Narrative Extraction from Semantic Graphs

Daniil Lystopadskyi
Mestrado em Ciência de Computadores
Departamento de Ciência de Computadores
2023

Orientador
José Paulo de Vilhena Geraldes Leal, Professor Auxiliar
Faculdade de Ciências da Universidade do Porto

Abstract

Narratives have long served as integral tools for education, communication, knowledge management, and cultural transmission through human history. Storytelling, as the art of conveying ideas and experiences through narratives, has evolved from ancient visual narratives to contemporary digital formats. The advancement of computational linguistics and Natural Language Processing (NLP) has greatly contributed to the extraction of narratives from textual data, yet the potential of leveraging semantic graphs for this purpose remains untapped.

This thesis explores the intersection of narrative extraction and semantic graphs, aiming to bridge the gap between structured representations and the fluidity of traditional narratives. It addresses the challenges of graph complexity, data sparsity, and domain-specific knowledge scarcity.

The objectives of this research were split into multiple research questions that include defining what constitutes a narrative in the context of semantic graphs, identifying relevant information within a semantic graph, constructing a narrative from this information, and developing a validation method for narrative extraction. Each research question represents a crucial aspect of the narrative extraction process.

The primary contribution of this thesis is the Narrative Extractor from Semantic Graphs (NESG) approach, which harnesses semantic graphs to capture both semantic and structural narrative information. It offers a user-friendly configuration feature, enabling users to manually set up their input semantic graphs quickly. Once configured, the narrative extraction process is largely automated, relying on main entity identification, relevant entity extraction, entity classification, event identification, event attribute mapping, and narrative building. To standardize the representation of extracted narratives, a narrative ontology has been developed.

Evaluating narratives in graph format presents unique challenges due to subjectivity and structural complexity. The evaluation procedure encompasses narrative and event analysis, utilizing comprehensive data gathering and statistics to assess narrative quality. The results are promising, indicating successful narrative extraction across different genres and domains, with room for improvement, especially in specific narrative genres. This research demonstrates the feasibility of leveraging semantic graphs for narrative extraction and contributes to the evolving field of narrative analysis and knowledge extraction.

Resumo

Narrativas têm servido desde há muito tempo como ferramentas essenciais para a educação, comunicação, gestão do conhecimento e transmissão cultural ao longo da história humana. A narrativa, como a arte de transmitir ideias e experiências através de histórias, evoluiu desde narrativas visuais antigas até formatos digitais contemporâneos. O avanço da linguística computacional e Natural Language Processing (NLP) contribuiu significativamente para a extração de narrativas a partir de dados textuais, no entanto, o potencial de aproveitar grafos semânticos para este fim permanece inexplorado.

Esta tese explora a interseção entre a extração de narrativas e grafos semânticos, com o objetivo de estreitar a lacuna entre representações estruturadas e a fluidez das narrativas tradicionais. Ela aborda os desafios da complexidade dos grafos, da escassez de dados e do conhecimento específico de domínio.

Os objetivos desta pesquisa foram divididos em várias questões de pesquisa que incluem a definição do que constitui uma narrativa no contexto de grafos semânticos, a identificação de informações relevantes dentro de um grafo semântico, a construção de uma narrativa a partir dessas informações e o desenvolvimento de um método de validação para a extração de narrativas. Cada questão de pesquisa representa um aspecto crucial do processo de extração de narrativas.

A principal contribuição desta tese é a abordagem Narrative Extractor from Semantic Graphs (NESG), que utiliza grafos semânticos para capturar informações narrativas tanto semânticas quanto estruturais. Ela oferece um recurso de configuração fácil de usar, permitindo que os usuários configurem rapidamente seus grafos semânticos manualmente. Uma vez configurado, o processo de extração de narrativas é em grande parte automatizado, baseando-se na identificação de entidade principal, extração de entidades relevantes, classificação de entidades, identificação de eventos, mapeamento de atributos de eventos e construção de narrativas. Para padronizar a representação das narrativas extraídas, uma ontologia foi desenvolvida.

A avaliação de narrativas no formato de grafo apresenta desafios únicos devido à subjetividade e complexidade estrutural. O procedimento de avaliação engloba a análise de narrativas e eventos, utilizando uma coleta de dados abrangente e estatísticas para avaliar a qualidade das narrativas. Os resultados são promissores, indicando uma extração de narrativas bem-sucedida em diferentes géneros e domínios, com espaço para melhorias, especialmente em gêneros de narrativas específicos. Esta pesquisa demonstra a viabilidade de aproveitar grafos semânticos para a extração de narrativas e contribui para o campo em evolução da análise de narrativas e extração de conhecimento.

Contents

List of Tables

List of Figures

Listings

Acronyms

DCC Departamento de Ciência de Computadores

FCUP Faculdade de Ciências da Universidade do Porto

HTTP Hypertext Transfer Protocol

NLP Natural Language Processing

RDF Resource Description Framework

SPARQL SPARQL Protocol and RDF Query Language

URI Uniform Resource Identifier

BERT Bidirectional Encoder Representations from Transformers

BFS Breadth-first search

GUI Graphical User Interface

NESG Narrative Extractor from Semantic Graphs

SLATE Symposium on Languages, Applications, and Technology

API Application Programming Interface

Chapter 1

Introduction

Narratives, an ancient and enduring form of human expression, have served as fundamental tools for education [29], communication [24], knowledge management [39] and cultural transmission [35] throughout the annals of human history. From captivating tales spun around the campfires to the intricate web of narratives woven through the pages of books, stories have played a pivotal role in shaping our understanding of the world and ourselves [1].

At its core, storytelling is the art of conveying ideas, beliefs, and experiences to others, often through spoken or written narratives. These narratives encapsulate a sequence of events, whether real or fantastical, forming the essence of the storyteller's craft.

The evolution of storytelling as a means of communication has been a dynamic journey, tracing its roots to ancient visual narratives such as cave paintings[11]. Over time, storytelling transitioned to oral traditions, where stories were passed down through generations via the spoken word. This evolution marked a gradual shift toward more structured and systematized narratives, spanning written, printed, typed, and, more recently, digital formats [7].

In a similar vein, the present thesis can be viewed as a narrative in itself. It possesses a clear beginning and a conclusive ending. The contents within this thesis unfold in a natural progression, guiding the reader from the inception to the conclusion through an interconnected series of chapters and sections. Each chapter, much like a plot line in a larger story, contains a narrative of its own, contributing to the coherence of the entire thesis. Each of these chapters plays a vital role in illuminating various facets of the overarching narrative presented here.

As narratives have evolved, so too have the tools and methodologies for understanding and extracting their essence. In recent years, the field of computational linguistics and Natural Language Processing (NLP) have taken significant strides in unraveling the intricate threads of narratives hidden within textual and multimedia data [28, 31]. Yet, despite the remarkable progress in narrative extraction from plain text, the potential of leveraging semantic graphs - structured representations of entities and their relationships - for this purpose remains largely untapped, brimming with unique challenges and opportunities [5].

In the remarkable evolution of narratives, we have witnessed their ability to unearth in sources that, on the surface, seem to defy the very principles that define narratives themselves. Semantic (or knowledge) graphs, with their structured, logical representations, stand in stark contrast to the inherent creativity, fluidity, and interpretative ambiguity that often characterize traditional narratives. However, it is from this paradox that we can draw strength to help us better understand the rich semantic information embedded in those narratives.

Semantic graphs, with their inherent structure, offer a more easily understandable and computationally accessible source of information. Despite the potential benefits of leveraging semantic graphs for narrative extraction, existing methods encounter several challenges. The complexity of graph structures, the sparsity of data, and the scarcity of domain-specific knowledge pose significant hurdles in effectively extracting narratives [5]. These challenges necessitate the development of novel techniques and methodologies tailored to the unique characteristics of semantic graphs.

1.1 Motivation

Narratives play a vital role in describing the world around us [15]. They act as a tool for making sense out of, rather, non-sensical situations[12]. Narratives can be used to communicate, share, and capitalize on the knowledge of individuals[34]. They are widely used as a teaching tool to convey knowledge in a practical manner[8].

By defining narratives as graphs, we establish a universal representation that simplifies information access and transfer between different systems. Those narratives can be repurposed for a variety of applications, such as modeling human mental states [20] by explicitly modeling the motivations, interactions, and internal states of characters of a narrative. Narrative, as well as, event-centric [16] graphs, can also be used for future event prediction by analyzing and learning event patterns [10, 17].

1.2 Objectives

This work aims to address the aforementioned challenges and contribute to the field of narrative extraction from semantic graphs. More specifically, we defined four key research questions that we aim to answer throughout this research.

- **Q1.** What is a narrative and how can we represent it using semantic graphs?
- **Q2.** How to identify information within a semantic graph relevant to a narrative?
- **Q3.** How can we build a narrative using the information found in the semantic graph?
- **Q4.** How can we validate our work?

Each one of those questions represents a major, yet necessary, part of the narrative extraction process.

1.3 Contribution

In this thesis, we propose NESG (Narrative Extractor from Semantic Graphs), an innovative approach for narrative extraction that leverages the nature of semantic graphs to capture both the semantic and structural information inherent in narratives, aiming to answer the research questions **Q2** and **Q3**. To address the primary challenges associated with this task, our approach incorporates a user-friendly feature that empowers users to configure their own semantic graphs manually, providing a partial answer to the research question **Q2**. The configuration process is straightforward, requiring only a matter of minutes to complete. Once the graph is configured, the narrative extraction process can begin, with minimal to no human interaction necessary. Each graph only needs to be configured once and can be modified at any point. The rest of the extraction procedure relies on graph search, using several **sparql!** (**sparql!**) queries, coupled with string matching and rule-based techniques. The search function can be further enhanced by manually tuning a vast number of available parameters. The final product of our algorithm is a new semantic graph, built from the data tied to the extracted narrative. To facilitate the representation of such a graph, we developed our own narrative ontology that provides a standardized framework for all extracted narratives, answering the research question **Q1**.

Evaluating narratives, particularly when represented in graph format, presents a set of intricate challenges. The inherent nature of narratives as subjective and context-dependent constructs complicate the establishment of rigorous evaluation metrics [30]. Moreover, semantic graphs introduce structural complexities that demand specialized evaluation methods [5].

Our evaluation procedure encompasses a diverse array of techniques for more precise and refined validation results in order to better answer the research question **Q4**. In the initial phase, we conducted a comprehensive data-gathering exercise, accumulating pertinent statistics from a curated selection of extracted narratives, which we can utilize to assess narrative quality. The validation step was split into two parts: narrative analysis and event analysis. Narrative analysis focuses on the overall performance of our approach, by performing a qualitative and quantitative evaluation of the extracted narratives. Event analysis provides insight into individual steps of our approach, allowing us to evaluate the algorithm at different stages.

Our approach showed promising results when it comes to narrative extraction across different genres and domains. However, during the evaluation, it was deduced that there is, still, a large room for improvement, mainly when it comes to specific narrative genres. Overall, we managed to produce narratives with satisfactory quality, proving that we were able to achieve our goals.

1.4 Organization

The remainder of this thesis is organized as follows:

- **Background Chapter**: In this chapter, we provide background information on essential concepts related to narrative extraction and semantic graphs.
- **State-of-the-Art Chapter**: This chapter offers an overview of existing work in the field, discussing the strengths and limitations of various approaches.
- **Design and Development Chapter**: In this chapter we present our approach for narrative extraction, detailing the techniques and methodologies employed.
- **Implementation Chapter**: This chapter includes practical implementation of our approach, describing tools used and the layout of our application.
- **Evaluation Chapter**: This chapter presents the experimental results, analyzing the quality of our approach across different domains and genres.
- **Conclusion Chapter**: This chapter concludes the thesis, summarizing the key findings and contributions of this work, as well as discussing potential future research directions, outlining avenues for further exploration and improvement.

Chapter 2

Basic Concepts

This section provides an overview of key concepts that are essential to fully comprehend the context of the field of study related to narrative extraction from semantic graphs. We begin by defining broader concepts, such as graphs and the semantic web, followed by more specific and specialized constructs.

2.1 Graphs

Graphs are intricate visual structures that represent complex relationships and connections between entities or data points. They serve as a powerful means of organizing and encoding information, consisting of nodes, which represent individual entities or elements, and edges, which depict the connections or relationships between these entities. These relationships can take various forms, ranging from simple binary connections to more complex associations.

Definition 2.1.1 (Graph Nodes)**.** Nodes are the fundamental building blocks of a normal graph. Each node represents a distinct entity, object, or data point.

For instance, in a social network graph, nodes could represent users, while in a transportation network graph, nodes might represent locations or junctions.

Definition 2.1.2 (Graph Edges)**.** Edges are the connectors between nodes and signify the relationships or interactions between them. Edges can be directional or undirected, weighted or unweighted, and they provide critical information about how nodes are connected.

In a social network graph, edges could represent friendships (undirected) or follower relationships (directed), while in a road network graph, edges might represent road segments between locations.

Graphs are versatile and find applications in various domains. They are employed in network analysis, social network modeling, transportation and logistics, recommendation systems, and more. Their simplicity and flexibility make them a valuable tool for representing and analyzing complex relationships and systems. Graphs are a foundational concept in graph theory and data analysis, serving as the basis for more specialized graph structures like semantic graphs.

2.2 Semantic Web

The Semantic Web is a vision of the future of the Internet in which information is, not only stored and presented for human consumption but also structured and linked in a way that can be understood and processed by machines. It aims to create a more intelligent, interconnected, and meaningful web by adding a layer of semantic information to the existing web content. The Semantic Web can be divided into the following components: Resource Description Framework (RDF), Ontologies, Uniform Resource Identifier (URI), Linked Data, and SPARQL Protocol and RDF Query Language (SPARQL). The following sub-sections provide more insight into each component.

2.2.1 Resource Description Framework

RDF is the backbone of the Semantic Web. It is a standardized data format used to describe resources on the web and the relationships between them. RDF provides a simple and flexible way to represent data as triples: **subject-predicate-object**. In this structure, the subject is the resource being described, the predicate signifies the relationship and the object is the target resource or value. For example, consider an RDF triple:

Subject	**Predicate**	**Object**
John	knows	Mary

This simple triple conveys a semantic relationship, stating that John knows Mary. RDF allows for the creation of structured, interconnected data by representing information in this format, thus enabling machines to understand and process it.

Now, consider a larger RDF dataset, which can be found in table 2.1, representing information about an individual, in this case, "Albert Einstein", including information such as "place of birth", "date of birth", "educated at", "date of death" and "father". Relationships can target other resources, in this case, "Ulm", "University of Zurich" and "Herman Einstein", or literal values, such as dates, labels, descriptions, and so on. Objects in RDF triples can also appear as subjects in other RDF triples. This can be witnessed in table 2.1, where there are resources, such as "University of Zurich", showing up as both subjects and objects, in different triples.

By examining these RDF triples, it becomes evident how RDF enables the representation of structured information and relationships, by defining an environment and listing facts within it. The simplicity of RDF and expressiveness make it a powerful tool for encoding diverse knowledge structures, fostering data interoperability, and enabling efficient data integration and querying on the web.

Table 2.1: Albert Einstein RDF example.

Subject	Predicate	Object
Albert Einstein	place of birth	Ulm
Albert Einstein	date of birth	14 March 1879
Albert Einstein	educated at	University of Zurich
Albert Einstein	date of death	18 April 1955
Albert Einstein	father	Hermann Einstein
Hermann Einstein	date of birth	30 August 1847
Hermann Einstein	mother	Helem Einstein
Hermann Einstein	occupation	Entrepreneur
University of Zurich	country	Switzerland
University of Zurich	student count	25,732

2.2.2 Semantic Graphs

Semantic graphs, sometimes also referred to as knowledge graphs, store domain/context-specific information about concepts and relations between them. The information stored in semantic graphs can also be viewed as a set of RDF triples, where each triple corresponds to (Subject, Predicate, Object). In graph terms, these triples are represented as (Node, Edge, Node).

Semantic graphs are structured representations of data where nodes represent entities and edges (also known as predicates or relationships) denote connections between these entities. These connections carry semantic meaning, which defines how entities are related to each other.

The use of RDF as a data model in the Semantic Web allows for the creation of semantic graphs that capture the underlying semantics of data, making it easier for machines to understand and interpret the relationships between entities. RDF provides a standardized way to encode and exchange data with semantic meaning, which is a key aspect of the vision of the Semantic Web.

In essence, semantic graphs are a specific type of RDF graph that emphasizes the representation of semantic relationships between entities, aligning with the core principles of the Semantic Web. They play a pivotal role in enabling the goal of the Semantic Web of enhancing data interoperability and enabling more intelligent data processing on the web.

An example of a small semantic graph can be found in figure 2.1, where we can find entities such as "Albert Einstein" and "University of Zurich", represented as nodes in the graph, as well as relationships between different entities. For instance, this graph suggests that "Albert Einstein" was educated at "University of Zurich" and was a child of "Hermann Einstein", among other facts.

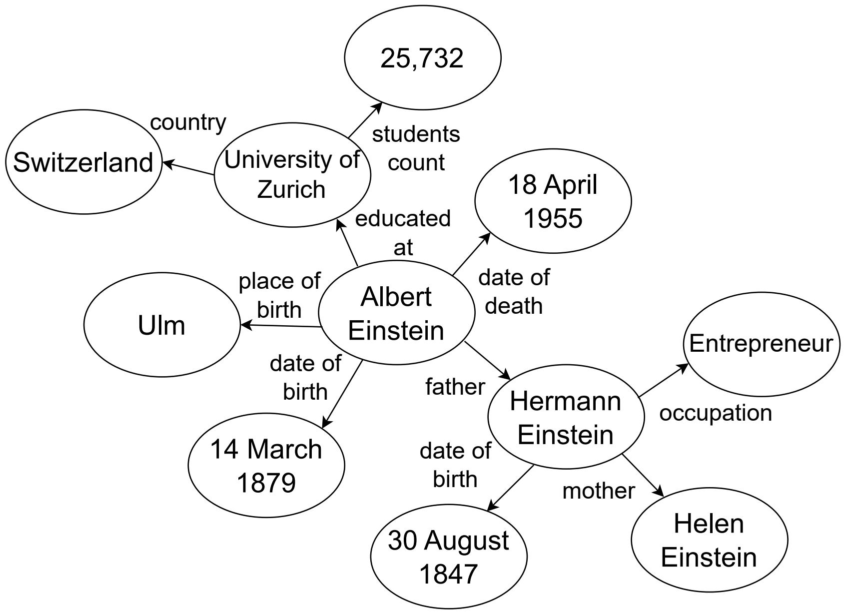


Figure 2.1: Semantic Graph.

2.2.2.1 Semantic vs Regular Graphs

As the name suggests, semantic graphs are specialized versions of regular graphs, which were adapted to serve the needs of the Semantic Web. The main differences between these two types of graphs can be found in table 2.2.

2.2.3 Ontologies

Semantic graphs structure information according to a web ontology. In the context of web semantics, an ontology refers to a standardized method of defining the hierarchy of concepts and relationships within a specific domain. Ontologies employ a set of classes, properties, and constraints to establish a common vocabulary for representing knowledge. By utilizing ontologies, semantic graphs ensure that the information contained within them is structured and organized, enabling more effective data management and analysis.

Ontologies are fundamental constructs in the realm of knowledge representation and artificial intelligence, providing a structured framework for modeling concepts, relationships, and the semantics of specific domains. They play a pivotal role in various fields, including computer science, information systems, and the Semantic Web. The following subsections detail the main features of ontologies.

Table 2.2: Differences between semantic and regular graphs

	Semantic Graphs	**Regular Graphs**
Purpose and Semantics	Designed to capture, not only the structural relationships between entities, but also their **semantic** meaning. They use standardized vocabularies and ontologies to provide rich context and semantics to the data.	Focus on representing **structural** relationships between nodes and edges. While they can model relationships, they do not inherently encode semantic meaning.
Representation	Relationships are explicitly defined using triples (subject-predicate-object) in RDF.	Relationships between nodes are typically represented without specific semantic meaning. Edges connect nodes, but the interpretation of these connections relies on external context or domain knowledge.
Interoperability	Designed to be highly interoperable. The use of standardized ontologies and RDF allows data to be easily integrated and linked across different domains and sources.	Do not inherently provide a standardized framework for interoperability. Their interpretations often depend on the specific context in which they are used.
Complexity and Expressiveness	High level of expressiveness due to inclusion of semantic meaning. They can represent intricate knowledge structures and support complex querying and reasoning.	Simpler in structure and representation. Excel in modeling straightforward relationships, but may lack the depth of expressiveness found in semantic graphs.
Applications	Applications that demand precise semantics, such as the Semantic Web, knowledge graphs, and ontology modeling.	Applications in diverse fields.

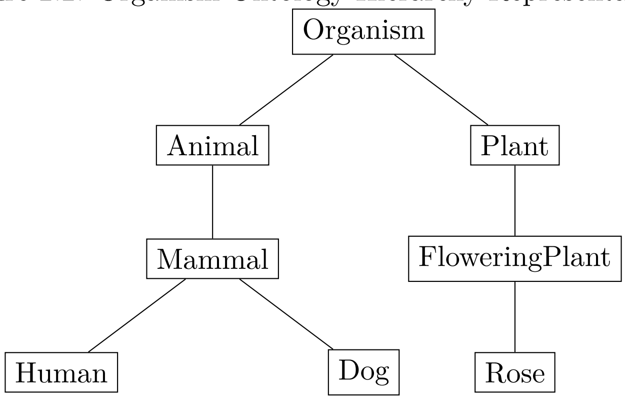
2.2.3.1 Key Components of Ontologies

- **Concepts (Classes or individuals)**: Concepts represent entities or categories within a domain. For example, in a medical ontology, "Disease" and "Medication" are concepts.
- **Properties (Attributes or Relations)**: Properties describe the characteristics or relationships between concepts. In a real estate ontology, `hasPrice` could be a property linking a "Property" concept to a numerical value.
- **Individuals (Instance)**: Individuals are specific instances of concepts. In a geographical ontology, "Paris" is an individual of the concept "City".
- **Axioms and Constraints**: Axioms establish formal relationships and constraints between concepts and properties, enhancing the expressiveness of ontologies.

2.2.3.2 Hierarchical Structure

Ontologies typically employ a hierarchical structure, organizing concepts into broader and narrower categories. Consider the example of a biological ontology that can be found in Figure 2.2. In this hierarchy, concepts become more specific as you move down the tree. For instance, "Human" is a more specific type of "Mammal" and "Mammal" is a more specific type of "Animal".

Figure 2.2: Organism Ontology Hierarchy Representation



2.2.3.3 Relationships

Ontologies capture various relationships between concepts and individuals:

- **Subclass Relationships**: Express hierarchical relationships where one concept is a subclass of another. For instance, "Dog" is a subclass of "Mammal", as seen in figure 2.2.
- **Property Relationships**: Define how properties relate to concepts. For example, `hasLegs` could be a property relating the concept of "Animal" to a numerical value representing the number of legs.
- **Equivalence Relationships**: Specify that two concepts are equivalent in meaning. For instance, "United States" and "USA" could be declared as equivalent individuals.

2.2.3.4 Example: OWL Ontology for Vehicles

Suppose we wanted to create an ontology for representing vehicle-related information. In order to accomplish that, we would need the following concepts and properties:

- **Concepts**:
  - "Vehicle"
  - "Car"
  - "Motorcycle"

  - "Bicycle"

- **Properties**:
  - `hasColor` (relates vehicles to their colors)
  - `hasWheels` (relates vehicles to the number of wheels)

- **Individuals**:
  - `RedCar` (an instance of "Car" with the property `hasColor` set to "Red" and `hasWheels` set to 4)
  - `BlueMotorcycle` (an instance of "Motorcycle" with the property `hasColor` set to "Blue" and `hasWheels` set to 2)

2.2.3.5 Semantic Web and Interoperability

Ontologies are integral to the Semantic Web, where they facilitate data integration and interoperability. When different systems use shared ontologies, they can understand and exchange data more effectively. For example, a product ontology allows e-commerce websites to exchange product information seamlessly.

2.2.3.6 Inference and Reasoning

Ontologies enable automated reasoning and inference. Given our vehicle ontology, an inference engine can deduce that a "Car" is a "Vehicle" and inherit properties and relationships accordingly.

In conclusion, ontologies serve as a powerful tool for modeling and structuring knowledge in a formal, machine-readable manner. They provide a common vocabulary for domains, enable data integration, and support intelligent reasoning, making them indispensable in fields where precise representation and understanding of concepts and relationships are paramount,

2.2.4 SPARQL

SPARQL is a powerful query language and protocol for querying, retrieving, and manipulating data stored in RDF format. SPARQL enables precise and expressive querying of RDF data, allowing users to extract meaningful information. In the following subsections, we will discuss the key concepts that makeup SPARQL.

2.2.4.1 Basic Components of SPARQL Queries

- **SELECT**: Specifies the variables you want to retrieve in the query results.
- **WHERE**: Defines the patterns to match in the RDF data.

- **FILTER**: Enables conditions for filtering results.
- **ORDER BY**: Sorts the query results based on specific criteria.
- **LIMIT**: Restricts the number of results to retrieve.
- **PREFIX**: Declares namespace prefixes for concise URIs in queries.

2.2.4.2 Basic SPARQL Query Structure

A typical SPARQL query follows this structure:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?subject ?predicate ?object WHERE {
  ?subject ?predicate ?object .
}
```

Listing 2.1: Sample SPARQL Query

This simple query retrieves all triples in the RDF data, returning three variables: `?subject`, `?predicate`, and `?object`.

2.2.4.3 Example: Retrieving Data

Suppose we have an RDF dataset describing books, authors, and genres. Here is an example query to retrieve all book titles and their authors in Listing 2.2.

```
PREFIX ex: <http://example.org/>

SELECT ?bookTitle ?authorName
WHERE {
  ?book ex:hasTitle ?bookTitle .
  ?book ex:hasAuthor ?author .
  ?author ex:hasName ?authorName .
}
```

Listing 2.2: Book title query

In this query, we use **PREFIX** to declare a namespace for our dataset and then **SELECT** to specify the variables we want in the result. The **WHERE** clause defines a pattern to match: a book with a title and an author with a name. The query retrieves book titles (`?bookTitle`) and author names (`?authorName`).

2.2.4.4 Filtering Data with Filter

SPARQL allows you to filter results using **FILTER**. For example, you can modify the previous query to retrieve only books published after 2000, as we can see in Listing 2.3.

```
PREFIX ex: <http://example.org/>

SELECT ?bookTitle ?authorName
WHERE {
  ?book ex:hasTitle ?bookTitle .
  ?book ex:hasAuthor ?author .
  ?author ex:hasName ?authorName .
  ?book ex:hasPublicationDate ?date .
  FILTER (?date > "2000-01-01"^^xsd:date)
}
```

Listing 2.3: SPARQL Filter Example

Here, we filter results based on the publication date, using the **FILTER** clause to include only books published after January 1, 2000.

2.2.4.5 Joining and Ordering Data

SPARQL also allows you to join data from different parts of the RDF graph and order results. Consider the query in Listing 2.4 to retrieve book titles and authors, ordered by the book title.

```
PREFIX ex: <http://example.org/>

SELECT ?bookTitle ?authorName
WHERE {
  ?book ex:hasTitle ?bookTitle .
  ?book ex:hasAuthor ?author .
  ?author ex:hasName ?authorName .
}
ORDER BY ?bookTitle
```

Listing 2.4: SPARQL Join Example

This query combines data from books and authors and orders the results alphabetically by book title.

2.2.4.6 Aggregating Data

SPARQL supports aggregating data, allowing you to perform operations like counting, summing, or averaging values. In Listing 2.5 is an example of counting the number of books by each author:

```
PREFIX ex: <http://example.org/>

SELECT ?authorName (COUNT(?book) AS ?bookCount)
WHERE {
  ?book ex:hasAuthor ?author .
  ?author ex:hasName ?authorName .
}
GROUP BY ?authorName
```

Listing 2.5: SPARQL Aggregation Example

In this query, we use **COUNT** to aggregate data, grouping the results by author name to count the number of books each author has written.

The expressive querying capabilities of SPARQL make it a vital tool for navigating and extracting meaningful information from RDF data, contributing to the effective utilization of Semantic Web technologies in diverse domains.

By understanding these fundamental concepts, researchers can grasp the underlying principles of narrative extraction from semantic graphs. The utilization of semantic graphs as a structured source of information offers unique opportunities for capturing narrative coherence and extracting meaningful insights. In the subsequent sections, we will explore the existing research landscape, outline our proposed approach, and present experimental results to demonstrate the effectiveness and adaptability of our methodology.

Chapter 3

State of the Art

Narrative extraction, a pivotal task in Natural Language Processing (NLP) and computational linguistics, has seen significant advancements driven by the proliferation of textual and multimedia content on the web. Extracting narratives involves identifying and structuring coherent and meaningful stories from unstructured data sources, and facilitating applications such as information retrieval, summarization, storytelling, and question-answering. Here, we delve into the state of the art in narrative extraction, drawing on key works and methodologies to provide a comprehensive overview.

In this chapter, we go over the key methodologies employed in the field of narrative extraction. We start by presenting the main NLP related approaches, by describing their general workflow and explaining the contribution of each work. We follow up by introducing semantic graphs for the task of narrative extraction and presenting a few methodologies that aim to leverage these graphs for this task. We then showcase approaches that seek benefits from both NLP and semantic graphs in order to extract narratives. Additionally, we reference some approaches that utilize different multimedia sources. Finally, we delve into different evaluation techniques employed in the field of narrative extraction and semantic graph analysis, finishing off by summarizing the main challenges of existing approaches and avenues for future work.

3.1 Natural Language Processing Techniques

NLP-based methods have been at the forefront of narrative extraction research [33], employing various techniques and stages to extract narratives from textual data. These approaches typically involve several stages, including pre-processing and parsing, identification and extraction of narrative components, linking components, representation of narratives, and evaluation.

Pre-processing and parsing stages, generally, involve text normalization and tokenization. These initial steps aim to transform unstructured text into a structured format that can be further analyzed and processed. The identification and extraction of narrative components involve techniques such as named entity recognition, event detection, and coreference resolution.

These techniques help identify key entities and events that form the basis of narratives.

The linking components stage focuses on establishing relationships and connections between the identified entities and events. This can involve semantic role labeling, which assigns roles to the participants in events, and event coreference resolution, which links similar events across different texts or documents. The representation of narratives stage aims to capture the extracted information in a structured and coherent manner.

Narrative extraction based on NLP spans out across a wide field of studies. One example of an NLP-based approach is the extraction of narratives from administrative records, as demonstrated in the work by Megerdoomian et al. [25]. The authors employed Apache cTAKES and other NLP tools to extract clinical terms, events, and relationships related to mental health and substance use from the text. Their system also involves domain-specific entity and event identification, considering source attribution and negation detection. Their work highlights the applicability of NLP-based methods in extracting narratives from domain-specific textual data.

Anantharama et al. [3] discusses the concept of narrative modeling and its potential applications, particularly in the field of social sciences. The authors highlight the importance of understanding how narratives evolve over time and how they can be analyzed using computational techniques. The article introduces CANarEx, a contextually-aware narrative modeling approach that uses Transformer models. The authors compare CANarEx to existing methods and demonstrate its superiority. The framework involves steps like co-reference resolution, micro-narrative extraction, and narrative clustering. The article also mentions the application of CANarEx to datasets related to "First Nations" in Australia, showing the progression through each step of the framework and the resulting micro-narratives.

Norambuena and Mitra [27] achieve narrative representation and extraction from large-scale online data, particularly in the context of new narratives. The authors propose a computational representation for narratives based on graph theory, with a focus on directed acyclic graphs to reflect the temporal structure of stories. They also develop an extraction algorithm that maximizes the coherence of the narrative map while considering structural and topic coverage constraints.

Hussain et al. [19] discuss the challenge of misinformation on social media, particularly during the COVID-19 pandemic, and propose a narrative visualization tool for blogs. The authors contribute by developing a tool that extracts and visualizes narratives from social media content, aiding in the understanding and analysis of narratives. They also detail a research methodology involving named entity extraction, network topic modeling, and NLP to extract narratives. Their tool aims to combat misinformation and enhance the ability of the user to explore and provide feedback on narratives.

Edwards et al. [9] present their work in which they aim to gain deeper insight into narratives and improve them. The authors focus on analyzing social networks within narratives, with characters as nodes and their interactions as edges. They claim that extracting social networks from unstructured text sources, like scripts or novels, is challenging. The paper compares three techniques for extracting social networks from TV scripts: manual extraction, NLP, and co-occurrence networks. The study uses the TV series "Friends" as a case study and presents findings on network properties and interactions between characters. The authors conclude that automatic extraction methods (NLP and co-occurrence) are efficient, but recommend manual extraction for analyses requiring clustering.

Maestre et al. [23] investigate linguistic patterns in different narrative genres, such as news, reviews, and tales of children, aiming to identify variations in linguistic features based on their communicative purposes. Using computational tools, diverse corpora from these genres are analyzed for the presence and frequency of specific features, providing insights into the distinct characteristics defining each genre and commonalities within the narrative typology. The findings of the research findings can benefit NLP areas like computational narratology, enhancing understanding of genre-specific features and supporting the development of more robust automatic story generation tools and NLP tasks such as question answering and automated journalism.

While NLP-based methods have achieved significant success in narrative extraction, they are not without limitations. These methods heavily rely on the availability of annotated training data and domain-specific resources, which may be scarce or non-existent for certain domains or languages. Moreover, the complexity of narratives, with their inherent ambiguity and variability, poses challenges for NLP techniques to accurately capture and represent the intended meaning. Therefore, further research is needed to enhance the robustness and adaptability of NLP-based approaches to effectively extract narratives from diverse textual sources.

3.2 Semantic Graphs for Narrative Extraction

Semantic graphs have emerged as a promising framework for narrative extraction [5]. They provide structured representations of entities and their relationships, enhancing both local and global coherence in narratives.

For example, Blin [5] explores the role of narrative cognition and the development of computational systems for narrative representation. Their methodology comprises a multi-fold approach. Firstly, the author distinguishes between event-level and narrative-level analysis using a modified Simple Event Model, with a focus on `who`, `what`, `when`, `where` and `why` dimensions of narrative elements. Events relevant to a specific narrative, such as the "French Revolution", were collected from Wikidata and Wikipedia using carefully chosen graph paths and this data was enriched with pertinent information, including participants, locations, timestamps, and causal links.

Furthermore, Blin [5] outlined the construction of narrative networks. Using the collected data, a narrative graph was generated by converting the original triples from Wikidata and key-value pairs from Wikipedia into a structured narrative format. The resulting graph represents the relationships and attributes of events, participants, and locations within the narrative. Lastly, their research delves into the hypothesis generation within these narrative graphs, distinguishing between offline inference (graph completion) and online inference (next event prediction). Rules for converting data into graph form were manually designed and reasoning steps focused on enriching node types and semantics within the graph. Their study also considers the potential for machine learning to automate event collection from knowledge graphs. Their work holds promise for improving narrative understanding and reasoning.

Gottschalk and Demidova [13][14] introduced EventKG, a multilingual Resource Description Framework (RDF) knowledge graph containing over 690 thousand events and 2.3 million temporal relations from sources like Wikidata, DBpedia and Wikipedia in five languages. EventKG enhances event-centric information integration and provides interlinking, relation strength, and event popularity data. The authors emphasize the relevance of EventKG in Semantic Web, Question Answering, and timeline generation applications and discuss its data model, including provenance information and schema.

Althoff et al. [2] presented `TIME-MACHINE`, an approach to generate timelines for entities in knowledge graphs considering relevance, temporal, and content diversity. Using sub-modular optimization, it selects a subset of diverse and important events. The proposed interactive timelines enable detailed exploration and user studies to confirm its effectiveness compared to existing methods, making it a valuable tool for organizing and presenting complex entity histories.

3.3 Hybrid Methods

Hybrid methods for narrative extraction combine the strengths of NLP-based and graph-based approaches to improve the accuracy and adaptability of narrative extraction. These methods aim to leverage the complementary nature of these approaches to enhance overall performance.

For instance, Metilli [26] proposed "NarraNext", a semi-automatic tool designed to extract narrative elements from user-uploaded natural language text and facilitate the creation of complete narratives based on this extracted knowledge. NarraNext employs NLP techniques, including deep neural networks, to identify narrative elements within text. It is integrated with the Wikidata knowledge graph to import relevant knowledge. This tool serves as an evolution of the previous "NBVT" (Narrative Building and Visualization Tool) and aims to streamline the process of narrative extraction from text, making it faster and more efficient.

Tang et al. [38] introduced a novel approach called Multi-Tier Knowledge Projection Network (MKPNet) for event relation extraction in event-centric knowledge graphs (EventKGs) by leveraging discourse knowledge. EventKGs represent events and their relations, offering valuable insights for various natural language understanding tasks. While previous methods relied on explicit connectives for event relation extraction, MPLNet addresses the challenge of extracting implicit event relations. MKPNet consists of three adaptors: token, semantic, and coarse category adaptors, enabling the effective projection of discourse knowledge onto narratives. The neural architecture of MKPNet combines Bidirectional Encoder Representations from

Transformers (BERT)-based token encoding, a VAE-based semantic encoder, and coarse category prediction. This approach significantly improves event relation extraction performance and enriches EventKGs with both explicit and implicit event relations, benefiting downstream NLP tasks.

Despite the advantages of hybrid methods, there is still room for improvement. Further research is needed to explore advanced techniques for effectively combining NLP-based and graph-based approaches, as well as developing methods that can dynamically adapt the balance between these two approaches based on the characteristics of the data and the narrative structure.

3.4 Cross-Domain Applications

Narrative extraction has extended beyond textual data to encompass multimedia sources, such as images and videos. Researchers are developing cross-modal approaches that combine textual and visual information for a more comprehensive narrative understanding. Berhe [4] presented a thesis in which they address the challenge of automatically extracting narrative structures from TV series, particularly "Game of Thrones" and "Breaking Bad", due to the exponential growth of multimedia content. Their main contributions involve the development of a comprehensive approach comprising scene segmentation, scene linking, and most reportable scene detection, supported by pre-processing and visualization components. This approach helps reveal intricate narrative structures within TV series, bridging the gap between unstructured multimedia archives and meaningful content organization. The applications of this research span video indexing, multimedia collection reorganization, narrative summarization, and structured story generation, presenting valuable tools for content creators and researchers.

Lee and Jung [21] addressed challenges in analyzing narrative multimedia, such as movies and TV shows, which contain stories in various formats. Existing methods have limitations in handling diverse media types. Authors proposed character network-based approaches to analyze these narratives directly. They aim to synchronize and integrate textual, auditory, and visual data sources to improve character network accuracy. Their paper outlines methods for synchronizing scenes, characters, and character networks in narrative multimedia, particularly movies, to create a unified character network (UCharNet). Their proposed methods were evaluated using real-world movie data to assess their effectiveness.

3.5 Evaluation Methods and Datasets

Efforts to standardize evaluations in narrative extraction have led to the creation of benchmark datasets and evaluation metrics. These resources enable researchers to compare and assess the performance of different models and approaches.

Cyril et al. [6] discuss the evaluation of automatic story generation (ASG) systems, which aim to generate narratives from short prompts. They highlight the importance of assessing the quality of generated stories, as ASG has applications in gaming, communication, and education. The paper introduces a comprehensive set of human criteria for ASG evaluation, including relevance, coherence, empathy, surprise, engagement, and complexity. The authors also present the "HANNA" dataset, which contains human stories and stories generated by various ASG systems, all annotated by human raters based on the proposed criteria. A meta-evaluation of ASG systems is performed using this dataset, analyzing the correlations between various automatic metrics and the proposed human criteria.

Narrative evaluation presents significant importance in education, particularly focusing on narrative quality assessment. Somasundaran et al. [37] explore this topic by highlighting the challenges, the scarcity of scored essays in the narrative genre, and the need for manual annotation to develop scoring models. The paper also describes the data used for their research, which consists of narrative essays written by school students in response to various prompts. The essays are evaluated based on three main traits: Organization, Development, and Conventions, with sub-traits within each. The authors manually scored these essays to create a dataset for training and evaluating automated scoring systems.

Somasundaran et al. [36] also proposed a novel approach using graphs to evaluate the development of ideas and exemplification in essays. The authors constructed graphs from essays, where concepts became nodes and links represented concepts occurring in adjacent sentences. They used properties like "PageRank" to predict essay quality, including development. The paper presents experiments on two essay datasets: persuasive essays and narrative essays. The results show that graph-based features are useful for essay scoring and can improve the state of the art in essay scoring.

Ronanki et al. [32] explored the use of ChatGPT, a large language model, for automatic evaluation of user story quality. Their study compares the evaluation of ChatGPT with human evaluations and an existing tool (AQUSA) on criteria like well-formedness, atomicity, and minimality. Their results show moderate agreement between ChatGPT and human evaluators, with potential for improvement. The consistency of ChatGPT in evaluations suggests it can be a useful tool for this task, but careful interpretation and further refinement are necessary.

Since the goal of this thesis is to develop an algorithm capable of producing narratives in the form of semantic graphs, knowledge graph assessment tools could be relevant to our research. Huaman [18] introduced a framework for assessing the quality of knowledge graphs (KGs). KGs can be valuable but often suffer from issues like incomplete or inconsistent information. To address this, the paper proposes a customizable quality assessment framework for KGs, involving four main steps:

- **Identification**: This step involves identifying users, use cases, and the specific KGs to be assessed. Users (domain experts) guide the process and relevant use cases and KGs are selected.

- **Setting**: In this phase, the framework is set up. It includes defining the KG, selecting quality dimensions (QDs), and assigning weights to these dimensions based on their importance. QDs are aspects of quality and weights reflect their significance.
- **Assessment**: This step involves evaluating the KG based on the selected QDs. Quality Metrics (QMs) are used to measure the dimensions and these can be assessed manually, semi-automatically, or automatically. The aggregated scores for QDs are calculated and a total score for the KG is derived.
- **Exploitation**: The results of the assessment can be adjusted by fine-tuning the weights assigned to QDs and QMs. Users can compare the quality of different KGs for specific use cases and access the results through a user interface.

The paper identifies 20 QDs applicable to KGs, including accessibility, accuracy, completeness, and others. These dimensions are used to assess KGs and the framework allows customization based on specific needs. The approach of this paper is rooted in the Goal Question Metric (GQM) approach, which involves defining goals, questions to achieve those goals, and metrics to answer those questions. It classifies metrics as quantitatively measured (QN) or qualitatively measured (QL). The framework aims to help users make informed decisions about which KGs to use for their specific tasks, considering the quality dimensions that matter most to them. It offers flexibility and adaptability to various domains and use cases.

3.6 Challenges and Future Directions

Overall, the existing literature demonstrates the progress made in narrative extraction from various types of textual data. However, there is still a need for further research to overcome the limitations of current methods. Specifically, there is a lack of approaches that effectively leverage the structured information available in semantic graphs to extract narratives.

In summary, the state of the art in narrative extraction showcases a convergence of traditional techniques, NLP advancements, semantic graph frameworks, hybrid methods, deep learning models, and cross-domain applications. Researchers continue to push boundaries, working towards a more accurate, coherent, and interpretable narrative extraction system with broader applications in information retrieval and knowledge management.

In the next section, we present our proposed approach for narrative graph extraction, which fills this gap by combining string matching and rule-based methods to capture both semantic and structural information in narratives.

Chapter 4

Design and Development

This chapter delves into the heart of our research, presenting a detailed and extensive description of the methodological approach employed in the extraction of narratives from semantic graphs, as well as trying to answer research questions **Q1**, **Q2** and **Q3**, described in Chapter 1.

Our methodology begins by trying to answer the first question (**Q1**) in section 4.1. For that, we emphasize the significance of maintaining consistency in describing narratives and events to ensure clear and meaningful analysis. To achieve this, we have devised an ontology tailored specifically for representing narrative information, which we defined as a sequence of interconnected events.

As part of question **Q2**, we introduce a graph configuration system in section 4.2 into our approach to help identify relevant information within semantic graphs by allowing users to manually configure and adjust graphs to suit their specific needs. This enables us to select information according to user-defined criteria, making our approach adaptable to different graphs and narrative types as a consequence.

Our main approach to narrative extraction, to which we refer as Narrative Extractor from Semantic Graphs (NESG), was divided into two parts: information extraction and narrative building, which can be found in sections 4.3 and 4.4, respectively. Information extraction completes the answer to the research question **Q2**, by specifying the detailed methodology of information extraction used by our approach. We employed graph searching techniques, such as **bfs!** (**bfs!**), and a set of rules to extract and identify relevant information to the selected narrative. This information is then processed through triple data parsing and entity classification techniques. On the other hand, narrative building explores research question **Q3**, by providing a narrative construction workflow. This workflow includes event property mapping and narrative construction through event linking, using the previously extracted data.

To summarize, our methodology consists of the following four phases:

- Event and Ontology Definition. (**Q1**)

- Graph Configuration. (**Q2**)
- Information Extraction. (**Q2**)
- Narrative Building. (**Q3**)

As was found out during our work, research question **Q4** is a non-trivial question and poses significant challenges that require their own, dedicated, attention. This question is explored further in chapter 6.

In relation to the existing work in this specific field, our contribution consists of expanding the concept of events by removing constraints such as narrative genres and event classes, allowing for more flexible narratives, as well as providing a web interface for deeper levels of interactability and, consequently, higher generalizability. Furthermore, we allow users to manually configure graphs through the user interface, as well as specify numerous parameters for the narrative extraction procedure. The results are tied directly to those parameters, including graph configuration. In the following sections, we explore each step of our workflow in detail.

4.1 Narrative Definition and Ontology Specification

To lay the foundation of our exploration and to help answer the research question **Q1**, we begin with the definition of narratives. In this study, narratives are defined as ordered sequences of events. This concise definition provides a suitable starting point for our investigation, recognizing that narratives are complex entities with multifaceted characteristics.

Narratives, as ordered sequences of events, imply that they have a structured temporal dimension. Events, on the other hand, are occurrences linked to specific points in time. The relationship between narratives and events is central to understanding how narratives are constructed and conveyed. A narrative unfolds a series of events that collectively shape the narrative's meaning, structure, and impact.

4.1.1 The 5W1H Method

To describe events within narratives, we employ the 5W1H method, as suggested by Blin [5]. This method involves answering six key questions:

- What happened? (What)
- Where did it happen? (Where)
- Who was involved? (Who)
- When did it happen? (When)

- Why did it happen? (Why)
- How did it happen? (How)

These questions serve as a structured framework for capturing the essential details of an event. By answering these questions, we create a comprehensive and nuanced description of the event, ensuring that no crucial aspect is overlooked. This structured approach not only aids in conveying events effectively but also facilitates the organization and analysis of narrative data.

4.1.2 Ontology

Consistency in the description of narratives and events is paramount to ensure clarity and meaningful analysis. To achieve this, we have developed a web ontology[1] tailored to the representation of narrative information. This ontology is a structured framework that defines the relationships and properties of events and narratives within the semantic graph framework.

Our ontology comprises seven main classes, that follow the hierarchy depicted in Figure 4.1, which contains classes and heritage links between them (if a class points to another class, then it is a child of that class). It uses the "nrtv" prefix to denote our namespace. The following subsections delve further into the nuances of each class.

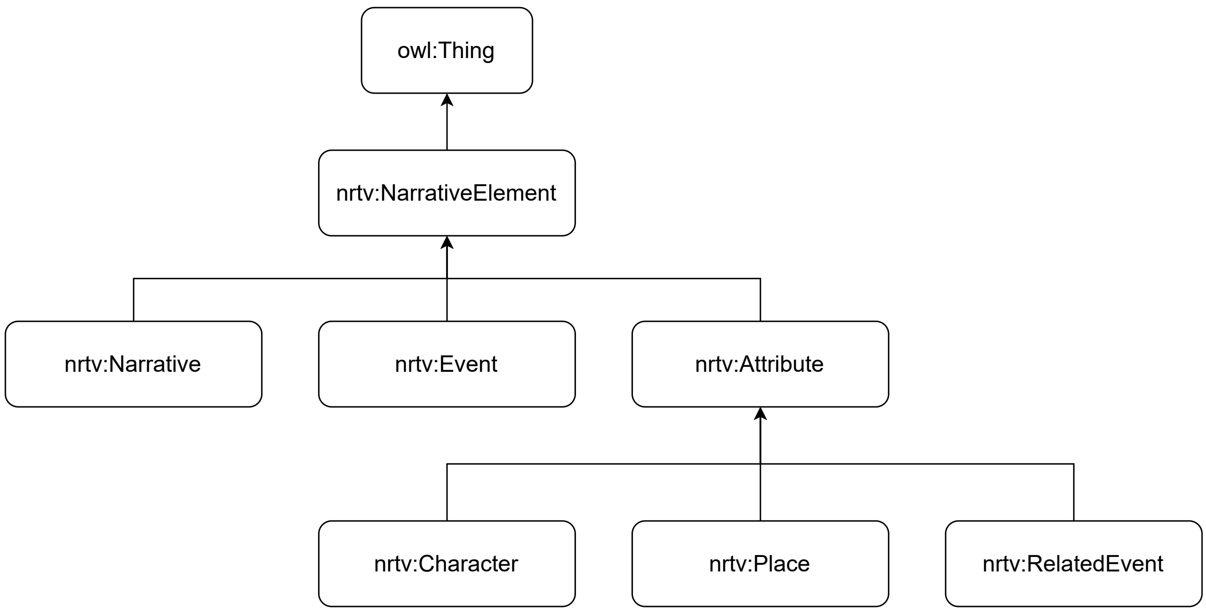


Figure 4.1: NRTV Ontology Class Hierarchy.

[1] nrtv ontology. https://purl.archive.org/purl/nrtv/v1/ontology.ttl

4.1.2.1 The NarrativeElement Class

The NarrativeElement class is the root class of our ontology. It established a domain to contain all of the classes within our ontology. When it comes to narratives, this class does not represent anything specific, but rather it provides a structural component to group and unite all of the other classes together. It has three child classes that make up the main building blocks for representing narratives within our ontology: Narrative, Event, and Attribute.

4.1.2.2 The Narrative Class

The narrative class in our ontology represents the overarching structure that binds events together to form a coherent and meaningful narrative. To describe a narrative, we employ the following properties:

- **nrtv:About**: Defines the topic that the narrative is supposed to convey. In technical terms, it is the topic that the user provided for that instance of narrative extraction. Thus, the range of this property is a literal. When applied to merged narratives, the respective topics are concatenated by an "and" into a single string.
- **nrtv:hasTheme**: Links all the classes that the main entity (the topic entity) derives from. Since our algorithm of narrative extraction revolves around a single entity within the graph, for each narrative, its classes denote the overall theme of the narrative, just as they denote the meaning of the respective entity.
- **nrtv:eventSequence**: The fundamental property that defines the body of a narrative. Points to an instance of rdf:Seq, which uses rdf:_# property to group all linked events in chronological order.

4.1.2.3 The Event Class

The event class within our ontology serves as the cornerstone for representing individual occurrences within a narrative. It encompasses six key properties, each corresponding to the 5W1H categories (What, Where, Who, When, Why and How). These properties enable a precise and detailed description of events based on their characteristics. Additionally, the event class includes a property for establishing relationships between events and other entities that may not fit into any of the specific attributes but are nevertheless fundamental for maintaining narrative flow and coherence. The domain and range of each property can be found in Table 4.1.

- **nrtv:what (Event Description)**: This property captures the essence of the event, providing a concise and informative description of what transpired. It serves as the core element in defining the event's content, thus it cannot be omitted.

- **nrtv:where (Event Location)**: To contextualize the event, this property specifies the geographical, spatial, or conceptual location where the event occurred. It adds a spatial dimension to the event's description.

- **nrtv:who (Event Participants)**: This property identifies the individuals, groups, or entities involved in the event. It helps establish the key actors and stakeholders in the narrative. This property is not limited, however, to individual people. It can also represent, in certain situations, groups of people, organizations, or whole nations.

- **nrtv:when (Event Timestamp)**: To anchor the event in time, this property records the specific point in time when the event occurred. It provides a temporal dimension to the event. Some events can have multiple timestamps, such as start time and end time. In this case, the event is split into two, one representing the beginning and the other the end.

- **nrtv:why (Event Reason)**: Gives the reason for the event to occur. This is rarely used given that semantic graphs, typically, do not contain information of this nature.

- **nrtv:how (Event Manner)**: Explains how the event occurred. Just as the "Why" property, it is rarely used due to not being often present within the original graph. Can be used, for example, to explain the manner of death of a character.

- **nrtv:relatedTo (Event Links)**: This property enables the establishment of connections between events and other entities, such as themes, motifs, or overarching narrative structures. It ensures that events are not isolated but interconnected within the narrative context.

Table 4.1: Event property domain and range.

	rdfs:domain	**rdfs:range**
nrtv:what	nrtv:Event	xsd:string
nrtv:where	nrtv:Event	nrtv:Place
nrtv:who	nrtv:Event	nrtv:Character
nrtv:when	nrtv:Event	xsd:string
nrtv:why	nrtv:Event	xsd:string
nrtv:how	nrtv:Event	xsd:string
nrtv:relatedTo	nrtv:Event	nrtv:RelatedEvent

4.1.2.4 The Attribute Class

Just like the NarrativeElement class, the Attribute class serves as a structural point to connect all classes that represent event attributes. Furthermore, it provides a property, for each one of its subclasses, "nrtv:subjectOf" that links the entity in the original graph to the event attribute in question. This class originates three child classes: Character, Place, and RelatedEvent.

4.1.2.5 The Character Class

Characters are, perhaps, the main plot drive of the narratives. Most of the narratives revolve around characters and their actions, bringing light to the importance of these when constructing a narrative. The roles of the characters are not static and can evolve along the narrative, from one event to another. To help depict that in our ontology, we created the character class that serves to provide further insight into the character's role in specific events, by utilizing the "nrtv:hasRole" property.

4.1.2.6 The Place Class

Locations play a pivotal role in describing narratives by localizing them in space. However, additional context may be necessary to provide a clearer context for each mentioned location. For that, we use the "nrtv:hasSetting" property that specifies the conditions (the property in the graph that links this location to the event) under which the location is mentioned within the event.

4.1.2.7 The RelatedEvent Class

The last class when it comes to the event attributes. The RelatedEvent class, just like other attribute classes, serves to provide additional context for each pair of linked events. To do that, it uses the property "nrtv:hasRelation" to specify the relation that links the two events.

4.1.3 Example Narrative

To illustrate the practical application of our ontology, let us consider an example narrative related to Abraham Lincoln. The narrative can be found, in Turtle format, in Listing 4.1.

```
narrative_1 a nrtv:Narrative;
    nrtv:about "Abraham Lincoln";
    nrtv:hasTheme http://www.wikidata.org/entity/Q5;
    nrtv:eventSequence event_sequence_1.

event_sequence_1 a rdf:Seq;
    rdf:_1 event_44;
    rdf:_2 event_45;

event_44 a nrtv:Event;
    nrtv:what "date of birth";
    nrtv:when "1809-02-12T00:00:00Z";
    nrtv:who nrtv:character_59;
```

```
    nrtv:where nrtv:location_22;
    nrtv:where nrtv:location_23;
    nrtv:where nrtv:location_24;
    nrtv:where nrtv:location_25;

event_45 a nrtv:Event;
    nrtv:what "military branch start time";
    nrtv:when "1832-04-21T00:00:00Z";
    nrtv:who nrtv:character_60;
    nrtv:who nrtv:character_61.

character_59 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q91;
    nrtv:hasRole perpetrator.

character_60 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q91;
    nrtv:hasRole perpetrator.

character_61 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q5999581;
    nrtv:hasRole http://www.wikidata.org/prop/P241.

location_22 a nrtv:Place;
    nrtv:subjectOf http://www.wikidata.org/entity/Q113363481;
    nrtv:setting http://www.wikidata.org/prop/direct/P19.

location_23 a nrtv:Place;
    nrtv:subjectOf http://www.wikidata.org/entity/Q1603;
    nrtv:setting http://www.wikidata.org/prop/P19.

location_24 a nrtv:Place;
    nrtv:subjectOf http://www.wikidata.org/entity/Q498128;
    nrtv:setting http://www.wikidata.org/prop/P19.

location_25 a nrtv:Place;
    nrtv:subjectOf http://www.wikidata.org/entity/Q547313;
    nrtv:setting http://www.wikidata.org/prop/P19.
```

Listing 4.1: Abraham Lincoln Narrative

In this example, we can see the main entity representing the narrative (narrative_1). This entity is linked to an rdf:Seq instance, which acts as a list of all connected events, in chronological order. In this particular example, we only have two events (event_43 and event_45), which are linked to different characters and places. Those characters and places are defined by their own entities, which consist of the character/location in question and their respective role/setting in the linked event.

The ontology-based approach we proposed ensures consistency in the description of narrative events. By defining events and narratives within an ontology, we establish a standardized framework for capturing and representing narrative information. This approach enables the extraction of narrative graphs from semantic graphs, facilitating the analysis, retrieval, and generation of narratives.

One of the primary benefits of using an ontology is the consistent and structured representation of events and narratives. Researchers and communicators can adhere to a predefined set of properties and relationships, ensuring that narrative information is conveyed uniformly. This consistency is crucial for conveying complex narratives effectively, particularly in contexts where multiple individuals or systems contribute to narrative construction.

The ontology-based representation of narratives and events also greatly facilitates analysis. Researchers can use the structured data to conduct in-depth analyses, such as sentiment analysis, event clustering, or trend identification. Additionally, the ontology allows for easy integration with data analytics and visualization tools, enabling the exploration of narrative data from various perspectives.

Moreover, the ontology-based approach supports the retrieval and generation of narratives. Search engines and recommendation systems can utilize structured data to provide users with relevant narratives based on their interests and preferences. Additionally, automated narrative generation systems can use ontology to assemble coherent and contextually relevant stories from a pool of events and information.

4.2 Narrative and Graph Configuration

Semantic graphs have revolutionized the representation of knowledge and information, but their universal processing poses significant challenges. The core issues stem from the variability in graph structure, schema, and the subjective nature of narratives. These challenges hinder the creation of a unified framework for semantic graph processing. In response, and to partially answer the research question **Q2**, our approach adopts a manual graph configuration strategy, empowering users to tailor the algorithm to their specific needs. In this comprehensive exploration, we elucidate the struggles encountered in working with semantic graphs and present our solution through detailed examples.

4.2.1 Challenges of Working with Semantic Graphs

4.2.1.1 Structural Inconsistencies

Despite there not being any rules to how a semantic graph should be structured, it is expected that all graphs should follow, approximately, the same principles and structure. However, some semantic graphs from various domains exhibit stark structural inconsistencies. One of the most notable examples is Wikidata. Unlike typical semantic graphs, Wikidata attempts to convey more information through, what they call, the "qualifiers". These "qualifiers" are, essentially, extensions of graph triples, containing more detailed information relative to those triples. To achieve this, it creates statement nodes that do not hold any particular information themselves, but rather link to all relevant qualifiers. This is problematic when it is unknown whether we are dealing with a statement node or a resource node. To deduct whether a node is a statement node or not, we would need to know the ontology class that represents statement nodes. An example of this issue can be visualized in Listing 4.2, representing normal semantic graph structure, and Listing 4.3, representing Wikidata's structure.

```
@prefix ex: <http://example.org/> .

ex:John ex:name "John Smith".
ex:John ex:date_of_birth "15 April 1979".
ex:John ex:spouse ex:Jane.
```

Listing 4.2: Normal Semantic Graph Structure

```
@prefix ex: <http://example.org/> .

ex:John ex:name "John Smith".

ex:John ex:date_of_birth ex:Statement1.
ex:Statement1 ex:date_of_birth "15 April 1979".
ex:Statement1 ex:place_of_birth ex:Canada.

ex:John ex:spouse ex:Statement2.
ex:Statement2 ex:spouse ex:Jane.
ex:Statement2 ex:place_of_mariage ex:France.
ex:Statement2 ex:end_cause "Death of subject".
```

Listing 4.3: Wikidata Graph Structure

In this example, we can see the clear differences between Wikidata and a regular semantic graph and how that can pose challenges when it comes to extracting data from an unknown source. However, the extra data that can be found in Wikidata can be extremely valuable for generating narratives, once the problem has been overcome.

Our approach deals with this issue by assuming that a node is a statement node if all of the following conditions are satisfied:

- The node does NOT extend any class.
- The node does NOT contain a label.

If, at least, one of these conditions fails, it is assumed that the node represents a real entity. This is not 100% reliable and can have rare exceptions. However, from our experiments, we found out that most of the real entity nodes either have a label or extend a class, meaning that they are not considered statements.

Moreover, this statement format in Wikidata creates additional problems when it comes to parsing data. Due to its nature, Wikidata contains both statement and non-statement variants for every triple in the graph. Continuing the previous example, the following graph representation can be found in Wikidata:

```
@prefix ex: <http://example.org/> .

ex:John ex:name "John Smith".
ex:John ex:name ex:Statement1.
ex:Statement1 ex:name "John Smith".

ex:John ex:date_of_birth "15 April 1979".
ex:John ex:date_of_birth ex:Statement2.
ex:Statement2 ex:date_of_birth "15 April 1979".
ex:Statement2 ex:place_of_birth ex:Canada.
```

Listing 4.4: Wikidata Graph Structure

In this example, we can witness how triples `ex:John ex:name` "John Smith"." and `ex:John ex:date_of_birth` "15 April 1979"." are present in their basic form, as well as in their statement form. When it comes to processing this data, we need to develop a mechanism capable of, reliably, filtering these triples to avoid duplicates, while maximizing the amount of information extracted.

When a statement node is detected, our approach replaces it with all of its Resource Description Framework (RDF) triples, substituting the statement node for the original subject entity. The modified version of the RDF graph in Listing ?? using this mechanism can be seen in Listing 4.5.

```
@prefix ex: <http://example.org/> .

ex:John ex:name "John Smith".
ex:John ex:name "John Smith". %created a duplicate triple as a
   defect

ex:John ex:date_of_birth "15 April 1979".
ex:John ex:date_of_birth "15 April 1979". %another duplicate
   triple
ex:John ex:place_of_birth ex:Canada. %converted triple
```

Listing 4.5: Statement Node Substitution

This has been just one example of possible structural inconsistencies that can be found inside semantic graphs.

4.2.1.2 Schema and Ontology Variations

Moreover, variations in schema and ontologies further compound the complexity of the problem. For instance, consider the two triples that come from different graphs in Listings 4.6 and 4.7.

```
ex:Paper1 ex:hasAuthor ex:Author1.
```

Listing 4.6: Has author property variant 1

```
ex:Paper1 ex:has_author ex:Author1.
```

Listing 4.7: Has author property variant 2

Both triples dictate that ex:Paper1 has author `ex:Author1`. However, the first triple uses `ex:hasAuthor` and the other uses `ex:has_author` as the property of the triple. Despite the meaning of both triples being the same, there is no trivial way of deducting that both properties convey the same information. This is especially problematic during the narrative generation process since we need to know how entities are related one to another.

4.2.1.3 Inferring Ontology Classes and Predicates

Lastly, inferring the meaning of ontology classes and predicates becomes challenging without a clear understanding of the specific ontology being used. The same predicate name may be used differently in different ontologies, making it necessary to consider ontology-specific semantics. Consider the previous example:

```
ex:Paper1 ex:has_author ex:Author1.
```

We, as humans, know that the property `has_author` must point to a person who is the author of the subject resource. However, for a computer to understand that, it must be, specifically, told to do so, as in it must be programmed to assume that the property `has_author` refers to a person. Ontologies do provide methods for inferring the meaning of entities through properties such as `rdf:domain` and `rdf:range`, or through specific class instances. The problem is that we cannot rely on the existence of this data, since it is not always present. Furthermore, some graphs, such as Wikidata, use their own, specialized and unknown, vocabularies, hindering data extraction and processing.

4.2.2 Our Solution: Manual Graph Configuration

Designing an algorithm capable of working with semantic graphs, regardless of its structure or domain, is a formidable task. To overcome these challenges, we propose a practical solution: a configurable framework that empowers users to provide specific configuration parameters for their graphs.

On of the main features of this approach is that the graph configurations are saved in a JSON file and can be accessed at any point, meaning that each graph has to be configured only once and can be modified at any time.

Configuration parameters are at the core of our approach. These parameters allow users to customize the algorithm based on the characteristics of their graphs, thus achieving adaptability and flexibility in extracting narratives. Next, we will delve into the crucial configuration parameters that define graph and narrative behavior. We separated configuration parameters into two categories: basic and class-related.

4.2.2.1 Basic Parameters

Basic parameters are essential to graph definition and configuration. Without them, the graphs will be inoperable and the program will not proceed. The basic parameters include:

- **Endpoint**: The SPARQL URL endpoint through which the data extraction is done.
- **rdf:type replacer**: Property that replaces the commonly used "rdf:type", if any. If left empty, it is assumed that the graph uses "rdf:type" to establish connections between entities and their classes.
- **rdf:subClassOf replacer**: Property that replaces `rdf:subClassOf`, which is used to establish hierarchical relationships between classes. If left empty, `rdf:subClassOf` will be used instead.

- **schema:description replacer**: Property that replaces `schema:description`, used to transmit more detailed information about entities to the end user.

The substitution parameters enhance compatibility with diverse graph structures and ensure that the algorithm can effectively identify and extract narrative components.

4.2.2.2 Class Parameters

Class parameters play a crucial role in narrative generation logic. They define classes for each of the following concepts:

- **Person/Group**: Contains classes that represent people or groups of people.
- **Location**: Contains classes that represent locations, both geographical and fictional.
- **Event**: Contains classes that represent events and occurrences.
- **Reason**: Contains classes that represent reason.
- **Manner**: Contains classes that represent manner.

As we can see, these concepts correspond to some of the 5W1H attributes. These classes will be used to classify each extracted entity into one of the five concepts. The goal is to include broader classes, rather than more specific ones since the classification process will traverse the entity's class hierarchy until it finds a match. How an entity will be used for the narrative will be based on the attributed concept.

Additionally, we ask users to provide any optional types that should be ignored for the relevant information-gathering step of our process.

This approach allows us to, essentially, give meaning to entities that a computer can understand, in an efficient and elegant manner.

As an example, here is the configuration that we used for the Wikidata graph:

- **Endpoint**: "https://query.wikidata.org/sparql".
- **rdf:type replacer**: `wdt:P31"`. It is a property in Wikidata labeled as "instance of", used to specifty "the class of which this sucject is a particular example and member".
- **rdf:subClassOf replacer**: `wdt:P279`. It is a property in Wikidata labeled as "subclass of", used to specify that "this item is a subclass (subset) of that item".
- **Excluded types**: `wd:Q19847637`. It is a property in Wikidata labeled as "Wikidata property for an identifier", used to specify "something in an external source".

- **Person/Group classes**: `wd:Q5"`, `wd:Q43229`. They are properties in Wikidata labeled as "human" and "organization", respectively.
- **Location classes**: `wd:Q58416391`. It is a property in Wikidata labeled as "spatial entity", used to specify "thing occupying some space".
- **Event classes**: `wd:Q1190554"`. It is a property in Wikidata labeled as "occurrence", used to specify "occurrence of a fact or object in space-time".
- **Reason classes**: None
- **Manner classes**: None

4.2.3 Narrative Configuration Parameters

4.2.3.1 Depth Parameter

Narratives, being subjective in nature, require user input to determine their appearance and behavior. The "depth" parameter is a primary narrative configuration parameter that controls the level of depth or expansion of the narrative. Given the interconnectivity of events, narratives can potentially span infinitely, branching out into various sub-events. By specifying the depth parameter, users can control the extent to which the narrative expands, allowing for a more controlled and focused representation of the story.

Suppose we have a narrative about a historical event with multiple sub-events:

- Depth set to 1: The narrative includes the main event (if applicable) and its immediate sub-events.
- Depth set to 2: The narrative encompasses the main event (if applicable), its sub-events, and their respective sub-events, creating a more detailed narrative.
- Depth set to 3: The narrative focuses on the main event (if applicable), its sub-events, their respective sub-events, and sub-events of those.

The further down we go along the event tree, the less relevant they start to become in relation to the topic, hence why it is important to select an appropriate depth value for the narrative.

4.2.3.2 Size Parameter

In addition to the depth parameter, the "size" parameter is critical in narrative configuration. Some graphs, such as Wikidata, can be exceptionally large and contain an overwhelming amount of information. Handling such extensive graphs in their entirety might not be feasible or efficient for the algorithm. Therefore, it is necessary to limit the amount of information retrieved from the graph to avoid long execution times or event timeouts. The size parameter allows users to specify the desired scope of information to be retrieved from the graph, striking a balance between comprehensiveness and computational efficiency. In technical terms, "size" is responsible for limiting the number of incoming links for each entity, of which there can be many.

4.2.3.3 Time Range

Finally, users can specify the start date and the end date for each narrative to restrict the results to a specific period. Only events that take place within this time period will be considered for narrative building.

4.2.4 Technical Parameters

Technical parameters play a crucial role in the intricate parts of our algorithm. They affect different stages of our approach.

4.2.4.1 Similarity Threshold

The similarity threshold is used as the reference point for comparison between property labels, using Ratcliff-Obershelp similarity. It is fundamental for triple clustering in the attribute mapping section.

4.2.4.2 In-Degree Centrality

In-degree centrality measures the number of incoming links for a node in a graph. It is used in our work to restrict entities that are considered too broad or popular. Those entities will not be used for the data extraction part of our approach. For instance, a person who was born in Poland has a triple representing this relationship. The triple itself could be relevant for the narrative. However, the entity of Poland should have a higher in-degree centrality than other, more specific, entities. Therefore, we assume that Poland, and its events and narratives, are not relevant to the current narrative, so it will not be expanded during the data extraction step, but the entity itself may still be used to provide information.

4.2.5 Classification Parameters

Classification parameters set up the classification system of our approach. The user can decide what type of classification they want to be performed: manual, automatic (all), or automatic (new entities). Manual classification forces the user to manually assign classes to entities. Automatic classification allows the algorithm, described in the classification section, to assign classes automatically to the extracted entities, based on the graph configuration parameters provided. The automatic classification can either involve all entities or only the ones that appear for the first time and have not been classified yet. The assigned classes are stored, regardless of the classification type picked, in data storage for further re-use.

When using automatic classification, entities can be assigned multiple classes. To disambiguate that, users can either do it manually or provide rules for automatic disambiguation. The disambiguation rules give priority to certain classes in the presence of the others. For instance, when an entity is assigned a class "Person" and a class "Location", users can give priority to the "Person" class in this case and the algorithm will, automatically, pick this class whenever confronted with the aforementioned conflict.

By incorporating these graph and narrative configuration parameters, our approach provides a customizable framework that adapts to different graph structures, narrative preferences, and computational limitations. The user's input and configuration choices play a crucial role in tailoring the algorithm to suit specific requirements and ensure the extraction of meaningful and relevant narratives.

4.3 Information Extraction

The information extraction step involves extracting, parsing, and analyzing triples extracted from the source semantic graph. It comprises relevant entity extraction, main entity identification, and entity classification steps, which are described, in detail, below.

4.3.1 Relevant Entity Extraction

Semantic graphs are rich sources of information, but extracting meaningful narratives from them requires a systematic and adaptable approach. Our algorithm aims to address this challenge by first identifying the main entity representing the user's input topic. Subsequently, it extracts relevant entities connected to the main entity using **bfs!**. The depth parameter in the **bfs!** algorithm plays a critical role in determining the depth of the narrative space. In the following sub-sections, we thoroughly explore this step of our algorithm, emphasizing the importance of relevant entities in narrative building.

4.3.1.1 Main Entity Identification

The initial step of our algorithm involves the identification of the main entity that represents the user's input topic. The user must supply a string as the topic and the algorithm queries the semantic graph to retrieve all entities labeled with that particular string. It is important to note that multiple entities may be associated with the input topic, as labels can be shared among different entities. Therefore, the user must specify which entity they want the narrative to focus on. This selected entity becomes the main entity, serving as the central point for event extraction.

Suppose the user provides the input topic "Mona Lisa". The algorithm queries the semantic graph and retrieves entities labeled as "Mona Lisa". Multiple entities are found, but the user selects the entity representing the famous painting as the main entity.

```
@prefix ex: <http://example.org/>.

ex:MonaLisa1 rdf:type ex:Artwork.
ex:MonaLisa2 rdf:type ex:Person.
ex:MonaLisa3 rdf:type ex:Location.
```

The query depicted in Listing 4.8 is used to identify and extract all entities, as well as all relevant associated data, such as respective descriptions and classes, whose label is equivalent to the provided topic string.

```
SELECT
?s
(SAMPLE(?d) as ?d)
((GROUP_CONCAT(DISTINCT ?tl; separator=", ")) as ?t)
(SAMPLE(?topic) as ?topic)
WHERE {
    VALUES ?topic {${topic_string}}
    ?s rdfs:label ?topic.
    ?s schema:description ?d.
    ?s ${graph_data.type} ?type.
    ?type rdfs:label ?tl.
    FILTER (lang(?d) = "en" && lang(?tl) = "en")
}
GROUP BY ?s
```

Listing 4.8: SPARQL Main Entity Extraction Query

In this query, since we are aggregating the results based on the entity identifier, we need to use appropriate aggregation methods for the rest of the properties. This is due to the fact that some entities can belong to multiple classes or, in rare cases, have multiple descriptions. We use the **SAMPLE** clause on the description property since it is not as relevant as other properties and one description variant should suffice. On the other hand, we need to know all the classes that the entity derives from, which is why is use the **GROUP_CONCAT** aggregator to compile all classes into a single list. Finally, the topic variable can be safely sampled as well, since all corresponding entities will have the same label.

Another important feature of this query to highlight is the use of the **VALUES** clause to specify multiple values for the topic variable. This is done in case the user decides to provide multiple topics for narrative extraction.

Lastly, all entries in the query that start with "$", such as `$topic_string`, define placeholder variables that will be replaced by the actual values during the run-time of the algorithm.

Once the main entity is determined, it serves as the central focal point for the event extraction. All narrative components will revolve around this main entity, ensuring that the narrative remains coherent and contextually relevant to the user's input topic. The main entity provides a structured starting point for exploring the semantic graph and extracting interconnected events and relationships.

4.3.1.2 Relevant Entity Extraction

The subsequent step in our algorithm involves the extraction of relevant entities connected to the main entity. These relevant entities form the narrative space and contribute significantly to the narrative's coherence and context. Extracting relevant entities ensures that the narrative is not limited to a single entity but encompasses a broader spectrum of interconnected events and relationships.

Definition 4.3.1 (Relevant Entity)**.** Entity that is connected, in some form, to the main entity, be that directly or recursively. Given the main entity X and two entities Y and Z, such that Y is directly linked to X and Z is directly linked to Y, both Z and Y are considered as relevant entities. This property applies to all depth levels.

To extract relevant entities, our algorithm employs the Breadth-first search (BFS) algorithm. BFS is a graph traversal technique that systematically explores the graph, starting from the main entity, and retrieves connected entities within a specified depth. The BFS algorithm ensures that entities closest to the main entity are extracted first, gradually expanding the search to entities further away in the graph.

Traditionally, semantic graphs are considered unidirectional graphs. However, for the relevant entity extraction process, we acknowledge the links of the graph as bidirectional.

Consider a simplified semantic graph representing a book and its related entities in Listing 4.9.

```
@prefix ex: <http://example.org/>.

ex:Book1 rdf:type ex:Book.
ex:Author1 rdf:type ex:Author.
ex:Genre1 rdf:type ex:Genre.
ex:Store1 rdf:type ex:Store.

ex:Book1 ex:hasAuthor ex:Author1.
ex:Book1 ex:belongsToGenre ex:Genre1.
ex:Store1 ex:sells ex:Book1
```

Listing 4.9: Book RDF Graph

If "ex:Book1" is the main entity, the BFS algorithm would first extract "ex:Author1", "ex:Genre1" and "ex:Store1" as relevant entities within a depth of one.

The depth of the BFS search is controlled by the parameter aptly named "depth". This parameter indicates the distance or level of separation from the main entity. A higher depth value results in a more extensive exploration of the graph, potentially capturing more specific or seemingly unrelated events in relation to the main entity.

Consider a scenario where the user is interested in a comprehensive narrative for the "World War II" topic. In Figure 4.2 we can see the main entity (marked with red) and all extracted relevant entities (marked with green) for the depth set to 1. On the other hand, in Figure 4.3 the same extraction process was performed, except for the depth set to 2, with relevant entities marked with green and yellow. We can witness a clear difference in the level of relevance between relevant entities for depth 1 and relevant entities for depth 2.

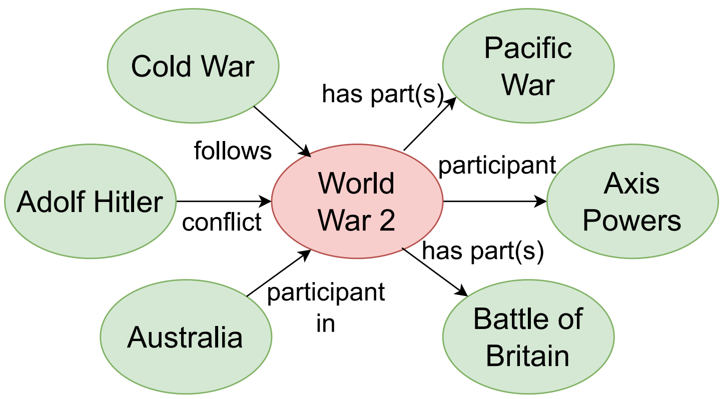


Figure 4.2: "World War II" Relevant Entities for Depth 1.

By configuring the depth parameter, users can tailor the level of detail and breadth of the narrative to match their specific needs.

The relevant entity extraction process was split into two **sparql!** (**sparql!**) queries: one

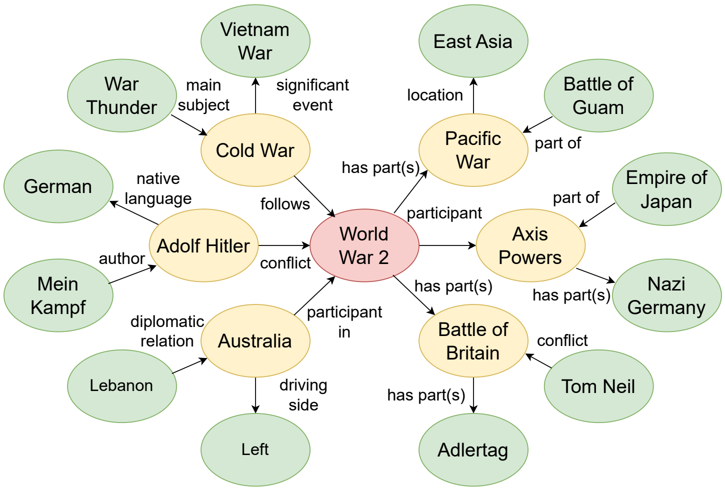


Figure 4.3: "World War II" Relevant Entities for Depth 2.

for the incoming links and the other for the outgoing link. The reason for this separation is, mainly, due to how statement nodes are processed. Both queries, as well as most of the other queries in this work, follow a modular design. They consist of three parts: head, body, and foot. The head, usually, contains the `SELECT` clause for defining the final output variables, as well as variable values definition using the `VALUES` clause. An example head of a query can be seen in Listing 4.10. The body of a query contains the main entity extraction pattern, that can be replicated multiple times with the `UNION` clause. An example of a body of a query can be found in Listing 4.11. Each body segment, typically, is represented as a nested query. The foot contains any additional processing and parsing of the entities extracted by the body segment, such as `OPTIONAL` or `FILTER` clauses. An example query foot is provided in Listing 4.12.

```
SELECT ?s ?l ?d ?t ?p ?o WHERE {
```

Listing 4.10: SPARQL Query Head Example

```
{
    SELECT DISTINCT ?s ?p ?o WHERE {
        VALUES ?excl {<wd:Q19847637> <wd:Q19847637>}
        VALUES ?o {<http://www.wikidata.org/entity/Q7186>}
        ?s ?p ?o .
        FILTER ISIRI(?s)
        FILTER (?p NOT IN (<wd:Q19847637>, <wd:Q19847637>))
        FILTER ( NOT EXISTS {?p wdt:P31/wdt:P279* ?excl})
    }
    LIMIT 5
}
```

Listing 4.11: SPARQL Query Body Example

```
    OPTIONAL {
        ?s rdfs:label ?l.
        FILTER (lang(?l) = "en")
    }
    OPTIONAL {
        ?s ${narrative_data.graph_data.description} ?d.
        FILTER (lang(?d) = "en")
    }
    FILTER(BOUND(?d) || BOUND(?l))
}
```

Listing 4.12: SPARQL Query Foot Example

The **sparql!** query responsible for extracting information about incoming links can be found in Listing 4.13 and the **sparql!** query that extracts all relevant information regarding outgoing links is depicted in Listing 4.14.

```
SELECT ?s ?l ?d ?p ?o WHERE {
    {
        SELECT DISTINCT ?s ?p ?o WHERE {
            VALUES ?excl {<wd:Q19847637> <wd:Q19847637>}
            VALUES ?o {<http://www.wikidata.org/entity/Q7186>}
            ?s ?p ?o .
            FILTER ISIRI(?s)
            FILTER (?p NOT IN (<wd:Q19847637>, <wd:Q19847637>))
            FILTER ( NOT EXISTS {?p wdt:P31/wdt:P279* ?excl})
        }
        LIMIT 5
    }
    OPTIONAL {
        ?s rdfs:label ?l.
        FILTER (lang(?l) = "en")
    }
    OPTIONAL {
        ?s schema:description ?d.
        FILTER (lang(?d) = "en")
    }
    FILTER(BOUND(?d) || BOUND(?l))
}
```

Listing 4.13: SPARQL Query for Incoming Link Extraction

```
SELECT ?s ?p ?o ?ol ?od ?pq ?st ?is_event WHERE {
    VALUES ?excl {<wd:Q19847637> <wd:Q19847637>}
    {
        SELECT DISTINCT ?s ?p ?o1 WHERE {
            VALUES ?s {<http://www.wikidata.org/entity/Q7186>}
            ?s ?p ?o1.
        }
    }
    OPTIONAL {
        ?o1 ?pq ?o2.
        OPTIONAL {
          ?o2 rdfs:label ?o2l.
          FILTER(lang(?o2l) = "en")
        }
        OPTIONAL {
          ?o2 schema:description ?o2d.
          FILTER(lang(?o2d) = "en")
        }
        FILTER(ISLITERAL(?o2) || BOUND(?o2l) || BOUND(?o2d))
        FILTER NOT EXISTS { {?o1 wdt:P31 ?t.} UNION {?o1
           wdt:P279 ?t.} }
        BIND(?o1 as ?st)
    }
    OPTIONAL {
        ?o1 rdfs:label ?o1l.
        FILTER(lang(?o1l) = "en")
    }
    OPTIONAL {
        ?o1 schema:description ?o1d.
        FILTER(lang(?o1d) = "en")
    }

    BIND(coalesce(?o2, ?o1) as ?o)
    BIND(coalesce(?o2l, ?o1l) as ?ol)
    BIND(coalesce(?o2d, ?o1d) as ?od)

    FILTER(?p NOT IN (skos:altLabel, rdfs:label, wdt:P31,
       schema:description, schema:dateModified))
    FILTER((ISIRI(?o) && (BOUND(?ol) || BOUND(?od))) || lang(?o)
       = "en" || (DATATYPE(?o) = xsd:dateTime || DATATYPE(?o) =
       xsd:date ))
    FILTER (?p NOT IN (<wd:Q19847637>, <wd:Q19847637>))
    FILTER ( NOT EXISTS {?p wdt:P31/wdt:P279* ?excl})
    BIND((DATATYPE(?o) = xsd:dateTime || DATATYPE(?o) =
       xsd:date) as ?is_event)
}
```

The incoming link query is the simplest one since it is only required to extract simple data, such as a triple that links the entity to the selected one and the label and the description of the entity. The reason for having a nested query for each possible entity value, provided in the `VALUES ?o` clause, is to impose a limit on the number of incoming entities extracted per target entity, by using the `LIMIT` clause on every nested query. If we were to transform it into a single query, with the `LIMIT` clause at the end of it, it would be limiting the total amount of incoming entities and not per single selected entity, of which there can be multiple per query.

The outgoing link query is the most complex one due to how we handle statement nodes, which was described in Section 4.2. It has the nested query modular format to perform the extraction for every query in the set. In this case, it, mainly, serves to improve the performance of the query. The most important part of this query is the foot. The first `OPTIONAL` clause handles statement nodes. The three `BIND` clauses perform the merging in case of a statement detection, which was, also, described and illustrated in Section 4.2. Finally, the data is filtered to exclude any irrelevant triples, such as the ones that represent identifiers in Wikidata. It is also worth mentioning that our queries extract all of their labels and descriptions in the English language. One of the possible prospects of future development is to allow multiple language support.

4.3.1.3 Entity Pruning

After each depth iteration, it is important to filter out any entities that we do not want to explore further. These entities typically include broader or more general concepts. It is done by a **sparql!** query that extracts, for each entity, its in-degree centrality. A query example can be found in Listing 4.15. This query, just like the previous ones, is highly modular, where each module represents a single nested query, for each entity. The nested queries are required to significantly improve the processing time of this query. Each nested query also has a limit on the number of output triples, since certain entities in certain graphs can have a plethora of incoming entities. Each entity is, then, filtered according to its in-degree centrality compared to the maximum in-degree centrality set up by the user in the narrative configuration stage.

```
SELECT ?x (COUNT(DISTINCT ?s) as ?count) WHERE {
    {
        SELECT DISTINCT ?s ?p ?x WHERE {
          VALUES ?x {<http://www.wikidata.org/entity/Q60504>}
          ?s ?p ?x
        } LIMIT 10000
    } UNION {
        SELECT DISTINCT ?s ?p ?x WHERE {
          VALUES ?x {<http://www.wikidata.org/entity/Q151985>}
          ?s ?p ?x
        } LIMIT 10000
    } UNION {
        SELECT DISTINCT ?s ?p ?x WHERE {
          VALUES ?x {<http://www.wikidata.org/entity/Q687572>}
          ?s ?p ?x
        } LIMIT 10000
    }
}
GROUP BY ?x
```

Listing 4.15: SPARQL Query Entity Pruning Example

The extracted relevant entities form the foundation for subsequent steps in the narrative extraction process, such as event identification, entity classification, and relationship establishment. These entities expand the scope of the narrative, enabling the exploration of diverse event occurrences associated with the main entity. For example, if the main entity is a historical figure, relevant entities may include their achievements, contemporaries, and historical events they were part of.

Suppose the main entity is "Leonardo da Vinci" and relevant entities include "Mona Lisa", "Vitruvian Man" and "Renaissance Art Movement". These relevant entities provide a broader context for exploring Leonardo's contributions, artworks, and historical significance.

The inclusion of relevant entities enhances the richness and depth of the extracted narrative. It offers a more holistic understanding of the interconnected events and relationships surrounding the main entity. The narrative becomes multi-dimensional, encompassing not only the main entity but also its associated events, context, and contributions.

4.3.2 Entity Classification

The process of entity classification is a fundamental step in structuring and organizing information within a narrative. This section explores the entity classification process in detail, including the classification types, SPARQL Protocol and RDF Query Language (SPARQL) queries for entity classification, conflict resolution strategies, and batch querying techniques.

Entity classification involves categorizing extracted relevant entities into one of the following six classes:

- **Person/Group**
- **Location**
- **Event**
- **Reason**
- **Manner**
- **Other**

These classes help define the characteristics and relationships of entities, facilitating data retrieval, analysis, and interpretation. The entity classification process can be divided into several key aspects, including classification type, **sparql!** query for entity classification, and conflict resolution strategies.

4.3.2.1 Manual Classification

Manual classification is the most basic of classification methods and involves human intervention in assigning classes to entities. It requires domain expertise and a deep understanding of the data and ontology. Users are presented with a full set of extracted relevant entities and must review each entity and determine the most appropriate class based on their knowledge and the entity's characteristics. Manual classification offers precision and accuracy but can be time-consuming and challenging to scale.

Manual classification plays a special role in cases where the entity's meaning is subjective in nature and may depend on the narrative type that the user expects. Consider entities defined in Listing 4.16. The `ex:UnitedStates` entity is used twice, both times in different contexts. When we are classifying entities, we do not account for this context, and, in this case, user intervention is necessary.

```
@prefix ex: <http://example.org/> .

ex:John ex:livesIn ex:UnitedStates.

ex:WorldWar2 ex:participant ex:UnitedStates.
```

Listing 4.16: Multiple Meanings for Entity "United States".

4.3.2.2 Automatic Classification

Automatic classification leverages algorithms and SPARQL queries to assign classes to entities without human intervention. This approach relies on the pre-defined classes in the graph's configuration. The first step of entity classification is to extract, for each entity, the pre-configured classes that it belongs to, if any, by using the query exemplified in Listing 4.17. In this query, the configured classes are defined inside the `VALUES ?t` clause, meanwhile, the target entities are included in the `VALUES ?s` clause. This simple query extracts each pair `(entity, class)` that it can find a match for. Entities without a match are classified as "other".

```
SELECT ?s ?t {
    VALUES ?s { <http://www.wikidata.org/entity/Q151018>
       <http://www.wikidata.org/entity/Q60504>
       <http://www.wikidata.org/entity/Q151985>
       <http://www.wikidata.org/entity/Q687572>
       <http://www.wikidata.org/entity/Q178561> }
    VALUES ?t { <http://www.wikidata.org/entity/Q5>
       <http://www.wikidata.org/entity/Q16334295>
       <http://www.wikidata.org/entity/Q18608993>
       <http://www.wikidata.org/entity/Q17334923>
       <http://www.wikidata.org/entity/Q52511956>
       <http://www.wikidata.org/entity/Q1190554>
       <http://www.wikidata.org/entity/Q2574811> }
    FILTER EXISTS {?s wdt:P31/wdt:P279* ?t}
} ORDER BY ?s ?t
```

Listing 4.17: SPARQL Query for Entity Classification.

4.3.2.3 Conflict Resolution

When using automatic classification, an entity may be assigned multiple classes if it gets multiple matches. In this case, the algorithm needs to decide which specific class the entity belongs to. This can either be done manually or automatically. Automatic conflict resolution relies on the rule set defined in the classification parameters at the beginning of the extraction pipeline. This rule set consists of giving priority to certain classes to the detriment of others. For instance, if the user specifies that, in case, both, "Person" and "Location" classes are found for an entity, the "Person" class should get the priority. Given this rule, the algorithm, whenever it runs into this conflict, will, automatically, assign the "Person" class to the entity. This can be done for every combination of classes.

The classification process plays a critical role in building the narrative as it determines how each entity is perceived by the algorithm and how it is placed within the narrative. Narratives that are about the same topic but use different classes for the same entities may vary greatly in aspect.

4.3.3 Event Identification

Narrative extraction from semantic graphs necessitates the systematic identification of events. Events serve as the building blocks of narratives, allowing users to comprehend and engage with the information effectively. In our approach, we leverage timestamps associated with graph resources to identify events.

Traditional event extraction algorithms based on semantic graphs rely on graph ontologies to explicitly identify events, which, in most cases, come as entities within the graph. However, these approaches fail to capture implicit events that often go unnoticed. Our solution to this problem is to define two types of events:

- Entity events. (Explicit)
- Property events. (Implicit)

The identification process differs for each type of event.

This section provides a comprehensive description of event identification, showcasing its significance in the narrative extraction process.

In our approach, timestamps associated with graph resources play a pivotal role in event identification. Timestamps provide temporal context and structure to the semantic graph, enabling the algorithm to pinpoint significant occurrences and phenomena. By analyzing the semantic graph and its contents, we can derive events that encapsulate temporal information.

We consider as timestamps all literal values in the graph that have a data type of **xsd:dateTime** or **xsd:date**. We use the **DATATYPE()** function to detect timestamp literals. Consider the Listing 4.18.

```
ex:Book1 ex:publicationDate "1996-01-01T00:00:00Z".
```

Listing 4.18: Time datatype example.

This triple has a literal object of value "1996-01-01T00:00:00Z". The **DATATYPE()** value for this object is **xsd:dateTime**, therefore this triple represents an event. The type of the event will be determined by the context in which the triple was discovered.

4.3.3.1 Entity Events

Entity events are represented by entities that directly correspond to real-world events. These entities encapsulate significant occurrences or phenomena and serve as explicit representations of events within the semantic graph. Entity events, in our approach, are characterized by a classification that identifies them as events. The main characteristics of entity events include:

- Explicit classification as events.
- Directly encapsulates event information.

Consider the Listing representing a historical event **??**.

```
@prefix ex: <http://example.org/>.

ex:WW2 rdf:type ex:Event.
ex:WW2 ex:startDate "1939"^^xsd:date.
ex:WW2 ex:endDate "1945"^^xsd:date.

ex:SwissExpedition rdf:type ex:Event.
ex:SwissExpedition ex:date "1952-05-01"^^xsd:date.
```

Listing 4.19: Historical Event Graph.

In this example, `ex:WW2` and `ex:SwissExpedition` are classified as events by the graph. Thus, every triple that contains a timestamp will create one instance of the respective event. For instance, the `ex:WW2` event will have two instances, one for each timestamp, and the `ex:SwissExpedition` will have only one instance.

4.3.3.2 Property Events

Property events, just like entity events, are derived from properties whose range is a literal timestamp. However, unlike entity events, property event triples do not belong to an event entity, i.e. their subject entity is not classified as an event. In a traditional event extraction algorithm, these events would be ignored. The main characteristics of property events include:

- Are not explicitly defined as events.
- Event information gathering is non-trivial.

Consider the Listing **??**.

```
@prefix ex: <http://example.org/>.

ex:Person1 rdf:type ex:Person.
ex:Person1 ex:dateOfBirth "1910-03-15"^^xsd:date.

ex:Book1 rdf:type ex:Book.
ex:Book1 ex:publicationDate "1955-07-20"^^xsd:date.
```

Listing 4.20: Property Event Identification.

In this example, we are dealing with entities that are not classified as events by the graph, yet they do contain timestamp triples. In this case, a brand new event is created for each timestamp. Since there is no event associated with the timestamp triple, its property plays a crucial role in defining the event itself. For instance, "ex:dateOfBirth" and "ex:publicationDate" represent property events associated with the birth of a person and the publication of a book, respectively.

Entity events play a crucial role in narrative building as they represent significant occurrences directly. They serve as anchor points for narrative development and provide clear event entities around which the narrative can revolve.

Unlike property events, entity events are represented as nodes in the semantic graph. When building a narrative, we only consider the main entity and its immediate property events, as well as all linked entity events in the original semantic graph. This is due to the fact that, typically, information regarding non-event entities that are linked, directly or indirectly, to the main entity is not relevant to the narrative. For instance, consider the graph in Listing 4.21.

```
@prefix ex: <http://example.org/>.

ex:LeonardoDaVinci rdf:type ex:Person.
ex:LeonardoDaVinci ex:dateOfBirth "1452-04-14"^^xsd:date.
ex:LeonardoDaVinci ex:occupation ex:Painter.
ex:LeonardoDaVinci ex:educatedAt ex:UniversityOfFlorence.

ex:UniversityOfFlorence ex:memberOf ex:ORCID.
ex:UniversityOfFlorence ex:rector ex:LuigiDei.
```

Listing 4.21: Irrelevant Triple Example.

In this example, assuming that the topic of the narrative is "Leonardo da Vinci", we can see that, despite "ex:UniversityOfFlorence" being linked to the main entity, its data is completely irrelevant to the biography of "Leonardo da Vinci".

This topic, however, is subjective and may depend on the person's preferences or personal views. Take the graph in Listing 4.22 for example.

```
@prefix ex: <http://example.org/>.

ex:WW2 rdf:type ex:Event.
ex:WW2 ex:startDate "1939"^^xsd:date.
ex:WW2 ex:endDate "1945"^^xsd:date.
ex:WW2 ex:hasCause ex:AdolfHitler.

ex:AdolfHitler ex:dateOfBirth "1889-04-20"^^xsd:date.
ex:AdolfHitler ex:allegiance ex:GermanEmpire.
```

Listing 4.22: Triple relevancy duality.

In this example, it is reasonable to consider that some of the personal information about "ex:AdolfHitler" is relevant to the narrative about "World War II". However, it is impossible to infer, directly, whether this information is relevant or not. Thus, we simply assume that every entity, besides the main one, that is not classified as an event, represents another narrative in itself.

Property events, while not representing events directly, contribute to the narrative by providing temporal context to other entities. They enrich the narrative by providing more information that could be crucial for narrative understanding.

4.4 Narrative Building

The second major phase of our approach involves narrative building. This step is responsible for data linking and the generation of final results. It consists of steps: attribute mapping, narrative merging, and conversion to turtle format.

4.4.1 Attribute Mapping

Event attributes mapping is a fundamental step in the narrative extraction process, facilitating the creation of structured narratives, composed of events with attributes like "Who", "What", "Where", "When", "Why", "How" and "Related To". Instead of relying on user input or hardcoded rules, our approach leverages graph configuration, specifically, defined classes for each fundamental concept ("Person/Group", "Location", "Event", "Reason" and "Manner"), to assign attributes to extracted entities and properties. In this section, we detail the attribute mapping process for both entity and property events, showcasing examples and highlighting the nuances of each mapping process.

4.4.1.1 Entity vs Property Events

Recall that entity events are entities that directly correspond to real-world events within the semantic graph. These entities serve as explicit representations of events.

For entity events, attribute mapping is straightforward, as it directly relies on pre-configured classes and a set of rules to map event attributes to either an entity, a literal, or a property within the graph. The mapping rules differ for each event attribute.

Property events, on the other hand, are derived from properties whose range is a literal with a **xsd:dateTime** or **xsd:date** data type and do not belong to any event entity in the graph.

The main difference between entity event and property event attribute mapping lies in the data pool from which the triples are pulled. For entity events, this pool corresponds to all outgoing links of the event entity. For example, all of the triples defined in Listing 4.23 will be considered when mapping attributes of the "WW2" event, since this entity was classified as an event.

```
@prefix ex: <http://example.org/>.

ex:WW2 rdf:type ex:Event.
ex:WW2 ex:startDate "1939"^^xsd:date.
ex:WW2 ex:endDate "1945"^^xsd:date.
ex:WW2 ex:hasCause ex:AdolfHitler.
```

Listing 4.23: Event entity data pool.

Unlike the case with entity events, property events are represented by a single triple that cannot encapsulate all of the information regarding the event. Therefore, it is necessary to develop a mechanism that would be able to group triples that belong to the same event instance together.

We achieve this by clustering RDF triples of the same entity based on the Ratcliff-Obershelp similarity between each triple's property labels. Only properties with similarity scores greater than or equal to a user-configured threshold are considered.

The Ratcliff-Obershelp similarity between two strings s_1 and s_2 is calculated as:

$$\text{Ratcliff-Obershelp}(s_1, s_2) = \frac{2 \times \text{Longest Common Subsequence}(s_1, s_2)}{|s_1| + |s_2|} \tag{4.1}$$

where:

$$\text{Longest Common Subsequence}(s_1, s_2) \text{ is the length of the longest common subsequence}$$
$$|s_1| \text{ is the length of string } s_1$$
$$|s_2| \text{ is the length of string } s_2$$

For instance, in Figure 4.4 we can find an example of a RDF triple clustering, based on the "date of death" event property.

Albert Einstein	
Property	Object
date of death	18 April 1955
father	Hermann Einstein
place of death	Princeton
manner of death	natural causes
occupation	scientist

Albert Einstein	
Property	Object
date of death	18 April 1955
father	Hermann Einstein
place of death	Princeton
manner of death	natural causes
occupation	scientist

Figure 4.4: Triple Clustering for Property Events

In this figure, properties "place of death", "manner of death" and "date of death" are clustered into a single event and represent all the available information regarding the event. The respective triples will be used as the data pool for the attribute mapping process for this event, following a similar pipeline to the one in entity event mapping.

4.4.1.2 "Who" Attribute

All entities in the source data pool classified as "Person/Group" are mapped as the "Who" attribute. These entities represent individuals or groups involved in the event.

Consider an example of an entity event representing "Battle of Britain" that can be found in Figure 4.5.

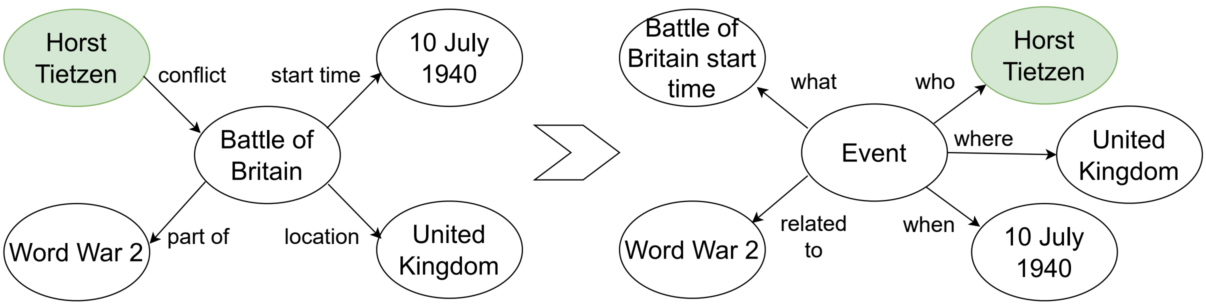


Figure 4.5: Entity Event "Who" Mapping

In this example, there is an entity, "Horst Tietzen", classified as a "Person/Group" that is included in the event data pool. Therefore, it will be used to create an instance of the "Who" attribute pointing to the entity itself.

4.4.1.3 "What" Attribute

The "what" attribute mapping differs, slightly, from property to entity events.

The label of the event entity, concatenated with the property label that contains the timestamp of the event (e.g., "start time"), is mapped as the "What" attribute. This provides a concise, yet accurate, description of the event.

Continuing with the "Battle of Britain" example, in Figure 4.6 we can find the respective "What" attribute mapping.

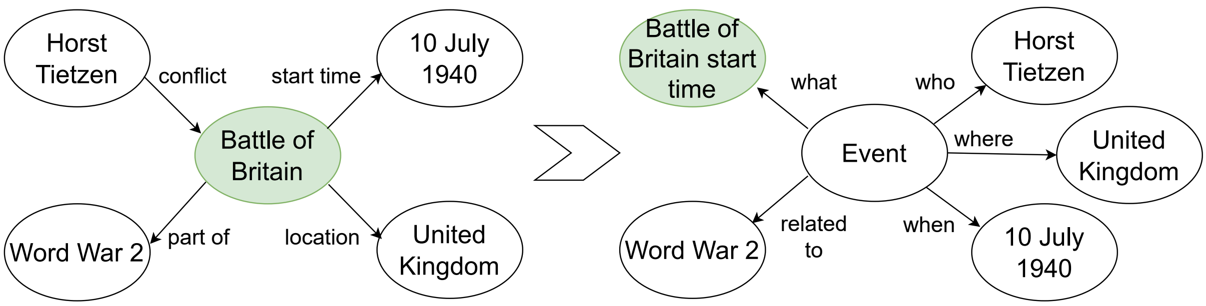


Figure 4.6: Entity Event "What" Mapping

Since we are dealing with an event entity, the value of the "What" attribute is equivalent to a merge of the respective event entity's label with the label of the property that instantiated the event, in this case: "start time". The resulting value of the "What" attribute is: "Battle of Britain start time".

On the other hand, for property events, the label of the property that instantiates the event itself is mapped as the "What" attribute. Consider an example of a property event data pool found in Figure 4.7.

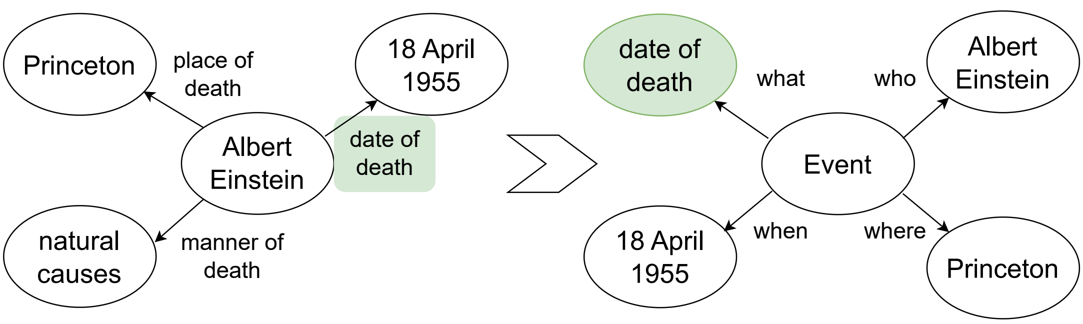


Figure 4.7: Property Event "What" Mapping

In this figure, the label of the property that originated the event is "date of death". This value is used as the value of the "What" attribute of the property event.

4.4.1.4 "Where" Attribute

Just as with the "Who" attribute, entities in the event data pool classified as "Location" are mapped as the "Where" attribute. The entities represent the locations associated with the event.

Once again, using the "Battle of Britain" example, the "Where" attribute mapping can be found in Figure 4.8.

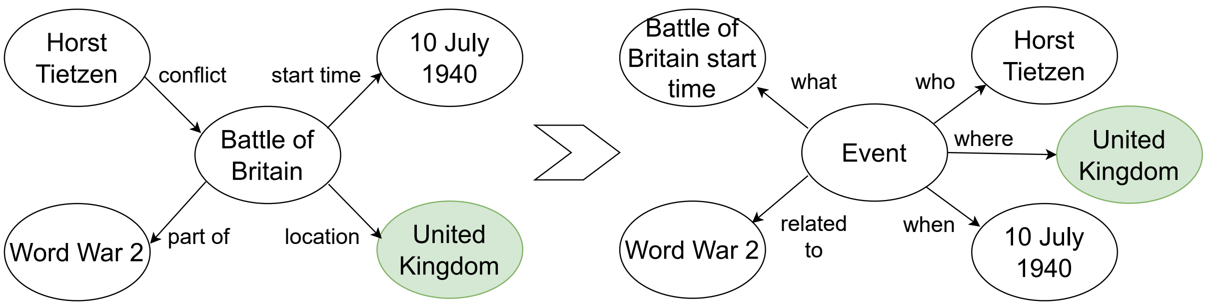


Figure 4.8: Entity Event "Where" Mapping

In this case, "United Kingdom", which was classified as a location, is mapped as the "Where" attribute for the entity event.

4.4.1.5 "When" Attribute

The value assigned to the property that contains the timestamp of the event is mapped as the "When" attribute. This attribute represents the temporal information of the event.

In the "Battle of Britain" example, the "When" attribute mapping is depicted in Figure 4.9.

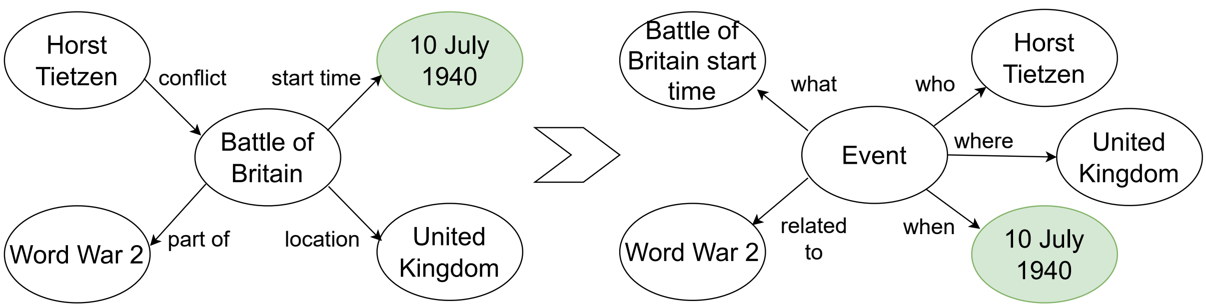


Figure 4.9: Entity Event "When" Mapping

We can see that the property that instantiated the event (due to having a `xsd:dateTime` or `xsd:date` value) is "start time". The value associated with that property, in this case, "10 July 1940", is mapped as the "When" attribute.

4.4.1.6 "Related To" Attribute

Entities in the event data pool classified as "Event" are mapped to the "Related To" attribute. These entities capture events related to other events, providing contextual connections.

Using the "Battle of Britain" as the example, the "Related To" attribute mapping is presented in Figure 4.10.

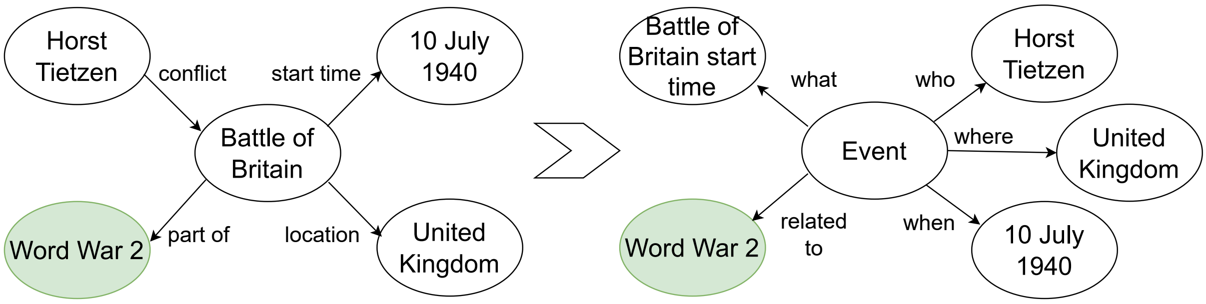


Figure 4.10: Entity Event "Related To" Mapping

In this figure, there is an entity, "World War II", classified as "Event", that is present in the data pool. Thus, it will be mapped as the "Related To" attribute for the event "Battle of Britain".

4.4.1.7 "Why" and "How" Attributes

The "why" and "how" attributes are mapped similarly to the "who", "where" and "related to" attributes. These attributes rely on assigned classes "reason" and "manner", respectively, to the entities in the event data pool. Entities classified as "reason" will be mapped to the "why" attributes. Similarly, entities classified as "manner" will be mapped to the "how" attribute.

After gathering relevant data and mapping entities to event attributes, the subsequent and final step is to assemble the narrative. In this phase, we focus on merging intersecting narratives, organizing events into a coherent and meaningful sequence, and converting intermediate narrative representations into turtle format.

4.4.2 Narrative Merging

In cases where intersecting narratives exist, merging them into one cohesive narrative can provide a more comprehensive and coherent storytelling experience. Narrative merging is optional and can be toggled in the configuration parameters.

Definition 4.4.1 (Intersecting Narratives)**.** Two narratives are intersecting if the respective main entities are linked in the original semantic graph.

Take the semantic graph in Listing 4.24 as an example. Assuming that both "John Smith" and "Jane Smith" were provided as topics for narrative extraction, since their respective entities are linked together in the semantic graph, their respective narratives will be merged into one.

```
@prefix ex: <http://example.org/> .

ex:John ex:name "John Smith".
ex:John ex:date_of_birth "15 April 1979".
ex:John ex:spouse ex:Jane.

ex:Jane ex:name "Jane Smith".
ex:John ex:date_of_birth "8 July 1983".
ex:Jane ex:countryOfResidence ex:Canada.
```

Listing 4.24: Intersecting Narratives example

The merging process simply involves combining all of the event sets of intersecting narratives into a single set, corresponding to the final narrative. Duplicate events, i.e., events that share the same attribute values, are also pruned so that only one copy remains. The `nrtv:about` attributes of respective narratives are also combined into one by concatenation. The resulting narrative is treated as a normal narrative, at the end.

In our approach, the narrative is conceptualized as an ordered sequence of events that occurred within a specific period of time. Each event in the extracted set comes with a timestamp, allowing for the establishment of a clear timeline. By arranging events chronologically, we create the backbone of the narrative, offering a structured timeline for the sequence of events.

Chronological ordering is crucial for constructing a narrative that makes temporal sense. It ensures that events are presented in a logical sequence, allowing readers or analysts to follow the narrative's timeline seamlessly. This chronological structure provides clarity and context to the story being told.

4.4.3 Converting to Turtle Format

To convert the intermediate narrative structures into a narrative semantic graph in turtle format, we follow these steps:

- Define the namespaces and prefixes as needed for the RDF data.
- Create nodes for events, attribute nodes, and the narrative node using RDF triples.
- Define the types of events, attributes, and narrative using `rdf:type` triples.
- Link events to their attributes and the narrative node using appropriate predicates.
- Link narrative attributes to the narrative node.

- Ensure that event timestamps, attribute values, and other relevant information are represented accurately in the RDF triples.

The final narrative graph can be seamlessly embedded into the original semantic graph, allowing for deeper analysis and exploration of the interconnected data. This integration preserves the context and relationships between narrative events and the broader knowledge graph.

In summary, in this chapter, we defined four key research questions that we aim to explore in our work. We started by defining the narratives, supported by an ontology specification, which should help answer the research question **Q1**. We followed by describing the graph configuration stage as a way to provide adaptability of our approach to different graphs and domains, as well as creating a framework that helps our algorithm understand the contents of each graph and ontology will minimal work, looking to answer, partially, the research question **Q2**. Then, we provided an in-depth description of the pipeline of our approach, which consists of two major steps: information retrieval and narrative building. By specifying an approach for narrative extraction from semantic graphs, we aim to answer the research questions **Q2** and **Q3**. In the following chapter, we describe the practical implementation of the approach detailed in this chapter.

Chapter 5

Implementation

In the pursuit of automating and streamlining the extraction of narratives from semantic graphs, our research led us to develop a practical solution. We recognized the need for a tool that could, not only facilitate this intricate process but also provide a user-friendly interface for users to interact with. To fulfill this requirement, we developed Narrative Extractor from Semantic Graphs (NESG).

In this section, we delve into the practical implementation of the NESG[1]. NESG consists of five key components: user, Graphical User Interface (GUI), node server, input semantic graph and data storage. These components, together, lay the groundwork for the entire NESG workflow, orchestrating the flow of data and interactions within the system. The central hub of activity is the node server, which plays a pivotal role as the control center, coordinating all processes and data exchanges. It is responsible for querying the input semantic graph, managing data storage, handling client requests, and delivering the final results.

The GUI[2], using NESG, standing as a bridge between users and the NESG, is a crucial element in making the narrative extraction process more approachable. Through the GUI, users gain access to a plethora of features, allowing them to interact with the system effortlessly. This web-based interface empowers users to configure their semantic graphs, fine-tune narrative extraction parameters, and navigate through the entire process with ease. By facilitating user interactions and presenting results in a comprehensible format, the GUI simplifies what would otherwise be a complex undertaking.

One fundamental aspect worth noting is the current data storage mechanism employed by NESG. As of now, all data is housed within a single JSON file, which serves as the central repository accessed by the node server. This approach allows for data persistence and retrieval, ensuring that the system maintains continuity and coherence throughout the narrative extraction process. The diagram of the NESG system can be visualized in Figure 5.1.

[1] NESG Source Code. https://archive.softwareheritage.org/browse/origin/https://github.com/DaniilLystopadskyi/nesg

[2] NESG User Interface. https://www.dcc.fc.up.pt/nesg/

In the subsequent sections of this chapter, we delve deeper into the specifics of how each component operates and contributes to the overall functionality of NESG. Furthermore, we explore various challenges faced during the implementation phase and the solutions that were devised to overcome them.

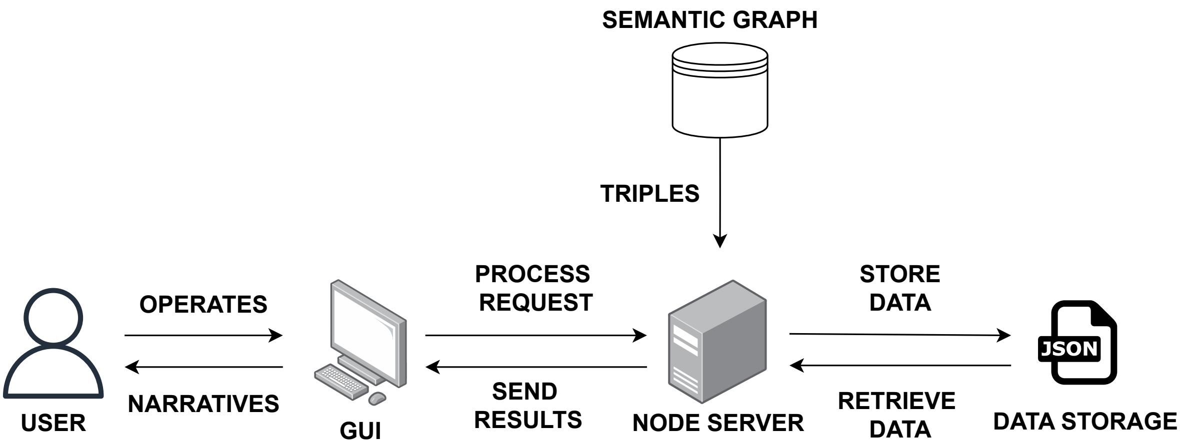


Figure 5.1: NESG system.

In the following sections, we provide an in-depth exploration of the software architecture, file structure, GUI, and the flow of the narrative extraction process.

5.1 Graphical User Interface

The web interface is designed to guide users through the narrative extraction process. It provides a modular architecture to represent contents within the main page. The main page contains a set of seven tabs in the header, which can be seen in Figure 5.2. Tabs that are greyed out are locked and cannot be accessed at the time. These tabs are unlocked by advancing the extraction process. Each tab corresponds to one of the steps of our narrative extraction process:

- Graph Configuration
- Startup
- Main Entity Selection
- Entity Classification
- Results Showcase
- Evaluation

To maintain a compact and organized layout, each of these steps is further subdivided into sub-tabs, allowing users to focus on specific aspects of the current step. For instance, the "Start"

Figure 5.2: NESG main tabs.

tab has the tabs depicted in Figure 5.3. The "graph" sub-tab includes graph selection. The "narrative" sub-tab contains all of the main narrative parameters. The "technical" sub-tab has technical parameters, such as similarity threshold and in-degree centrality limit. Finally, the "classification" sub-tab provides access to all of the parameters related to classification.

Figure 5.3: NESG start sub-tabs.

The front page contains the main container where all of the contents are placed. Each tab has an associated html file that contains its contents. When a new tab is loaded, the contents of its **HTML!** (**HTML!**) file are transferred into the main container, replacing any existing ones. Each page is broken down into sections, representing sub-tabs. By switching sub-tabs, the respective sections get shown or hidden. The main container can be seen in Figure 5.4.

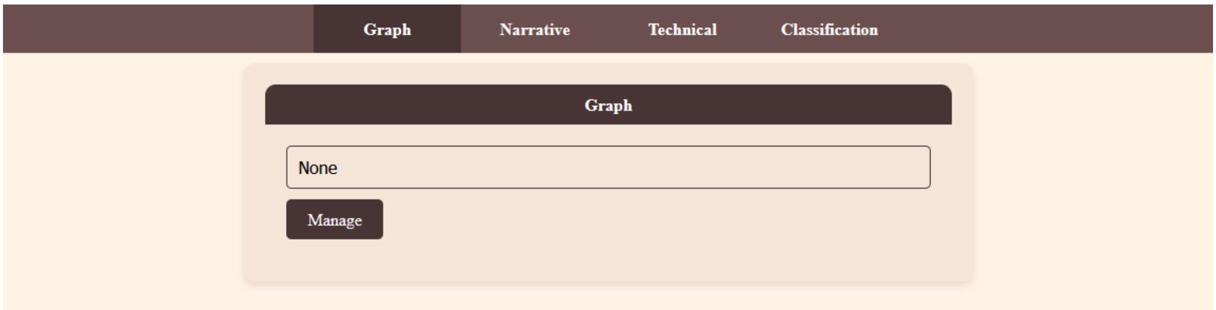


Figure 5.4: NESG main contents container.

The contents of each sub-tub are divided even further into modules. Each module consists of a label and a contents container. Modules are supposed to represent individual elements of a page in a modular and unified format. This structure allows us to insert and remove modules with ease. An example module representing the graph selection element can be found in Figure 5.5.

An example of a full page can be found in Appendix A.

5.1.1 Page Navigation

At the bottom of the main page, two main buttons control the flow of the narrative extraction process: "Back" and "Next". The pages are linked as a sequence, meaning that each page has a previous page and a next page. Each page comes with its own loading and transition functions.

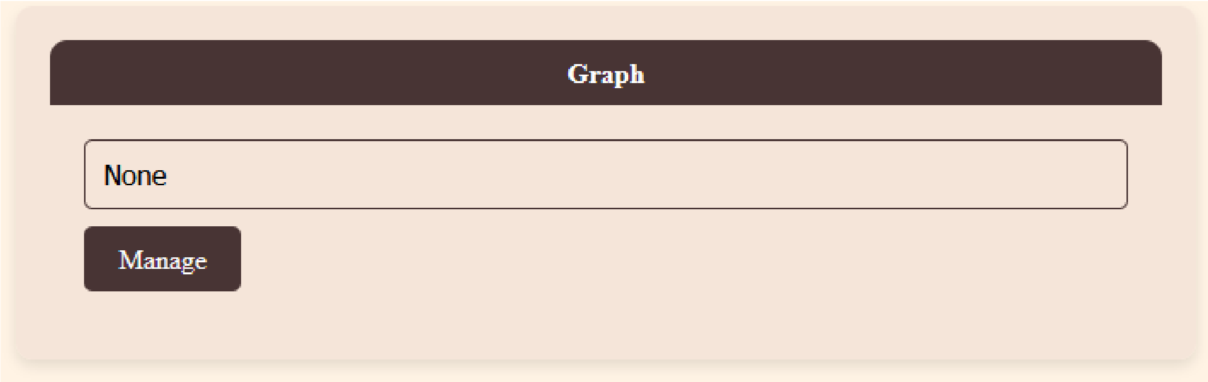


Figure 5.5: NESG module.

The loading function is responsible for setting up the contents of the page and performing any necessary calculations and DOM modifications. The transition function executes the code necessary to transition to the next step of the process. It involves extracting data, processing it, and storing it in suitable variables. When the "Next" button is pressed, the transition function of the current page is called and, once it finishes, the loading function of the next page is called. When the "Back" button is pressed, only the loading function of the previous page is called. These buttons can be seen in Figure 5.6.

Figure 5.6: NESG flow buttons.

5.2 Data Storage

All of the data that is extracted and computed must be stored for later re-use. This data includes graph configuration parameters, classes assigned to entities, in-degree entity centrality, and property data. This data is stored for each graph in a JSON file, located on the server machine. This data is retrieved by the server when it is needed. An example of the JSON file used to store data regarding the Wikidata graph can be found in Figure 5.1.

```json
{
  "Wikidata": {
    "classes": {
      "person": [
        "http://www.wikidata.org/entity/Q5"
      ],
      "location": [
        "http://www.wikidata.org/entity/Q17334923"
      ],
      "event": [
        "http://www.wikidata.org/entity/Q1190554"
      ],
      "reason": [],
      "manner": []
    },
    "entityClasses": {
      "http://www.wikidata.org/entity/Q56226": "person"
    },
    "entityCentrality": {
      "http://www.wikidata.org/entity/Q155647": 2878
    },
    "propertyData": {
      "P7993": {
        "label": "Treccani's Dizionario di Filosofia ID",
        "description": "identifier for an item in the Dizionario di Filosofia"
      }
    },
    "excludedTypes": [
      "wd:Q19847637"
    ],
    "endpoint": "https://query.wikidata.org/sparql",
    "type": "wdt:P31",
    "subclass": "wdt:P279",
    "description": "schema:description"
  }
```

Listing 5.1: JSON graph data storage

This implementation presents several limitations when it comes to handling large amounts of data. As an avenue for future work, we need to replace this simple storage system with a fully functional database.

5.3 Challenges Encountered

During the implementation of our methodology, several challenges were encountered, most of them centered around semantic graph querying and scalability. In the following subsections, we detail each major obstacle and what techniques and methods we used to overcome them. All of the queries were sent using the `fetch` Application Programming Interface (API), as POST requests to the endpoint.

5.3.1 Query Limit

On certain occasions, such as during the relevant entity extraction process, we were required to provide a large number of entities as input to the SPARQL Protocol and RDF Query Language (SPARQL).

For instance, in order to implement the depth into our algorithm, we were required to do a recursive querying where each consecutive query would take the previously found entities as input and return data for each entity. The process starts with just one entity - the main entity. It searches for all data linked to this entity and retrieves all linked entities. It, then, applies the same function for the newly found entities that are going to be treated as input entities. This process can be summarized in the Algorithm 1.

Algorithm 1 Relevant Entity Extraction

1: **function** ExtractRelevantEntities(lastEntities, data)
2: $\quad data \leftarrow data + \textsc{SarqlQuery}(lastEntities)$
3: $\quad lastEntities \leftarrow data.lastEntities$
4: $\quad$ ExtractRelevantEntities(lastEntities, data)
5: **end function**

If the depth is set to a value greater than 1, we will perform multiple iterations of this function, with each iteration receiving an exponentially increasing number of input entities. Thus, we cannot simply put all of the entities inside a `VALUES` clause, since the query will return a timeout. If we had done one query per entity, which would have been efficient, we would run into the "HTTP 429: too many requests" error.

To solve this issue, we developed "modular" queries. A "modular" query is a query that comprises three main parts: head, body, and foot. The head and the foot are static query pieces. However, the body of this type of query is composed of one or more modules. Each module represents a single nested query that retrieves the desired results only for one entity. Multiple modules are combined using the `UNION` clause to form the body of the main query, which would be equivalent to putting all of those entities inside a `VALUES` clause, but much more efficient. More specific examples of this query implementation can be found in Chapter **??**.

5.3.2 Query Timeouts

Sometimes, the number of input entities is so big that we cannot use modular queries. This leads us to resort to more desperate measures.

To accommodate this case, we developed a batch querying mechanism. Batch querying involves separating input entities into multiple, equally sized, batches. The number of batches per event set is determined by the `mach_batch_size` technical parameter, which is, by default, 20. This means that, given 100 entities as input, when using batch querying, those entities would be allocated to 5 different batches and a query will be made for each batch. This way, we establish a balance between the rate of queries and the number of entities per query. This process can be summarized in Algorithm 2.

Algorithm 2 Batch Querying

```
batches ← GetBatches(entities)
for all batch in batches do
    SarqlQuery(batch)
end for
```

With this approach, we are able to reliably perform any query on the SPARQL endpoint, while maximizing the response times and minimizing errors.

Chapter 6

Evaluation

The evaluation of narrative extraction algorithms presents several challenges due to the multi-faceted nature of narratives and the inherent subjectivity involved in assessing their quality.

Narrative quality is inherently subjective in nature, varying from person to person. What one individual considers a compelling narrative might differ significantly from the perspective of another person. This subjectivity poses challenges in devising objective evaluation criteria and metrics that can consistently measure the quality of extracted narratives. The standard way (and the simplest one) is to perform a manual assessment of extracted narratives, which we incorporate into our evaluation methodology.

Most of the existing evaluation procedures focus on human-readable text, such as student essays [37] and narratives produced by generative models [6], making the manual evaluation more accessible. However, in our case, narratives are presented in the form of semantic graphs. This creates a division when it comes to evaluation criteria. When we are trying to assess the quality of a narrative, the question becomes whether we are assessing the quality of the narrative itself or the quality of the structure that represents it. In order to accommodate both worlds, we consider both quantitative and qualitative measures.

Furthermore, unlike many natural language processing tasks that have well-defined benchmarks and ground truth datasets for evaluation, narrative extraction lacks a definitive ground truth, especially when it comes to semantic graphs. Narratives can vary widely in style, content, and structure, making it challenging to establish a universal set of corrected answers for evaluation purposes.

Narratives encompass a wide range of genres, from biographies and historical accounts to event descriptions and religious narratives. Evaluating algorithms across diverse narrative genres requires a nuanced understanding of the characteristics and storytelling conventions of each genre.

Efficiently evaluating narrative extraction algorithms across large datasets and diverse sources, such as semantic graphs, requires scalable evaluation methodologies and tools. Scalability challenges arise when dealing with extensive data sources and resource-intensive algorithms.

Addressing these challenges required a thoughtful and multifaceted approach to evaluation. We designed the evaluation methodology with these challenges in mind, aiming to strike a balance between quantitative metrics and qualitative assessments to provide a comprehensive evaluation of our approach.

In this section, we delve into the comprehensive evaluation methodology, data collection process, the various statistics utilized, and data analysis and comparison to measure the performance of our approach. This evaluation methodology is designed to measure the effectiveness of the approach across a variety of dimensions, including semantic graphs, narrative depths, and genres.

Since there are no prominent tools or datasets that can be used for comparison and assessment of our approach, we have to rely on our own methods to validate our results. We performed manual annotation of the narratives, which is further described in Section 6.2. Furthermore, we developed our own baseline algorithm, based on the approach proposed by Blin [5], which aims to provide ground truth to our results.

Every narrative is represented as a set of Resource Description Framework (RDF) triples, hence a meaningful narrative corresponds to triples that provide some storyline, with a beginning and an ending. The contents that are encapsulated within these margins, make up the main story body of the narrative. It should provide a clear and coherent flow of, interconnected, events that form one or more plot lines of different lengths.

For the purpose of evaluation, we extracted a handful amount of narratives[1], using Narrative Extractor from Semantic Graphs (NESG), representing different topics and narrative genres. The evaluation was performed by calculating a variety of statistics from the acquired narratives. It consists of comparing the results across different graphs, genres, and depth levels, as well as the baseline results. The evaluation process was split into three categories: narrative evaluation, event evaluation, and triple evaluation. The narrative evaluation allows us to assess the quality of the results on a global scale, by analyzing narratives as a whole. The events and triples evaluation allow us to validate the event building and the triple selection parts, respectively, of our approach.

Furthermore, the narrative evaluation was split according to different narrative quality elements:

Complexity Complexity conveys the richness and the extensiveness of the narrative.

Coherence Coherence represents the logical consistency or connection between the events of the narrative.

Character development Character development expresses character progression and presence along the narrative.

[1]NESG Results (results branch). https://github.com/DaniilLystopadskyi/nesg/tree/results

Pacing Narrative pacing is related to time, representing the temporal flow of the narrative.

These elements were evaluated based on certain metrics, representative of each narrative element, that were extracted from the resulting data.

Each evaluation category was validated according to the respective target features. The narratives were evaluated based on the "score" feature, which is a value, from 0 to 5, that was, manually assigned, by the author of NESG, to each narrative, representing the quality of the narrative as a whole, as well as in comparison to other extracted narratives. The events were evaluated based on the "relevant" feature, which is a binary, manually annotated, feature that tells whether the event is relevant or not to the narrative. Finally, the triples were evaluated based on the "used" and "relevant" binary variables, which represent, respectively, whether the triples were used for the narrative generation and whether the triple is relevant to the narrative, which was, also, manually annotated by the authors.

Through this comprehensive evaluation methodology, we aim to provide a thorough understanding of the capabilities and limitations of our narrative extraction approach, as well as try to answer the research question **Q4**. Moreover, the main goal of this chapter is to determine whether we succeeded at our objective of extracting meaningful narratives from semantic graphs. By considering semantic graphs, narrative depths, and genres, we ensure that our approach can adapt to diverse narrative styles and effectively extract meaningful information. The evaluation section will delve deeper into the specific methodologies and metrics used to assess the results of our approach, ultimately contributing valuable insights to the field of narrative extraction.

6.1 Data Extraction

The evaluation methodology aims to measure the performance of our narrative extraction approach across different semantic graphs, narrative depths, topics, and genres. The lightweight evaluation focused on two prominent semantic graphs:

- **Wikidata**
- **DBpedia**

To facilitate a robust evaluation, a total of **100** narratives (50 per graph) were extracted, annotated, and analyzed. These narratives were obtained from **25** unique topics, categorized into five distinct narrative genres (5 topics per genre): People (Biographies), Places, Events, Religions, and Movements. By choosing topics from different genres, we are able to analyze and compare the quality of our approach across different domains.

6.1.1 Narrative Genres and Topics

Narratives span across diverse genres, each with its own unique characteristics and storytelling conventions. To ensure a well-rounded evaluation, we deliberately selected a range of narrative genres for assessment:

- **People (Biographies)**: Biographical narratives focus on the life stories of individuals, detailing their achievements, experiences, and contributions to society. These narratives often follow a chronological structure and provide insights into the personal and professional journey of a person.
- **Places**: Narratives centered around places describe locations, their historical significance, cultural attributes, and geographical features. Place-based narratives may include descriptions of landmarks, historical regions, or events that took place there.
- **Events**: Event narratives revolve around specific occurrences or incidents providing details about what happened, when it happened, where it took place, who was involved, and what other events are linked to this one that, together, form a story behind a single event.
- **Religions**: Religious narratives delve into the beliefs, doctrines, and stories associated with different faiths and religious traditions. These narratives may include religious texts, myths, and accounts of spiritual experiences.
- **Movements**: Movement narratives explore events that span across multiple generations, offering insights into the cultural, social, and political dynamics of a particular movement period.

By including narratives from these diverse genres, our evaluation methodology accounts for variations in narrative content, structure, and context. This ensures that our approach can adapt to different narrative styles and effectively extract meaningful information, regardless of the genre.

For each one of the five genres, **5 different topics** were manually picked, such that each topic can be found in any of the three selected graphs, for extraction purposes. The selected topics and their distribution across genres can be found in Table 6.1.

6.1.2 Narrative Depth

The depth parameter of a narrative directly influences the amount of information extracted and the distance of it from the main entity (topic). Meaning that, higher depth values should result in more complex, yet, sometimes, disconnected narratives.

Narrative depth plays a pivotal role in our evaluation methodology. We chose to evaluate our approach at two distinct depth levels: $depth = 1$ and $depth = 2$. The rationale behind this choice

Table 6.1: Chosen topics per genre.

Person (PE)	Kim Jong-un, Pablo Picasso, Wolfgang Amadeus Mozart, Abraham Lincoln, Isaac Newton
Place (PL)	Mount Everest, Wembley Stadium, Kabul, Myspace, Manas National Park
Event (E)	Assassination of John F. Kennedy, Battle of the Bulge, 2020 United States presidential election, Apollo 11, Industrial Revolution
Religion (R)	Christianity, Hinduism, Buddhism, Islam, Sikhism
Movement (M)	Modernism, Purism, Futurism, Renaissance, Impressionism

is to examine how narrative depth influences the quality and comprehensiveness of extracted narratives.

- $depth = 1$: At this level, the narrative extraction process focuses on capturing immediate and essential information to the main entity. The narrative is concise and provides a broad overview of the attributes and relationships of the entity. This depth level aims to extract core information efficiently.
- $depth = 2$: Represents a deeper narrative exploration, where the extraction process goes beyond the immediate attributes of the main entity. It involves retrieving additional layers of information, including interconnected entities and events. This depth level aims to provide a more comprehensive and detailed narrative, offering a richer contextual understanding of the main entity.

By evaluating our approach at both depth levels, we aim to assess its adaptability to different user requirements and scenarios. Some users may prefer concise narratives for quick overviews, while others may seek more detailed and extensive narratives for in-depth research or analysis. This duality in-depth evaluation allows us to cater to a broader spectrum of user preferences and information needs.

Moreover, this approach helps us to assess the underlying defect of the depth parameter. It allows us to better understand how depth affects the event and triple relevance in relation to the narrative, by comparing results from different depth values for the same narratives.

In summary, our evaluation methodology takes into account the diversity of narrative genres and the impact of narrative depth, ensuring a robust assessment of the capabilities across various dimensions of our narrative extraction approach.

6.2 Results and Analysis

The data extraction process took multiple days to complete and resulted in a total of 100 extracted narratives, which were composed of 1209 events and 105467 triples, in total. Each narrative and event was manually analyzed and annotated during the extraction process.

In this section, we perform an analysis of the extracted data, with the goal of deducting whether we managed to achieve our goals or not. The narrative analysis was further split into the "Wikdiata" and the "DBpedia" analysis.

Our analysis and evaluation methodology consists of evaluating different aspects of our approach. We start off with a global evaluation by analyzing the narratives based on their assigned scores. This gives us insight into the overall performance of our approach, across different genres and semantic graphs. To further solidify our results, we then explore other components of a narrative: complexity, coherence, character development, and pacing. These components consist of different metrics, selected for their analysis.

In order to evaluate, exclusively, the narrative-building step of our approach, we conducted an exploratory analysis of the data regarding the extracted events and their attributes. This allows us to assess the quality of the event-building mechanism and how it relates to the rest of the process.

Finally, the triple selection step was evaluated by analyzing the data of the extracted triples. Similarly to the analysis of the events, we extracted meaningful metrics and statistics regarding the triples and performed an exploratory analysis of those.

6.2.1 Narratives Evaluation

The narrative evaluation methodology consists of performing the validation based on a set of different metrics. The selection of metrics is driven by the need to assess the quality, coherence, pacing, and character development of the narrative. These metrics are carefully chosen to capture both quantitative and qualitative dimensions of narratives, allowing for a nuanced evaluation.

Each extracted narrative was assigned a "score". Score is a numerical feature, with values ranging from 0 to 5, which indicates how "good" a narrative is. This value was attributed manually by the author responsible for the extraction program during the extraction process as a way to introduce human evaluation into our methodology. Obviously, the assigned value is subjective and may not reflect the general perception of the narrative, which is why our evaluation is experimental and should be taken with a grain of salt.

However, the point distribution of the scores was not completely random and had a system behind it. Generally, narratives that had a well-defined beginning and an end were given, at least, 2 points to the score. If a narrative only had either a beginning or an end, but not both, it was given a minimum of 1 point. The rest of the points were given according to the intermediate contents of the narrative. If a narrative showed more specific and detailed events, it was given 1 point, depending on the density of these events. Lastly, the last points were given according to the relevancy of present events to the topic and its genre. The point distribution is summarized in Table 6.2.

Table 6.2: Score point distribution.

	has beginning	has end	density of specific and detailed events	density of relevant events	Total
points given	0 - 1	0 - 1	0 - 1	0 - 2	0 - 5

Furthermore, in order to better understand the importance of intermediate events with respect to each genre, the score assignment also took into account the reference narratives. For each genre, a reference narrative was extracted, representing one of the best narratives produced by our algorithm, for that genre. The reference narratives were: "Barack Obama" (Biographies), "Vienna" (Places), "World War II" (Events), and "Cubism" (Movements). An example narrative about "Wembley Stadium" can be found in the Appendix B as an example. These narratives are not directly evaluated in this section and were manually picked, through an extensive search, and extracted using our approach. Not every reference narrative is equal to one another, because different genres have demonstrated different quality of results. When assigning a score to a narrative, we took into consideration our own preferences, as well as how the narrative compared to the respective reference narrative (a score of 5 would indicate a perfect narrative).

With that being said, it allows us to create a reference point when comparing the results with each other. For instance, we could deduct whether narratives with a higher event count tend to get higher scores than the ones with a lower event count. In order to establish the baseline for narrative metric evaluation, we consider a value of 3.0 for the "score" variable as the baseline, meaning that, any narrative with a score of 3.0 or above is considered a feasible narrative.

To give an example of how a narrative, in RDF format, can be evaluated based on its quality, we showcase a narrative about "Pablo Picasso". The final version of this narrative is extensive and, for clarity and better readability, we will focus only on the important parts of this narrative.

As we have established previously, a good narrative should have a start and an end. How those are represented and interpreted depends on the genre. Consider the first events of the "Pablo Picasso" narrative in Listing 6.1. For the purpose of this illustration, we decided to exclude entities that represent characters, locations, and related events, since those take up large space, however, we will mention them, when need be, in our description. In this short example, the first event represents the birth of the subject in question. It is followed by an event that indicates the start of an education. Both events mark the beginning of something bigger and, since we are dealing with a biography (person) genre, we can conclude that those starting events are a good fit to represent the beginning of the life of a person.

```
event_1 a nrtv:Event;
    nrtv:what "date of birth";
    nrtv:when "1881-10-25T00:00:00Z";
    nrtv:who nrtv:character_0;
    nrtv:where nrtv:location_0;
    nrtv:where nrtv:location_1;
    nrtv:where nrtv:location_2.

event_2 a nrtv:Event;
    nrtv:what "educated at start time";
    nrtv:when "1897-01-01T00:00:00Z";
    nrtv:who nrtv:character_1;
    nrtv:where nrtv:location_3.
```

Listing 6.1: First events related to "Pablo Picasso".

Some of the intermediate events in the "Pablo Picasso" narrative include achievements, personal life events, and general life occurrences. Listing 6.2 contains a sample of some intermediate events in the life of "Pablo Picasso". In this example, we can see events responsible for dictating the residency change of the subject, marriage to another person, and an award acquirement. Since those events represent different parts of the life of a person, we can conclude that they are relevant to the biographical narrative. Furthermore, those events happen within a, relatively, short period of time, yet not on exactly the same date. This tells us that the narrative covers a wide range of events, without massive time gaps in the middle.

```
event_18 a nrtv:Event;
    nrtv:what "residence start time";
    nrtv:when "1961-01-01T00:00:00Z";
    nrtv:who nrtv:character_28;
    nrtv:where nrtv:location_8.

event_19 a nrtv:Event;
    nrtv:what "spouse start time";
    nrtv:when "1961-03-02T00:00:00Z";
    nrtv:who nrtv:character_29;
    nrtv:who nrtv:character_30;
    nrtv:where nrtv:location_9;
    nrtv:related_to nrtv:related_event_0.

event_20 a nrtv:Event;
    nrtv:what "award received Lenin Peace Prize point in time";
    nrtv:when "1962-01-01T00:00:00Z";
    nrtv:who nrtv:character_31.
```

Listing 6.2: Intermediate events related to "Pablo Picasso".

Lastly, the final events that can be encountered in the "Pablo Picasso" narratives are the ones depicted in Listing 6.3. All three of those events represent a closure to another event that was instantiated at an earlier point in the narrative. Moreover, the last event, "date of death", is the direct opposite of the first event, "date of birth". This provides closure to the narrative as a whole, with most, sometimes all, plot lines being closed. Overall, we consider this to be a good narrative, since it includes a start, and an end and encapsulates relevant information between these two.

```
event_22 a nrtv:Event;
    nrtv:what "spouse end time";
    nrtv:when "1973-01-01T00:00:00Z";
    nrtv:who nrtv:character_33;
    nrtv:who nrtv:character_34;
    nrtv:where nrtv:location_10;
    nrtv:related_to nrtv:related_event_1.

event_23 a nrtv:Event;
    nrtv:what "residence end time";
    nrtv:when "1973-01-01T00:00:00Z";
    nrtv:who nrtv:character_35;
    nrtv:where nrtv:location_11.

event_24 a nrtv:Event;
    nrtv:what "date of death";
    nrtv:when "1973-04-08T00:00:00Z";
    nrtv:who nrtv:character_36;
    nrtv:where nrtv:location_12;
    nrtv:where nrtv:location_13;
    nrtv:where nrtv:location_14;
    nrtv:related_to nrtv:related_event_2.
```

Listing 6.3: Last events related to "Pablo Picasso".

With that being said, it allows us to create a reference point when comparing the results with each other. For instance, we could deduct whether narratives with a higher event count tend to get higher scores than the ones with a lower event count. In order to establish the baseline for narrative metric evaluation, we consider a value of 3.0 for the "score" variable as the baseline, meaning that, any narrative with a score of 3.0 or above is considered a feasible narrative.

The final mean scores per narrative genre can be found in Table 6.3.

Table 6.3: Wikidata narrative scores.

	Wikidata		**DBpedia**	
	depth 1	**depth 2**	**depth 1**	**depth 2**
PE	3.67	3.20	1.80	2.00
PL	2.67	3.40	1.00	1.60
E	1.80	3.50	1.20	2.00
R	1.20	1.20	0.60	0.50
M	1.40	2.67	0.60	0.75
Avg.	2.15	2.80	1.04	1.37

For the Wikidata graph, we can see that the genres that tend to get higher scores, at depth level 1, are "People (PE)" and "Places" (PL). At depth level 2, there is a general **30.07%** increase in scores when compared to depth level 1, with the exception of narratives about "Religion" and "People".

The reason for depth affecting negatively narratives about people is due to the fact that all of the relevant information tends to be in close proximity to the main entity, in this narrative genre. Thus, by exploring the semantic graph deeper, we uncover entities and relationships that are irrelevant to the biographical narrative.

"Religions", on the other hand, show poor results for both levels of depth. This could signify that, either our approach fails to capture the religious narratives, or that the used semantic graphs are not tailored well enough to represent this type of narrative information, or both. In order to come up with an answer to this question, we would need to, manually, analyze religious data in the semantic graphs and conclude whether it is possible to develop an approach capable of extracting the respective narratives. This, however, is an avenue for future work.

As for the DBpedia, the average score per narrative genre (**1.21**) is much smaller than the one for the Wikidata (**2.47**), resulting in an overall **105.06% decrease** in scores. This is, mainly, due to the fact that DBpedia contains significantly less temporal information than Wikidata. Most of the time related information, in Wikidata, is stored in its qualifiers, which DBpedia does not possess. However, the score increase from depth level 1 to depth level 2, in DBpedia (**31.72%**), is similar to the one from Wikidata.

This score analysis leads us to conclude that our approach performs better for the "People" and "Places" narrative genres than the others. Also, there is a clear gap between the quality of results extracted from different graphs. This is due to the fact that our approach relies on the presence of temporal information that some graphs, such as DBpedia, may not be rich in. This raises the need to lift this constraint and, perhaps, develop a version that can accept information not tied, directly, to a timestamp. This could be achieved, for example, by inferring timestamps for certain entities. For instance, if there is a triple that links a person to the birth of their child, but the triple itself does not present any temporal information, we could infer the timestamp of this triple by considering the "date of birth" property of the child.

In summary, to answer the question of whether we were able to achieve our goals, our approach shows promising results for narrative extraction for narratives about people and places, while falling short for other genres. Despite our approach being able to extract narratives from any, non-specific, semantic graph, this feature required a finer tuning in order to achieve better results across different graphs.

In order to further evaluate the expressiveness of our results, we prepared more sets of metrics that were designed to evaluate each one of the following narrative components:

- Complexity

- Coherence
- Character Development
- Pacing

6.2.1.1 Baseline

In our evaluation methodology, we introduce the concept of a baseline algorithm and the corresponding baseline dataset. The inclusion of a baseline serves several crucial purposes in assessing the performance and effectiveness of our main narrative extraction approach.

The baseline algorithm represents a fundamental, rule-based approach to narrative extraction, specifically tailored for extracting narratives from Wikidata. It provides a performance benchmark against which our main approach can be evaluated. By comparing the narrative extraction results of the baseline with those of the main approach, we can assess the advantages and limitations of our primary method. This validation ensures the reliability of our narrative extraction approach.

The baseline algorithm was created by adapting the approach proposed by [5]. In our contribution to their approach, we automated the data extraction part and integrated it into our graphical user interface. The only configuration parameters that this algorithm utilizes are the topic and the depth. It relies on a set of pre-defined Wikidata properties and entity mappings to the event attributes. These mappings can be found in Table 6.4.

Table 6.4: Baseline Wikidata Mappings

Event Attribute	**Wikidata Property Identifier**
Who	wdt:P710, wdt:P664, wdt:P112
Where	wdt:P276, wdt:P17, wdt:P625, wdt:P131, wdt:P30
When	wdt:P585, wdt:P580, wdt:P582, wdt:P571, wdt:P576, wdt:P577
Causal Links	wdt:P828, wdt:P1542, wdt:P361, wdt:P156, wdt:P155

The baseline dataset includes a set of feature sand statistics that mirror those found in the main narratives dataset. These features are designed to capture various aspects of the narratives extracted by the baseline algorithm. We can utilize both, baseline and narratives datasets to compare our main approach with the baseline.

6.2.1.2 Complexity

We analyze the complexity of the extracted narratives by obtaining the following statistics from the narrative corpus:

- Number of events (NUM-E)

- Number of events with "who" attribute (NUM-WHO)
- Number of events with "where" attribute (NUM-WHERE)
- Number of events with "why" attribute (NUM-WHY)
- Number of events with "how" attribute (NUM-HOW)
- Number of events with "relatedTo" attribute (NUM-REL)
- Event completeness. Represents the ratio between the number of attributes with a value in an event and the total number of attributes. (E-COMP)
- "What" attribute word count (WHAT-WC)
- Number of unique characters (NUM-UC)
- Number of unique locations (NUM-UL)

The resulting statistics can be visualized, as mean values, in Table 6.5 for Wikidata, Table 6.6 for DBpedia, and Table 6.7 for the baseline.

Table 6.5: Wikidata narrative complexity.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NUM-E	1.6	3.0	37.0	35.5	5.0	23.33	7.67	27.0	29.6	5.0
NUM-WHO	1.2	0.0	37.0	8.33	0.0	21.0	6.0	27.0	11.8	0.0
NUM-WHERE	1.6	0.0	11.33	35.5	0.0	21.67	5.67	10.2	29.6	0.0
NUM-WHY	0.4	0.0	1.67	0.0	0.0	0.83	0.0	1.6	0.0	0.0
NUM-HOW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NUM-REL	1.6	0.0	6.17	1.0	0.0	23.33	7.67	6.0	9.6	0.0
E-COMP	0.74	0.29	0.5	0.47	0.29	0.66	0.66	0.51	0.52	0.29
WHAT-WC	12.4	4.96	5.96	5.28	5.89	9.15	3.86	6.55	6.31	5.89
NUM-UC	24.8	0.0	8.83	4.83	0.0	50.5	39.83	14.2	24.8	0.0
NUM-UL	4.4	0.0	9.5	12.33	0.0	21.17	23.5	10.2	10.2	0.0

Using these tables, we can compare our results, in more detail, across different graphs, genres, and approaches. We can see that the total number of extracted events (NUM-E), on average, differs greatly from one narrative genre to another. The genres that tend to get the highest amount of events are: "People" (PE) and "Places" (PL) for depth level 1, which coincide with the highest-rated genres according to the score analysis. This leads us to conclude, that a higher number of events is associated with higher narrative quality.

Furthermore, we can see a dramatic increase in narrative complexity, from depth level 1 and depth level 2, for the "Events" genre (E). This is a result of the algorithm being able to identify

Table 6.6: DBpedia narrative complexity.

	Depth = 1					Depth = 2				
GENRE	E	M	PE	PL	R	E	M	PE	PL	R
NUM-E	1.2	0.6	5.6	1.4	0.8	21.4	1.0	7.8	5.6	2.0
NUM-WHO	1.2	0.0	5.6	0.0	0.2	12.0	0.25	7.8	4.2	0.5
NUM-WHERE	1.2	0.0	2.8	1.4	0.2	12.8	0.25	5.0	3.8	0.5
NUM-WHY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NUM-HOW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NUM-REL	1.0	0.0	0.8	0.0	0.0	21.2	0.25	0.8	4.2	0.0
E-COMP	0.54	0.17	0.52	0.26	0.23	0.42	0.27	0.53	0.41	0.18
WHAT-WC	4.4	0.6	1.72	0.83	0.6	3.04	1.25	2.29	2.17	0.5
NUM-UC	15.2	0.0	4.6	0.0	0.6	16.8	2.75	5.4	11.2	1.5
NUM-UL	4.4	0.0	2.4	0.6	0.2	9.6	0.75	3.0	4.4	0.5

Table 6.7: Baseline narrative complexity.

	depth = 1					depth = 2				
GENRE	E	M	PE	PL	R	E	M	PE	PL	R
NUM-E	23.4	10.6	106.17	190.83	32.8	695.0	47.17	229.4	180.2	100.4
NUM-WHO	6.8	1.4	61.5	72.5	8.0	155.17	13.5	82.8	95.8	19.0
NUM-WHERE	20.0	8.8	89.67	186.67	18.8	524.5	35.67	215.0	167.6	50.8
NUM-WHY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NUM-HOW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NUM-REL	16.6	6.6	30.5	97.67	16.8	535.67	24.33	146.2	109.2	57.4
E-COMP	0.43	0.37	0.43	0.34	0.33	0.44	0.37	0.43	0.38	0.34
WHAT-WC	5.5	2.43	4.42	3.56	2.33	4.06	2.61	3.87	3.55	2.61
NUM-UC	11.6	5.8	164.83	343.5	15.4	238.67	39.17	285.6	401.8	37.2
NUM-UL	6.2	10.2	18.17	40.17	26.8	350.83	38.67	106.2	48.0	65.0

more events connected to the main one as it searches deeper into the graph, thus, also, the dramatic increase in the number of related events (NUM-REL) for the "Events" genre (E).

It is also worth noting that, unlike in Wikidata, in DBpedia, there were narratives that contained no events, hence the average number of events, for the "Movement" (M) and the "Religion" (R) genres, being below 1.0. Once again, this phenomenon is a result of our algorithm only considering data with a timestamp, which may not always be present.

The most common event attributes are: "who" (NUM-WHO) and "where" (NUM-WHERE), independently of genre, depth, and graph. The attributes "Why" (NUM-WHY) and "How" (NUM-HOW) tend to show up the least. This discrepancy can be a result of entity misclassification from the part of, either the algorithm or the user if the classification was performed manually. Although, generally, more descriptive information, such as the one defined by the "Why" and

"How" attributes, tends to be omitted in traditional semantic graphs, which prioritize semantic information over descriptiveness.

The mean event completeness metric (E-COMP) summarizes the overall event completeness for each narrative genre. It shows that the most complete narratives are the ones that belong to the "Events" (E) genre, regardless of the graph. This proves that "Event" narratives, despite not always including certain attributes, tend to be, overall, the ones with the most attributes.

Furthermore, we showcase the word count for the "What" attribute, since it is a required attribute. This metric is meant to display the overall detail and descriptiveness of the events in the narrative. The most detailed events are, once again, from the "Events" (E) genre, leading us to believe that it may not be correlated to the quality of the narrative, judging by the score analysis.

The number of unique characters (NUM-UC) and locations (NUM-UL) attributes further display the complexity of narratives.

When compared to the baseline results in Table 6.7, our results show a significantly lower overall number of events. However, our results demonstrate higher event completeness, meaning that, despite having fewer events, our narratives have more detailed and complete event representations than the ones produced by the baseline algorithm.

6.2.1.3 Coherence

We analyze the coherence of the extracted narratives by obtaining the following statistics from the narrative corpus:

- Number of event chains (NUM-EC)
- Length of the largest event chain (LONG-EC)
- Length of the shortest event chain (SHORT-EC)
- Average event chain length (EC-LEN)

An event chain is defined as a set of events that appear in a sequence on the narrative timeline and have at least one character of place in common. This allows us to measure the flow of the narrative, by analyzing how events are connected, within the narrative. The resulting statistics can be visualized, as mean values, in Table 6.8 for Wikidata, Table 6.9 for DBpedia, and Table 6.10 for the baseline.

For depth level 1, most of the genres tend to have only one event chain, that spans out from the first event to the last one. The number of event chains should be evaluated in relation to the total number of events in the narrative. For example, if there is a low number of event chains in long narratives, it means that there is not much variability when it comes to trends, since almost

Table 6.8: Wikidata narrative coherence.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NUM-EC	1.0	3.0	1.0	1.0	5.0	5.83	5.5	1.0	1.0	5.0
LONG-EC	1.6	1.0	37.0	35.5	1.0	14.5	3.0	27.0	29.6	1.0
SHORT-EC	1.6	1.0	37.0	35.5	1.0	11.33	1.17	27.0	29.6	1.0
AVG-ECLEN	1.6	1.0	37.0	35.5	1.0	12.13	1.63	27.0	29.6	1.0

Table 6.9: DBpedia narrative coherence.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NUM-EC	0.8	0.6	1.0	0.6	0.8	10.6	1.0	1.0	3.0	2.0
LONG-EC	1.2	0.6	5.6	1.4	0.6	5.6	0.75	7.8	2.4	0.5
SHORT-EC	1.2	0.6	5.6	1.4	0.6	1.0	0.75	7.8	1.2	0.5
AVG-ECLEN	1.2	0.6	5.6	1.4	0.6	1.29	0.75	7.8	1.4	0.5

all of the events in the narrative have at least one element in common. On the other hand, short narratives with multiple event chains indicate the disconnect between the events and the lack of any major trends. Ideally, we would strike a balance between the number of events and the number of event chains.

We can calculate the ratio between the number of event chains and the number of events in the narrative in order to assess the coherence of a narrative, where lower values would indicate onesided events with little diversity, and high values would indicate narratives that lack logical connections within them.

For example, in the Wikidata graph, the highest event chain ratio (1.0) belongs to the "Religion" (R) genre. Meanwhile, the lowest ratio (0.031) belongs to the "Places" (PL) genre. The genre that showed the most balanced event chain ratio (0.8) is the "Event" (E) genre. Similar results can be observed in the DBpedia narratives.

The baseline narratives tend to have more balanced event chain ratios, as can be seen in Table 6.10, than the narratives produced by our method.

6.2.1.4 Character Development

We analyze the character development in the extracted narratives by obtaining the following statistics from the narrative corpus:

- Main character presence (MC-PRES). Calculated as:

$$mc\_pres = n_{mc}/n_e$$

Table 6.10: Baseline narrative coherence.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NUM-EC	6.4	9.6	26.17	8.17	27.2	528.33	40.5	65.2	44.8	91.4
LONG-EC	14.8	1.8	66.67	96.17	5.0	30.17	3.33	72.2	26.4	5.2
SHORT-EC	1.0	1.0	1.0	0.67	1.0	1.0	1.0	1.0	0.8	1.0
AVG-ECLEN	3.16	1.14	6.54	16.98	1.36	2.39	1.19	2.73	2.53	1.14

- Side character presence (SC-PRES). Calculated as:

$$sc\_pres = n_{sc}/n_e$$

- Main character relevance (MC-REL). Calculated as:

$$mc\_rel = n_{mc} - n_{mc2}$$

- Character persistence (CH-PERS). Calculated as:

$$ch\_pers = (\sum_{c \subset ch} cons_e(c))/n_e$$

Where:

n_{mc} is the number of events that include the most reoccurring (main) character

n_{mc2} is the number of events that include the second most reoccurring character

n_e is the total number of events

n_{sc} is the number of events that include characters besides the main one

$cons_e(c)$ is the number of consecutive events that include character c

The resulting statistics can be visualized, as mean values, in Table 6.11 for Wikidata, Table 6.12 for DBpedia, and Table 6.13 for the baseline.

Table 6.11: Wikidata character development.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
MC-PRES	0.8	0.0	1.0	0.11	0.0	0.28	0.43	1.0	0.19	0.0
SC-PRES	0.6	0.0	0.34	0.14	0.0	0.71	0.83	0.41	0.32	0.0
MC-REL	0.4	0.0	33.17	1.17	0.0	3.0	0.17	28.8	5.4	0.0
CH-PERS	1.03	0.0	2.36	0.73	0.0	1.25	1.07	1.83	0.97	0.0

Character development metrics aim to evaluate how characters are utilized within the narrative. The main character presence indicates whether the narrative contains a main character, which would be indicated by the high values of the main character presence.

Table 6.12: DBpedia character development.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
MC-PRES	0.8	0.0	1.0	0.0	0.2	0.31	0.12	1.0	0.3	0.12
SC-PRES	0.8	0.0	0.15	0.0	0.2	0.32	0.12	0.21	0.37	0.12
MC-REL	0.4	0.0	4.4	0.0	0.0	5.4	0.25	10.0	0.6	0.0
CH-PERS	0.88	0.0	1.81	0.0	0.2	1.06	0.27	1.73	0.41	0.5

Table 6.13: Baseline character development.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
MC-PRES	0.08	0.07	0.36	0.08	0.08	0.02	0.06	0.28	0.1	0.03
SC-PRES	0.21	0.11	0.23	0.27	0.16	0.12	0.25	0.19	0.27	0.17
MC-REL	0.0	0.0	54.83	2.33	0.6	30.0	3.0	65.0	9.2	13.0
CH-PERS	0.72	0.6	1.32	0.88	1.22	2.68	1.46	1.48	1.52	2.08

In order to contrast the main character, side character presence metrics aim to provide an understanding of the presence of all non-main characters, in relation to the main one. Biographical narratives tend to focus, primarily, on a single character, leaving little room for other characters.

Main character relevancy measures the distance from the main character to the second most reoccurring character, in terms of relevancy and presence.

Finally, character persistence calculates how long, on average, a character persists within an event chain.

In these tables, we can see that "People" (PE) narratives dominate character development, which is the result we were expecting to achieve since this genre focuses on detailed different parts of the life of a person. In this regard, we can conclude that we are able to extract meaningful biographical narratives using our approach.

Overall, both graphs show similar results, regardless of depth, which could indicate that our approach was able to provide the same quality of character development, regardless of the input graph.

In the baseline results, the character persistence and the main character presence have lower values for the biography genres (PE) than the ones from our results. This leads us to conclude that our approach outperforms the baseline algorithm when it comes to biographical narrative extraction.

6.2.1.5 Pacing

We analyze the pacing of the narratives by obtaining the following statistics from the narrative corpus:

- Narrative duration in years (NARR-DUR)
- Time standard deviation in years (TIME-STD)

The resulting statistics can be visualized, as mean values, in Table 6.14 for Wikidata, Table 6.15 for DBpedia, and Table 6.16 for the baseline.

Table 6.14: Wikidata pacing.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NARR-DUR	0.68	7.22	6.7	7.82	32.96	3.05	5.92	7.74	9.38	32.96
TIME-STD	0.0	39.54	18.51	30.56	289.87	12.91	17.68	20.69	16.91	289.8

Table 6.15: DBpedia pacing.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NARR-DUR	0.0	0.0	2.52	0.57	0.01	0.56	1.81	2.52	0.64	0.0
TIME-STD	0.0	0.0	10.88	0.44	0.0	1.18	0.0	8.21	0.18	0.93

Table 6.16: Baseline pacing.

	depth = 1					depth = 2				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
NARR-DUR	0.0	0.0	0.1	4.65	0.0	0.0	0.0	0.0	0.61	0.0
TIME-STD	2.6	0.0	6.82	4.27	44.13	9.36	6.98	6.11	5.61	136.61

The narrative pacing pays respect to the temporal pacing of the narrative. It helps us understand the event flow within the narrative. The goal is to minimize the time standard deviation, which corresponds to narratives with less time gaps.

The null narrative duration and time standard deviation may be indicative of narratives that started and ended on the same date. The baseline algorithm has produced a large number of narratives with a duration of 0. This is due to the fact that, unlike our approach, the baseline algorithm does not consider only temporal information when building a narrative. Therefore, it is impossible to infer the duration of a narrative if one of the events within the narrative does not have a timestamp since we can not place it on the timeline.

For DBpedia, we managed to achieve reasonable results, with the highest time standard deviation (TIME-STD) being 10.88 for the "Person" (PE) genre. Excluding the values of 0, our DBpedia results outperform the baseline results in most of the narrative genres.

On the other hand, for Wikidata, the results were less satisfactory, as they contain, generally, high time standard deviation (TIME-STD) values, across all genres and depths.

In conclusion, we performed an in-depth exploratory analysis of the extracted narratives and concluded that our approach presents promising results when compared to the baseline. However, it still fails to capture, equally, every narrative genre, from any graph. For future work, we could explore the option to remove the timestamp constraint from our approach in order to allow more information within our narratives.

6.2.2 Events Evaluation

The target variable used for event evaluation is "relevance". Relevance is a binary variable indicating whether the event is relevant or not to the respective, narrative. It was manually assigned by the authors based on their personal judgment, therefore, as a disclaimer, this data is not 100% correct. The judgment was made based on the research on the topic of the event, which led us to decide whether an event belonged to a narrative or not. Other criteria used for selecting relevant events were whether they represent any meaningful information, such as birth or proclamation of Independence, or information that does not add anything to the narrative, such as mentions of the topic in different sources. Nevertheless, a number of metrics and analysis techniques can be performed based on this variable, such as comparing the impact of each variable on the "relevance" of the event.

Furthermore, we manually calculated the number of correctly assigned attributes for each event. This should give us insight into the classification and attribute mapping steps of our approach. The events were evaluated independently of the original graph and the respective results can be found in Table 6.17.

Table 6.17: Event relevancy and attribute correctness.

	depth = 1					**depth = 2**				
GENRE	**E**	**M**	**PE**	**PL**	**R**	**E**	**M**	**PE**	**PL**	**R**
ATTRIBUTE CORRECTNESS	0.91	0.95	0.99	1.0	1.0	1.0	0.97	0.99	1.0	1.0
RELEVANCY	1.0	0.56	0.75	0.69	0.31	0.96	1.0	0.76	0.79	0.31

The attribute correctness represents the percentage of attributes that were correctly classified and allocated to the events. Our results show extremely positive attribute correctness, across all genres and depths. This indicates that our classification and attribute mapping systems were correctly selected and implemented.

Relevancy, on the other hand, shows a bigger disparity across different narrative genres. It represents the percentage of events that are relevant to their narratives. The lowest relevancy was achieved for the "Religion" (R) genre, with only 31% of the event being relevant. The highest relevancy was obtained for the "Event" (E) genre, with a small decrease of 4% from depth level 1 to depth level 2.

Overall, there is a direct increase in relevancy going from depth level 1 to depth level 2. This could indicate that the depth parameter was correctly implemented, as it further enriches the narratives with more relevant data.

In conclusion, in this section, we tried answering the research question **Q4**, by presenting our experimental methodology for evaluating our approach for narrative extraction from semantic graphs. In order to achieve that, we extracted a total of 100 narratives, from different semantic graphs and genres and for different depth levels. We, also, developed a baseline algorithm based on the approach proposed by Blin [5] for narrative extraction from Wikidata. We analyzed the results by devising different metrics and statistics, grouped them into different categories, and compared the results across different graphs, genres, depth levels, and baseline results for the same narratives. Our analysis, despite being experimental in nature and not totally reliable, showed promising results for the biography and places narratives, outperforming the baseline. As for the other genres, the movement and religion genres showed poor results, overall. We concluded that the depth parameter plays a vital role in enhancing the richness and completeness of the narratives. In our experiments, Wikidata showed more positive results than DBpedia, thus motivating us to develop a new version of our approach where we remove the constraint of having timestamps as a requirement for narrative building.

Chapter 7

Conclusions

The aim of this thesis has been to explore the world of narrative extraction from semantic graphs and find out whether we could implement our own narrative extraction tool. This goal was later split into four key research questions:

- **Q1.** What is a narrative and how can we represent it using semantic graphs?
- **Q2.** How to identify information within a semantic graph relevant to a narrative?
- **Q3.** How can we build a narrative using information found in the semantic graph?
- **Q4.** How can we validate our work ?

Each research question aims to fill in a major part of the narrative extraction process and was answered throughout this thesis.

Question **Q1** was answered by the development of an ontology1 to represent narratives in Resource Description Framework (RDF) format. This ontology aims to provide a formal and standardized representation of narratives and their events.

In this thesis, we proposed Narrative Extractor from Semantic Graphs (NESG)12, an approach designed to extract narratives from any semantic graph. To accomplish this, it utilizes graph configurations that promote higher adaptability to different domains and vocabularies. This step of our approach serves to partially answer the research question **Q2**. After configuring the graph, NESG can be utilized to extract narratives from it according to user-pre-defined criteria. It relies on a user-provided topic as input, indicating the central theme of the narrative. This topic is then associated with an entity within the graph, which is marked as the main entity in the narrative. After acquiring the main entity, our approach proceeds by extracting all relevant entities connected to the main one through semantic links, using Breadth-first search (BFS), a graph search algorithm. By this point, we have already answered the question **Q2**.

Once all of the relevant information is acquired, it needs to be sorted and re-utilized into something useful. To do that, we begin by identifying events within these entities. Our methodology considers two types of events: entity events (explicit) and property events (implicit), depending on how they were identified. Using these events as building blocks, we map their attributes to the remaining of the extracted data, forming complete event representations. These events are linked and a narrative is created, which is then converted to the Turtle (.ttl) format, answering the research question **Q3**. In summary, our approach takes a graph configuration and a topic string as input and produces one narrative for each topic, in Turtle format.

This theoretical approach was implemented as a web graphical user interface, supported by `Node.js` to host a server on a remote machine. The server acts as the central point of the system, providing communication tools with the graph endpoint and data storage. The graph configuration data, as well as graph cached data, such as entity classes and property labels, are stored in a JSON file on the server machine. During implementation, we encountered several challenges, most of which were related to querying semantic graphs. We overcame these challenges by utilizing techniques such as batch querying and modular queries, which significantly improved the performance and consistency of our tool.

After developing and implementing NESG, we performed an experimental evaluation of our approach. We began by highlighting the main challenges and obstacles related to story and narrative evaluation. To address some of those challenges, we attempted to implement both quantitative and qualitative metrics into our evaluation methodology. We performed a data extraction task in which we extracted a total of 100 narratives[1], from 2 separate graphs (Wikidata and DBpedia), which comprise 25 unique topics, categorized into one of the following genres: "Person", "Place", "Event", "Religion", and "Movement". Each topic was extracted twice, one for depth level 1 and the other for depth level 2, making it a total of two narratives per topic. Our goal was to evaluate the full spectrum of narratives, across different domains, genres, and depth levels, to assess the adaptability of our approach.

We separated the evaluation methodology into two steps: narrative evaluation and event evaluation. Narrative evaluation allowed us to evaluate narratives as a whole, by comparing a variety of metrics and statistics between graphs, genre, and depth levels. We discovered that our approach shows significantly better results for the "Person" and "Place" genres when compared to other genres. Furthermore, the results from Wikidata presented especially higher quality than the ones from DBpedia, hindering the adaptability of our approach across different graphs.

Individual events were evaluated to better assess the specific parts of our approach, namely the event identification and attribute mapping steps. Our event-building mechanism showed promising results, although more rigorous testing is required. While our preliminary results showcase the potential of our approach, there are still several avenues for future work to further enhance and extend our method.

During the development, we participated in the Symposium on Languages, Applications, and Technology (SLATE) conference, by submitting and presenting a paper about our work in its current stage. The paper can be found in [22].

7.1 Future Work

One direction for future work is the integration of graph-based algorithms, such as graph embedding and graph neural networks. These techniques can enhance narrative understanding by capturing the structural and semantic relationships within the graph. By incorporating graph-based algorithms, we can leverage the inherent graph topology and semantics to improve event extraction, attribute mapping, and narrative construction.

The web interface, as well as the NESG system as a whole, requires further evaluation to ensure stability and consistency. Particularly, the data storage, which is, currently, implemented as a single JSON file, is unviable and needs a replacement, preferably, using a database. Security could also be improved dramatically.

Another area of future exploration is the development of machine learning models for property and entity classification. The large amount of data available in knowledge graphs can be leveraged to train robust models that can accurately classify entities and properties. By employing machine learning techniques, we can automate the assignment of entity and property classes, reducing the reliance on manual user configuration and improving the accuracy of the classification process.

As an alternative way of representing narratives, we could develop a method that converts narratives in RDF format to natural language text. This, however, is no simple task and, in order to achieve high-quality results, it would require significant work dedicated to it. This could improve the readability of our narratives, as well as provide more robust methods for narrative evaluation.

To ensure the generalizability and applicability of our approach, it is crucial to evaluate it on other benchmark graphs and real-world datasets. This evaluation will allow us to assess the adaptability and robustness of our approach across different knowledge domains and data characteristics. By testing our method on diverse datasets, we can identify potential challenges and further refine the algorithms to handle real-world complexities.

Significant strides need to be made in the field of narrative evaluation, as our proposed evaluation methodology is merely experimental and does not provide fool-proof results. Narratives are subjective in nature, thus requiring manual evaluation from a large group of people in order to define a consensus for narrative quality. In the future, we could look into hiring individuals to assess the narratives produced by our approach. For that, we could utilize platforms such as Amazon Mechanical Turk.

In summary, in this thesis the presented NESG, a narrative extraction pipeline, which was implemented using `Node.js`, as well as an ontology for representing narratives. We evaluated our approach on several narratives and concluded that, despite certain failures, our approach has demonstrated satisfactory results towards biographical and location narrative extraction from Wikidata. We hope that our work will inspire and guide future research in this exciting field, fostering advancements in narrative extraction and semantic graph analysis.

Bibliography

[1] Ali Al-Jafar and Cary A Buzzelli. The art of storytelling for cross cultural understanding. *International Journal of Early Childhood*, 36:35–48, 2004.

[2] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. TimeMachine. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2015. doi:10.1145/2783258.2783325.

[3] Nandini Anantharama, Simon Angus, and Lachlan O'Neill. CANarEx: Contextually aware narrative extraction for semantically rich text-as-data applications. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3551–3564, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.findings-emnlp.260.

[4] Aman Berhe. *Extraction of Narrative Structure from TV Series. (Extraction de la structure narrative de séries TV)*. PhD thesis, University of Paris-Saclay, France, 2021.

[5] Inès Blin. Building narrative structures from knowledge graphs. In Paul Groth, Anisa Rula, Jodi Schneider, Ilaria Tiddi, Elena Simperl, Panos Alexopoulos, Rinke Hoekstra, Mehwish Alam, Anastasia Dimou, and Minna Tamper, editors, *The Semantic Web: ESWC 2022 Satellite Events - Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13384 of *Lecture Notes in Computer Science*, pages 234–251. Springer, 2022. doi:10.1007/978-3-031-11609-4_38.

[6] Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5794–5836, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[7] Paolo Ciancarini, Mirko Farina, Ozioma Okonicha, Marina Smirnova, and Giancarlo Succi. Software as storytelling: A systematic literature review. *Computer Science Review*, 47: 100517, 2023. ISSN: 1574-0137. doi:https://doi.org/10.1016/j.cosrev.2022.100517.

[8] Ruth Davidhizar and Giny Lonser. Storytelling as a teaching technique. *Nurse educator*, 28 (5):217–221, 2003.

[9] Michelle Edwards, Jonathan Tuke, Matthew Roughan, and Lewis Mitchell. The one comparing narrative social network extraction techniques. In Martin Atzmüller, Michele Coscia, and Rokia Missaoui, editors, *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020, The Hague, Netherlands, December 7-10, 2020*, pages 905–913. IEEE, 2020. doi:10.1109/ASONAM49781.2020.9381346.

[10] Wenfei Fan, Ruochun Jin, Ping Lu, Chao Tian, and Ruiqi Xu. Towards event prediction in temporal graphs. *Proc. VLDB Endow.*, 15(9):1861–1874, 2022.

[11] John Feliks. A prehistory of hiking—neanderthal storytelling. *Pleistocene Coalition News*, 3 (2):1–2, 2011.

[12] Yiannis Gabriel et al. Narratives, stories and texts. *The Sage handbook of organizational discourse*, (Part I):61–78, 2004.

[13] Simon Gottschalk and Elena Demidova. Eventkg: A multilingual event-centric temporal knowledge graph. In *Proceedings of the Extended Semantic Web Conference (ESWC 2018)*. Springer, 2018.

[14] Simon Gottschalk and Elena Demidova. EventKG - the Hub of Event Knowledge on the Web - and Biographical Timeline Generation. volume 10, pages 1039–1070. IOS Press, 2019.

[15] Trisha Greenhalgh and Brian Hurwitz. Why study narrative? *Bmj*, 318(7175):48–50, 1999.

[16] Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. What is event knowledge graph: A survey, 2022.

[17] Zepeng Huai, Dawei zhang, Guohua Yang, and Jianhua Tao. Spatial-temporal knowledge graph network for event prediction. *Neurocomputing*, 553:126557, 2023. ISSN: 0925-2312. doi:https://doi.org/10.1016/j.neucom.2023.126557.

[18] Elwin Huaman. Steps to knowledge graphs quality assessment. *CoRR*, abs/2208.07779, 2022. doi:10.48550/arXiv.2208.07779.

[19] Muhammad Nihal Hussain, Hayder Al Rubaye, Kiran Kumar Bandeli, and Nitin Agarwal. Stories from blogs: Computational extraction and visualization of narratives. In Ricardo Campos, Alípio Mário Jorge, Adam Jatowt, Sumit Bhatia, and Mark A. Finlayson, editors, *Proceedings of Text2Story - Fourth Workshop on Narrative Extraction From Texts held in conjunction with the 43rd European Conference on Information Retrieval (ECIR 2021), Lucca, Italy, April 1, 2021 (online event due to Covid-19 outbreak)*, volume 2860 of *CEUR Workshop Proceedings*, pages 33–40. CEUR-WS.org, 2021.

[20] I-Ta Lee, Maria Leonor Pacheco, and Dan Goldwasser. Modeling human mental states with an entity-based narrative graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4916–4926, Online, June 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.naacl-main.391.

[21] O-Joun Lee and Jason J. Jung. Integrating character networks for extracting narratives from multimodal data. *Information Processing Management*, 56(5):1894–1923, 2019. ISSN: 0306-4573. doi:https://doi.org/10.1016/j.ipm.2019.02.005.

[22] Daniil Lystopadskyi, André Santos, and José Paulo Leal. Narrative extraction from semantic graphs (short paper). In Alberto Simões, Mario Marcelo Berón, and Filipe Portela, editors, *12th Symposium on Languages, Applications and Technologies, SLATE 2023, June 26-28, 2023, Vila do Conde, Portugal*, volume 113 of *OASIcs*, pages 9:1–9:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/OASIcs.SLATE.2023.9.

[23] María Miró Maestre, Marta Esther Vicente, Elena Lloret, and Armando Suárez Cueto. Extracting narrative patterns in different textual genres: A multilevel feature discourse analysis. *Inf.*, 14(1):28, 2023. doi:10.3390/info14010028.

[24] Sorin Adam Matei and Lucas Hunter. Data storytelling is not storytelling with data: A framework for storytelling in science communication and data journalism. *The Information Society*, 37(5):312–322, 2021. doi:10.1080/01972243.2021.1951415.

[25] Karine Megerdoomian, Karl Branting, Charles Horowitz, Amy Marsh, Stacy Petersen, and Eric Scott. Automated narrative extraction from administrative records. In Luther Karl Branting, editor, *Proceedings of the Workshop on Artificial Intelligence and the Administrative State co-located with 17th International Conference on AI and Law (ICAIL 2019), Montreal, QC, Canada, June 17, 2019*, volume 2471 of *CEUR Workshop Proceedings*, pages 38–48. CEUR-WS.org, 2019.

[26] Daniele Metilli. *Enhancing the Computational Representation of Narrative and Its Extraction from Text*. PhD thesis, University of Pisa, Italy, 2021.

[27] Brian Keith Norambuena and Tanushree Mitra. Narrative maps: An algorithmic approach to represent and extract information narratives. *Proc. ACM Hum. Comput. Interact.*, 4 (CSCW3):1–33, 2020. doi:10.1145/3432927.

[28] Andrew Piper, Richard Jean So, and David Bamman. Narrative theory for computational narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.emnlp-main.26.

[29] Chia Yi Quah and Kher Hui Ng. A systematic literature review on digital storytelling authoring tool in education: January 2010 to january 2020. *International Journal of Human–Computer Interaction*, 38(9):851–867, 2022. doi:10.1080/10447318.2021.1972608.

[30] Priyanka Ranade, Sanorita Dey, Anupam Joshi, and Tim Finin. Computational understanding of narratives: A survey. *IEEE Access*, 10:101575–101594, 2022. doi:10.1109/ACCESS.2022.3205314.

[31] Priyanka Ranade, Sanorita Dey, Anupam Joshi, and Tim Finin. Computational understanding of narratives: A survey. *IEEE Access*, 10:101575–101594, 2022. doi:10.1109/ACCESS.2022.3205314.

[32] Krishna Ronanki, Beatriz Cabrero-Daniel, and Christian Berger. Chatgpt as a tool for user story quality evaluation: Trustworthy out of the box?, 2023.

[33] Brenda Salenave Santana, Ricardo Campos, Evelin Amorim, Alípio Jorge, Purificação Silvano, and Sérgio Nunes. A survey on narrative extraction from textual data. *Artif. Intell. Rev.*, 56(8):8393–8435, 2023. doi:10.1007/s10462-022-10338-7.

[34] Olivier Serrat. Storytelling. 2008.

[35] Xurong Shan, Dawei Wang, and Jiaxiang Li. Research on data storytelling strategies for cultural heritage transmission and dissemination. In Constantine Stephanidis, Margherita Antona, Stavroula Ntoa, and Gavriel Salvendy, editors, *HCI International 2022 – Late Breaking Posters*, pages 344–353, Cham, 2022. Springer Nature Switzerland.

[36] Swapna Somasundaran, Brian Riordan, Binod Gyawali, and Su-Youn Yoon. Evaluating argumentative and narrative essays using graphs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1568–1578, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[37] Swapna Somasundaran, Michael Flor, Martin Chodorow, Hillary Molloy, Binod Gyawali, and Laura McCulla. Towards evaluating narrative quality in student writing. *Transactions of the Association for Computational Linguistics*, 6:91–106, 2018. doi:10.1162/tacl$_{a0}$0007.

[38] Jialong Tang, Hongyu Lin, Meng Liao, Yaojie Lu, Xianpei Han, Le Sun, Weijian Xie, and Jin Xu. From discourse to narrative: Knowledge projection for event relation extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 732–742. Association for Computational Linguistics, 2021. doi:10.18653/v1/2021.acl-long.60.

[39] Karin Thier, Russin, and Mahagaonkar. *Storytelling in Organizations.* Springer, 2018.

Appendix A

NESG Page Example

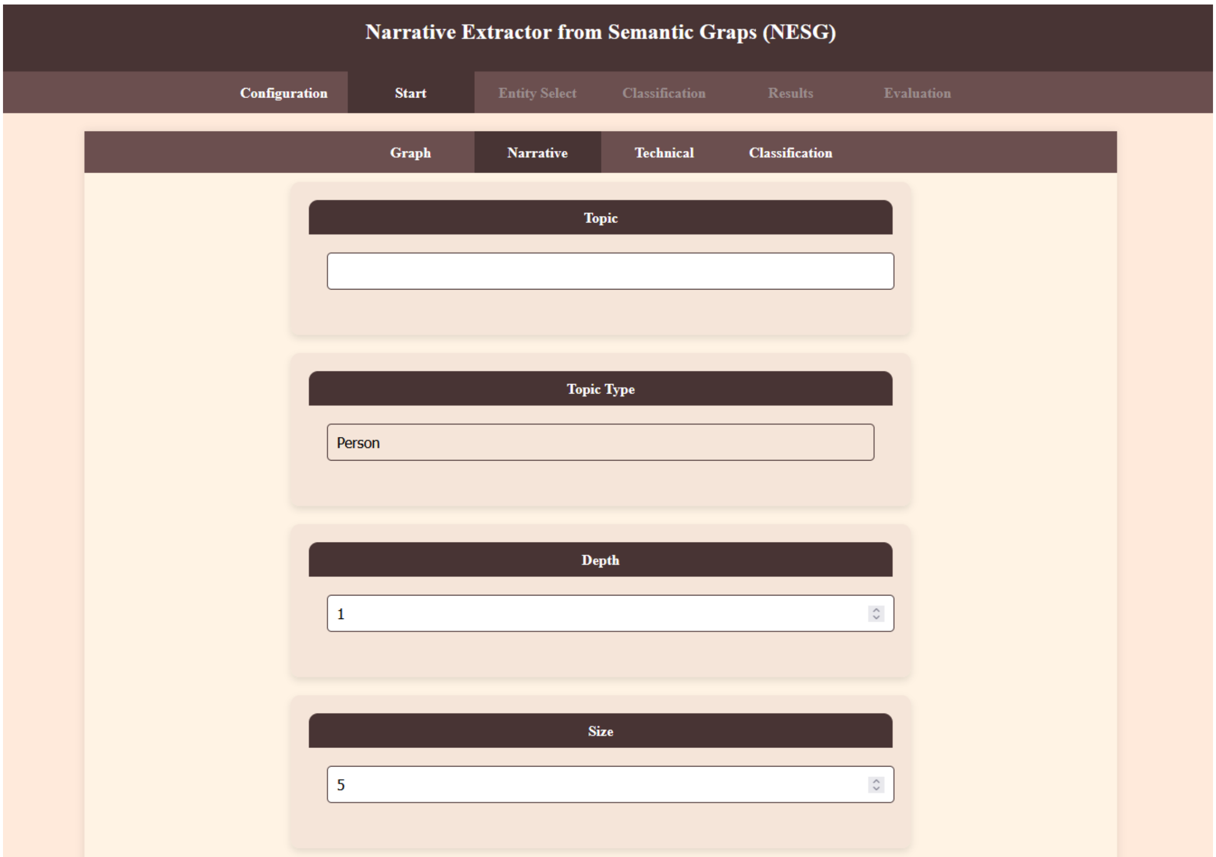


Figure A.1: NESG Page Example.

Appendix B

Wembley Stadium Narrative

```
narrative_0 a nrtv:Narrative;
    nrtv:about "Wembley Stadium";
    nrtv:hasTheme http://www.wikidata.org/entity/Q74539696;
    nrtv:hasTheme  http://www.wikidata.org/entity/Q19860854;
    nrtv:eventSequence event_sequence_0.

event_sequence_0 a rdf:Seq;
    rdf:_1 event_1;
    rdf:_2 event_2;
    rdf:_3 event_3;
    rdf:_4 event_4;
    rdf:_5 event_5;
    rdf:_6 event_6;
    rdf:_7 event_7;
    rdf:_8 event_8;
    rdf:_9 event_9;
    rdf:_10 event_10;
    rdf:_11 event_11;
    rdf:_12 event_12.

event_1 a nrtv:Event;
    nrtv:what "date of official opening";
    nrtv:when "1923-04-28T00:00:00Z";
    nrtv:where nrtv:location_0.

event_2 a nrtv:Event;
    nrtv:what "owned by start time";
    nrtv:when "1924-01-01T00:00:00Z";
    nrtv:who nrtv:character_0;
```

```
    nrtv:where nrtv:location_1;
    nrtv:where nrtv:location_2.

event_3 a nrtv:Event;
    nrtv:what "significant event point in time";
    nrtv:when "1966-01-01T00:00:00Z";
    nrtv:where nrtv:location_3;
    nrtv:relatedTo nrtv:related_event_0.

event_4 a nrtv:Event;
    nrtv:what "significant event point in time";
    nrtv:when "1992-01-01T00:00:00Z";
    nrtv:where nrtv:location_4;
    nrtv:relatedTo nrtv:related_event_1.

event_5 a nrtv:Event;
    nrtv:what "significant event point in time";
    nrtv:when "1995-01-01T00:00:00Z";
    nrtv:where nrtv:location_5;
    nrtv:relatedTo nrtv:related_event_2.

event_6 a nrtv:Event;
    nrtv:what "significant event point in time";
    nrtv:when "1996-01-01T00:00:00Z";
    nrtv:where nrtv:location_6;
    nrtv:relatedTo nrtv:related_event_3.

event_7 a nrtv:Event;
    nrtv:what "owned by end time";
    nrtv:when "1999-01-01T00:00:00Z";
    nrtv:who nrtv:character_1;
    nrtv:where nrtv:location_7;
    nrtv:where nrtv:location_8.

event_8 a nrtv:Event;
    nrtv:what "owned by start time";
    nrtv:when "1999-01-01T00:00:00Z";
    nrtv:who nrtv:character_2;
    nrtv:where nrtv:location_9;
    nrtv:where nrtv:location_10.

event_9 a nrtv:Event;
```

```
    nrtv:what "date of official closure";
    nrtv:when "2000-10-07T00:00:00Z";
    nrtv:where nrtv:location_11.

event_10 a nrtv:Event;
    nrtv:what "owned by end time";
    nrtv:when "2002-01-01T00:00:00Z";
    nrtv:who nrtv:character_3;
    nrtv:where nrtv:location_12;
    nrtv:where nrtv:location_13.

event_11 a nrtv:Event;
    nrtv:what "dissolved, abolished or demolished date";
    nrtv:when "2002-09-30T00:00:00Z";
    nrtv:where nrtv:location_14.

event_12 a nrtv:Event;
    nrtv:what "on focus list of Wikimedia project
       Wikipedia:Vital articles/Level/4 point in time";
    nrtv:when "2022-10-31T00:00:00Z";
    nrtv:where nrtv:location_15.

character_0 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q9500;
    nrtv:hasRole http://www.wikidata.org/prop/P127.

character_1 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q9500;
    nrtv:hasRole http://www.wikidata.org/prop/P127.

character_2 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q107370446;
    nrtv:hasRole http://www.wikidata.org/prop/P127.

character_3 a nrtv:Character;
    nrtv:subjectOf http://www.wikidata.org/entity/Q107370446;
    nrtv:hasRole http://www.wikidata.org/prop/P127.

location_0 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.
```

```
location_1 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_2 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q128468;
    nrtv:hasSetting http://www.wikidata.org/prop/direct/P1366.

location_3 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_4 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_5 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_6 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_7 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_8 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q128468;
    nrtv:hasSetting http://www.wikidata.org/prop/direct/P1366.

location_9 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_10 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q128468;
    nrtv:hasSetting http://www.wikidata.org/prop/direct/P1366.

location_11 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
```

```
    nrtv:hasSetting origin.

location_12 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_13 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q128468;
    nrtv:hasSetting http://www.wikidata.org/prop/direct/P1366.

location_14 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

location_15 a nrtv:Location;
    nrtv:subjectOf http://www.wikidata.org/entity/Q43279;
    nrtv:hasSetting origin.

related_event_0 a nrtv:RelatedEvent;
    nrtv:subjectOf http://www.wikidata.org/entity/Q134202;
    nrtv:hasRelation http://www.wikidata.org/prop/P793.

related_event_1 a nrtv:RelatedEvent;
    nrtv:subjectOf http://www.wikidata.org/entity/Q22098085;
    nrtv:hasRelation http://www.wikidata.org/prop/P793.

related_event_2 a nrtv:RelatedEvent;
    nrtv:subjectOf http://www.wikidata.org/entity/Q3000715;
    nrtv:hasRelation http://www.wikidata.org/prop/P793.

related_event_3 a nrtv:RelatedEvent;
    nrtv:subjectOf http://www.wikidata.org/entity/Q180563;
    nrtv:hasRelation http://www.wikidata.org/prop/P793.
```

Listing B.1: "Wembley Stadium" Narrative.