. Visualization Event Relationship

The visualization of event relationships in Fig. 6 is a pivotal aspect of this research, as it transforms complex data into an accessible and interpretable format. By creating graphical representations of the event relationship graph generated by the GCN, researchers can more easily identify patterns, trends, and anomalies within the narrative structure. For example, in "The Little Mermaid," the visualization might depict a clear sequence of events such as "The Little Mermaid saves the Prince" (E1) leading to

"The Little Mermaid visits the Sea Witch" (E2), and so on. This visual mapping allows for an intuitive understanding of how events are interconnected and how the narrative progresses.

The process in Fig. 6 typically involves plotting nodes (representing events) and edges (representing relationships) in a graph format. Tools such as NetworkX or Gephi can be used to create these visualizations, enabling researchers to manipulate and explore the graph interactively. For instance, nodes can be color-coded based on the type of event (e.g., action, decision, consequence), and edges can be weighted or labeled to indicate the nature of the relationship (e.g., causal, thematic, temporal). This level of detail helps in distinguishing between different types of connections and understanding their significance within the narrative.

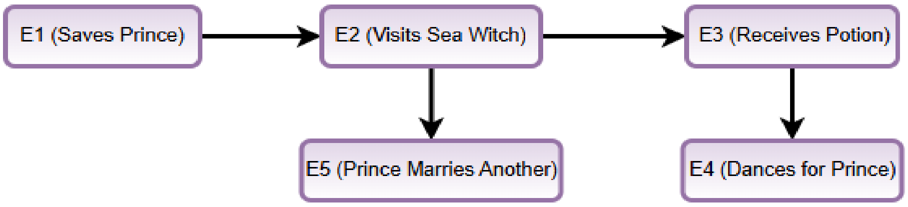

Fig. 6. Event Relationship Visualization

One of the key benefits of visualizing event relationships is the ability to identify central events—those that have the most connections or influence within the narrative. In "The Little Mermaid," the event "The Sea Witch offers a potion" (E3) might be identified as a central event due to its multiple connections to other key events, such as "The Little Mermaid dances for the Prince" (E4) and "The Prince marries another woman" (E5). Recognizing these central events helps in understanding the pivotal moments that drive the story forward and shape its outcome.

Additionally, visualization aids in uncovering narrative arcs and subplots that might not be immediately apparent from the text alone. For example, the graph might reveal a secondary arc involving the Sea Witch's influence on the Little Mermaid's decisions, which runs parallel to the main storyline of her love for the Prince. These insights enrich the analysis by providing a more comprehensive view of the narrative structure, highlighting how different elements of the story interact and contribute to the overall plot.

4.5. Evaluation Metric

Quantitative metrics play a crucial role in evaluating the performance and effectiveness of the methodologies employed in this research. These metrics provide objective measures that help assess the accuracy, reliability, and robustness of both the event extraction process using BERT and the event relationship modeling using GCN. For instance in Table 3, event extraction accuracy measures how well BERT identifies and classifies events from the text. In this study, BERT achieved an accuracy of 92.5%, indicating its high proficiency in understanding and extracting relevant events from Andersen's fairy tales. This metric is essential for ensuring that the foundational data for further analysis is both precise and reliable.

Table 3. Evaluation Result

Metric	Value	Description
Event Extraction Accuracy	92.50%	The accuracy of BERT in correctly identifying events from the text.
Relationship Precision	89.30%	The precision of GCN in correctly identifying event relationships.
F1 Score (Event Linking)	90.80%	The balance between precision and recall for linking events in the graph.

Another critical metric is relationship precision, which evaluates the correctness of the relationships identified by the GCN. With a precision of 89.3%, the GCN demonstrates a strong ability to accurately model the connections between events. This metric is particularly important because it directly impacts the quality of the event relationship graph and, consequently, the insights derived from it. High precision ensures that the relationships depicted in the graph are meaningful and reflective of the actual narrative structure, thereby enhancing the credibility of the analysis.

The F1 score is another vital metric that balances precision and recall, providing a comprehensive measure of the model's performance in linking events. In this research, the F1 score for event linking is 90.8%, indicating a well-balanced performance where both the identification of relevant relationships (precision) and the coverage of all significant relationships (recall) are effectively managed. This balance is crucial for ensuring that the analysis captures the full spectrum of event interactions without overemphasizing or neglecting any particular aspect.

These quantitative metrics not only validate the effectiveness of the proposed methodologies but also provide a benchmark for future research. By achieving high scores in accuracy, precision, and F1, this study demonstrates the potential of combining BERT and GCN for narrative analysis. These metrics offer a clear and measurable way to compare different approaches and methodologies, facilitating continuous improvement and innovation in the field.

5. Conclusion

In conclusion, this study successfully demonstrates the effectiveness of combining BERT and Graph Convolutional Networks (GCN) for analyzing event relationships in Andersen's fairy tales. The integration of BERT's advanced contextual understanding and GCN's ability to model complex relational structures has enabled a deep and nuanced exploration of narrative patterns. The results reveal intricate causal, sequential, and thematic connections between events, highlighting Andersen's storytelling techniques, such as the use of linear progression, contrast, and recurring themes like sacrifice and transformation. Quantitatively, the models achieved high accuracy in event extraction (92.5%) and relationship identification (89.3% precision), underscoring the robustness of the proposed methodology. Qualitatively, the analysis provided valuable insights into the narrative structures, offering a new perspective on how events are interconnected to drive the plot and evoke emotional impact. These findings not only contribute to the field of computational linguistics by advancing techniques for narrative analysis but also enrich literary studies by providing a data-driven approach to understanding classic texts. Overall, this research underscores the potential of combining state-of-the-art NLP models with graph-based machine learning techniques to uncover hidden patterns in textual data. The methodology developed here can be extended to other genres and domains, paving the way for further interdisciplinary research at the intersection of literature, linguistics, and artificial intelligence.

Acknowledgment

We would like to express our deepest gratitude to all those who contributed to the successful completion of this research. First and foremost, we thank our academic advisors and mentors for their invaluable guidance, insightful feedback, and unwavering support throughout the research process. We are also grateful to our institution for providing the necessary resources and infrastructure to conduct this study. Special thanks go to the developers and contributors of the BERT and GCN frameworks, whose groundbreaking work made this research possible. Additionally, we acknowledge the literary scholars and researchers whose prior work on Andersen's fairy tales provided a strong foundation for our analysis. Finally, we extend our heartfelt appreciation to our colleagues, friends, and family for their encouragement and understanding during this endeavor. This research would not have been possible without their collective support.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper

References

[1] N. H. Varnier and R. F. de Lima Rodrigues, "A contribuição dos contos de fadas: um percurso entre o imaginário e a consciência de si na infância," *Res. Soc. Dev.*, vol. 9, no. 10, 2020, doi: 10.33448/RSD-V9I10.8242.

[2] A. M. Jorge, R. Campos, A. Jatowt, and S. Bhatia, "The 2nd International Workshop on Narrative Extraction from Text: Text2Story 2019," 2019, pp. 389–393, doi: 10.1007/978-3-030-15719-7_54.

[3] T. L. M. Suryanto, A. P. Wibawa, H. Hariyono, and A. Nafalski, "Comparative Performance of Transformer Models for Cultural Heritage in NLP Tasks," *Adv. Sustain. Sci. Eng. Technol.*, vol. 7, no. 1, p. 0250115, Jan. 2025, doi: 10.26877/asset.v7i1.1211.

[4] M. A. N. A. Latif, E. Pratiwi, and A. Prameswari, "Analyzing Early Childhood Fairy Tales: Looking for Studies on the Local Wisdom of Madurese Society," *Deleted J.*, vol. 10, no. 1, pp. 11–18, 2024, doi: 10.14421/al-athfal.2024.101-02.

[5] D. E. Cahyani, A. P. Wibawa, D. D. Prasetya, L. Gumilar, F. Akhbar, and E. R. Triyulinar, "Emotion Detection in Text Using Convolutional Neural Network," in *2022 International Conference on Electrical and Information Technology (IEIT)*, Sep. 2022, pp. 372–376, doi: 10.1109/IEIT56384.2022.9967913.

[6] L. He, Q. Zhang, J. Duan, and H. Wang, "An Open-Domain Event Extraction Method Incorporating Semantic and Dependent Syntactic Information," *Appl. Sci.*, vol. 13, no. 13, p. 7942, Jul. 2023, doi: 10.3390/app13137942.

[7] P. Zhu, Z. Wang, H. Wang, S. Li, and G. Zhou, "Event Detection with Document Structure and Graph Modelling," in *Natural Language Processing and Chinese Computing*, Springer International Publishing, 2020, pp. 593–603, doi: 10.1007/978-3-030-60450-9_47.

[8] T. Vo, "SynSeq4ED: A Novel Event-Aware Text Representation Learning for Event Detection," *Neural Process. Lett.*, vol. 54, no. 1, pp. 227–249, Feb. 2022, doi: 10.1007/s11063-021-10627-2.

[9] Y. Chen, T. Chen, S. Ebner, A. S. White, and B. Van Durme, "Reading the Manual: Event Extraction as Definition Comprehension," in *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, 2020, pp. 74–83, doi: 10.18653/v1/2020.spnlp-1.9.

[10] Agata Cybulska and P. Vossen, "Historical Event Extraction from Text," in *5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 2011, pp. 39–43. [Online]. Available at: https://aclanthology.org/W11-1506/.

[11] A. R. Hofer, "Sources: The Greenwood Encyclopedia of Folktales and Fairy Tales," *Ref. User Serv. Q.*, vol. 48, no. 1, pp. 94–95, Sep. 2008, doi: 10.5860/rusq.48n1.94.

[12] H. C. Andersen, *Andersen's Fairy Tales*, 1st ed., no. January 1999. Guttenberg, p. 36, 1999. [Online]. Available at: https://www.andersenstories.com/en/andersen_fairy-tales/list.

[13] B. Santana, R. Campos, E. Amorim, A. Jorge, P. Silvano, and S. Nunes, *A survey on narrative extraction from textual data*, vol. 56, no. 8. pp, 8393–8435, Springer Netherlands, 2023, doi: 10.1007/s10462-022-10338-7.

[14] A.-U.-N. Fatima, H. Ahmad, M. Ahmad, W. Ahmad, and N. Faisal, "Extraction of Temporal Events Frequency from Online News Channels," in *2020 30th International Conference on Computer Theory and Applications (ICCTA)*, pp. 1-25, Dec. 2020, doi: 10.1109/iccta52020.2020.9477659.

[15] A. B. P. Utama, A. P. Wibawa, A. N. Handayani, W. S. G. Irianto, Aripriharta, and A. Nyoto, "Improving Time-Series Forecasting Performance Using Imputation Techniques in Deep Learning," in *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, Jun. 2024, pp. 232–238, doi: 10.1109/SIML61815.2024.10578273.

[16] H. Zhang, H. Song, S. Wang, and B. Xu, "Bert (A Bert-Based End-To-End Model For Chinese Document-Level Event Extraction)," in *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, 2020, pp. 390–401. [Online]. Available at: https://aclanthology.org/2020.ccl-1.36/.

[17] Zecheng Zhang, Yuncheng Wu, and Zesheng Wang, "A Survey of Open Domain Event Extraction." p.30, 2020. [Online]. Available at: https://pdfs.semanticscholar.org.

[18] H. M. D. Trong, N. T. Ngo, L. Van, and T. Nguyen, "Selecting Optimal Context Sentences for Event-Event Relation Extraction," *Proc. ... AAAI Conf. Artif. Intell.*, vol. 36, no. 10, pp. 11058–11066, 2022, doi: 10.1609/aaai.v36i10.21354.

[19] R. Duan, P. Guo, and D. Zhang, "Multi-Grained Ranking Network for Event Extraction," *J. Phys. Conf. Ser.*, vol. 2347, no. 1, p. 12023, Sep. 2022, doi: 10.1088/1742-6596/2347/1/012023.

[20] R. Han, Y. Zhou, and N. Peng, "Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5717–5729, doi: 10.18653/v1/2020.emnlp-main.461.

[21] Q. Li *et al.*, "A Survey on Deep Learning Event Extraction: Approaches and Applications," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 5, pp. 6301–6321, May 2024, doi: 10.1109/TNNLS.2022.3213168.

[22] C. Heitzinger and S. Woltran, "A Short Introduction to Artificial Intelligence: Methods, Success Stories, and Current Limitations," 2023, pp. 135–149, doi: 10.1007/978-3-031-45304-5_9.

[23] A. Bendimerad, M. Plantevit, C. Robardet, and S. Amer-Yahia, "User-Driven Geolocated Event Detection in Social Media," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 796–809, 2021, doi: 10.1109/TKDE.2019.2931340.

[24] J. Vasilakes, P. Georgiadis, N. T. H. Nguyen, M. Miwa, and S. Ananiadou, "Contextualized medication event extraction with levitated markers," *J. Biomed. Inform.*, vol. 141, no. March, p. 104347, 2023, doi: 10.1016/j.jbi.2023.104347.

[25] J. Liu, Y. Chen, K. Liu, W. Bi, and X. Liu, "Event Extraction as Machine Reading Comprehension," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1641–1651, doi: 10.18653/v1/2020.emnlp-main.128.

[26] Z. Li, N. Cheng, and W. Song, "Research on Chinese Event Extraction Method Based on RoBERTa-WWM-CRF," pp. 100-104, Aug. 2021, doi: 10.1109/icsess52187.2021.9522150.

[27] Y. Chen, L. Kong, Y. Wang, and D. Kong, "Multi-Grained Attention Representation With ALBERT for Aspect-Level Sentiment Classification," *IEEE Access*, vol. 9, pp. 106703–106713, 2021, doi: 10.1109/ACCESS.2021.3100299.

[28] J. Li, R. Hu, K. Zhang, H. Liu, and Y. Ma, "DEERE: Document-Level Event Extraction as Relation Extraction," *Mob. Inf. Syst.*, vol. 2022, pp. 1–8, 2022, doi: 10.1155/2022/2742796.

[29] A. Shreyas, G. Priyanka, Merlyn Pearl, and Shinde Swapnal, "Event Information Extraction From E-Mail And Updating Event In Calendar," *Int. J. Adv. Res. Innov. Ideas Educ.*, vol. 4, no. 3, pp. 1242-1247, 2018. [Online]. Available at: https://ijariie.com/AdminUploadPdf/ .

[30] F. Li, G. Chen, and X. Liu, “An event-based automatic annotation method for datasets of interpersonal relation extraction,” *Appl. Intell.*, vol. 53, no. 3, pp. 2629–2639, 2023, doi: 10.1007/s10489-022-03547-8.

[31] Q. Wan, C. Wan, K. Xiao, R. Hu, D. Liu, and X. Liu, “CFERE: Multi-type Chinese financial event relation extraction,” *Inf. Sci. (Ny).*, vol. 630, no. January, pp. 119–134, 2023, doi: 10.1016/j.ins.2023.01.143.

[32] Z. Yan and X. Tang, “Narrative Graph: Telling Evolving Stories Based on Event-centric Temporal Knowledge Graph,” *J. Syst. Sci. Syst. Eng.*, vol. 32, no. 2, pp. 206–221, 2023, doi: 10.1007/s11518-023-5561-0.

[33] W. Liu *et al.*, “Beyond Single-Event Extraction: Towards Efficient Document-Level Multi-Event Argument Extraction,” *ArXiv*, vol. 1, pp. 1-18, no. 2, 2023. [Online]. Available at: https://arxiv.org/abs/2405.01884.

[34] A. Romadhony, D. H. Widyantoro, and A. Purwarianti, “Utilizing structured knowledge bases in open IE based event template extraction,” *Appl. Intell.*, vol. 49, no. 1, pp. 206–219, 2019, doi: 10.1007/s10489-018-1269-0.

[35] R. N. Devendra Kumar, K. Srihari, C. Arvind, and W. Viriyasitavat, “Biomedical event extraction on input text corpora using combination technique based capsule network,” *Sādhanā*, vol. 47, no. 4, p. 198, Sep. 2022, doi: 10.1007/s12046-022-01978-0.

[36] G. Li, P. Wang, J. Xie, R. Cui, and Z. Deng, “FEED: A Chinese Financial Event Extraction Dataset Constructed by Distant Supervision,” in *Proceedings of the 10th International Joint Conference on Knowledge Graphs*, Dec. 2021, pp. 45–53, doi: 10.1145/3502223.3502229.

[37] B. Portelli, D. Passabi, E. Lenzi, G. Serra, E. Santus, and E. Chersoni, “Improving Adverse Drug Event Extraction with SpanBERT on Different Text Typologies,” 2022, pp. 87–99, doi: 10.1007/978-3-030-93080-6_8.

[38] H.-Y. Yu and M.-H. Kim, “Automatic Event Extraction method for Analyzing Text Narrative Structure,” in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2021, pp. 1–4, doi: 10.1109/IMCOM51814.2021.9377386.

[39] P. Purnawansyah *et al.*, “Congestion Predictive Modelling on Network Dataset Using Ensemble Deep Learning,” *J. Appl. Data Sci.*, vol. 5, no. 4, pp. 1597–1613, Dec. 2024, doi: 10.47738/jads.v5i4.333.

[40] A. P. Wibawa *et al.*, “Decoding and preserving Indonesia’s iconic Keris via A CNN-based classification,” *Telemat. Informatics Reports*, vol. 13, p. 100120, Mar. 2024, doi: 10.1016/J.TELER.2024.100120.