IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 November 2023, accepted 11 January 2024, date of publication 31 January 2024, date of current version 18 March 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3360480

RESEARCH ARTICLE

Exploring Sentence Parsing: OpenAI API-Based and Hybrid Parser-Based Approaches

WALELIGN TEWABE SEWUNETIE[1,2] AND LÁSZLÓ KOVÁCS[1]
[1]Institute of Informatics, University of Miskolc, 3515 Miskolc, Hungary
[2]Institute of Technology, Debre Markos University, 269 Debre Markos, Ethiopia
Corresponding author: Walelign Tewabe Sewunetie (waleligntewabe@gmail.com)

This work was supported in part by the European Union, and in part by the European Social Fund.

ABSTRACT This study focuses on the fundamental process of parsing sentences to create semantic graphs from textual documents. It introduces novel techniques for parsing phrases within semantic graph-based induction, employing both ChatGPT-based and Hybrid parser-based approaches. Through a thorough analysis, the study evaluates the performance of these methods in generating semantic networks from text, particularly in capturing detailed event descriptions and relationships. Results indicate a slight advantage in accuracy for the Hybrid parser-based approach (87%) compared to ChatGPT (85%) in sentence parsing tasks. Furthermore, efficiency analysis reveals that ChatGPT's response quality varies with prompt sizes, while the Hybrid parser-based method consistently maintains excellent response quality.

INDEX TERMS Application of sentence parsing, adverb prediction, ChatGPT, hybrid parser, natural language processing, sentence parsing, semantic graph.

I. INTRODUCTION

Semantic graphs play a multifaceted role in various applications, spanning information retrieval, knowledge representation, question answering, text summarization, document clustering, and classification. Furthermore, we can find diverse real-world applications across healthcare, finance, cybersecurity, and entertainment industries [7] as illustrated in Figure 1. In healthcare, semantic graphs are instrumental in integrating medical knowledge [8], providing a centralized view of IoT information, and enhancing virtual assistants and chatbots by delivering more contextually relevant responses to medical queries [10]. Furthermore, in the finance industry, semantic graphs have emerged as a crucial tool for managing financial knowledge securely [11], enabling applications like transaction surveillance, financial crime detection and prevention, and non-compliant user detection [12]. In the entertainment industry, particularly social media, knowledge graphs power social graphs that help platforms like Facebook connect users within the context of their relationships, while also enhancing recommender systems to offer personalized content recommendations based on user interests [13]. Moreover, semantic graphs play a vital role in cybersecurity by mapping historical cyber attacks and predicting potential future breaches, thus bolstering cyber defense strategies [14].

This study explores the creation of semantic graphs, which are visual representations of knowledge and the interconnections between concepts. Specific tools within the domain of NLP parsing are working for constructing these semantic graphs. However, there are limitations in their ability to present detailed event descriptions, particularly concerning time and place. Recognizing the limitations present in current NLP parsing tools, the primary objective of this research is to enhance the existing approach. To address these limitations, this paper introduces a solution that involves identifying all functional components, including Subject, Predicate, Direct Object, Indirect Object, and Conjunction. Simultaneously, the method explores the prediction of adverb types, encompassing Time, Place, Manner, Degree, and Frequency, thus enriching the depth of linguistic analysis.

The special focus of this research is the role of adverbs, which are integral elements in language. It provides essential details regarding how actions are performed, the timing

The associate editor coordinating the review of this manuscript and approving it for publication was Mehedi Masud.

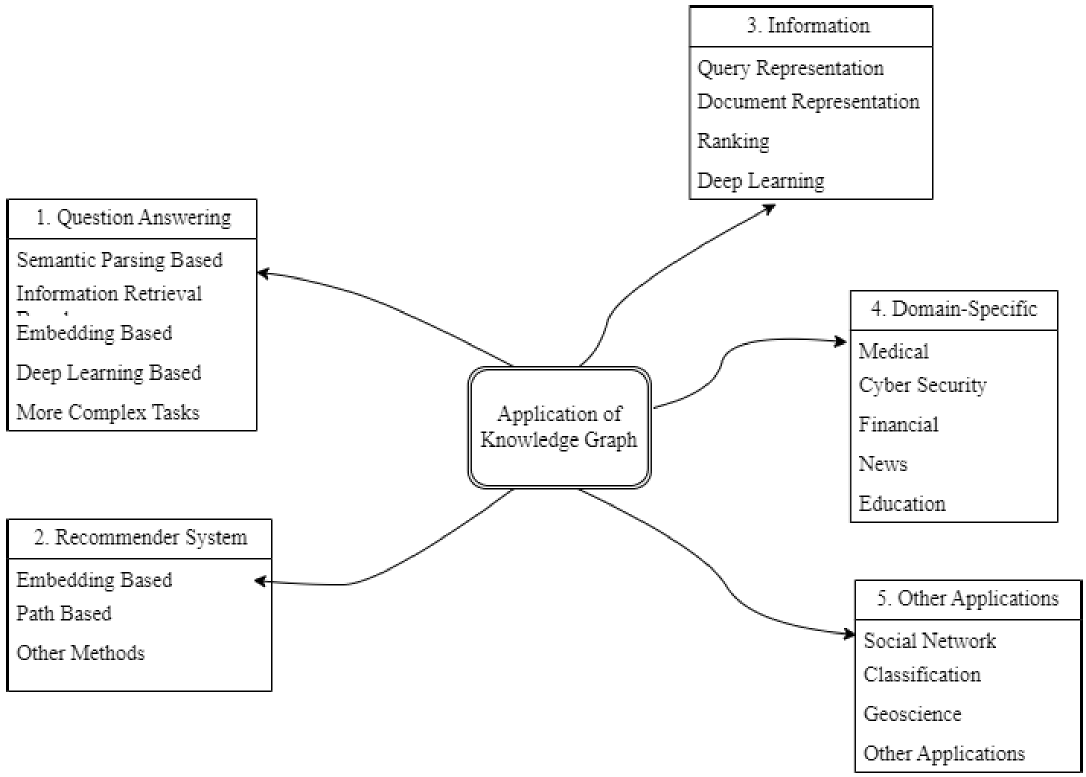

FIGURE 1. **Application fields of knowledge graphs [7].**

of events, specific locations, frequency, and the degree of attributes [15]. These linguistic modifiers play a fundamental role in parsing sentences and contribute significantly to our comprehension of context and details within statements.

To gain a deeper understanding of knowledge, concepts, and the complex web of relationships between them, this research extends beyond traditional limitations by incorporating a more comprehensive set of components. Specifically, the study introduces novel ChatGPT-based and Hybrid Parser-based Semantic Graph Construction and conducts a comparative analysis. This analysis assesses the details of these two approaches, dissecting their respective strengths, weaknesses, and applications.

In this regard, ChatGPT is one of the state-of-the-art LLMs (Large Language Models) [17], that has emerged as a transformative force in the field of NLP. It plays a pivotal role in the construction of semantic graphs by leveraging their natural language understanding capabilities. These models are trained on extensive text corpora and can extract and encode intricate relationships between concepts and entities within textual data. ChatGPT's previous experiences with these tasks are informed by its extensive pre-training on a diverse range of internet text [18]. This pre-training allows it to understand and generate human-like text and perform tasks related to semantic graph construction with high accuracy. By leveraging this understanding, ChatGPT can contribute significantly to the creation and enrichment of semantic graphs across various domains, from healthcare and finance to information retrieval and content recommendation [19]. It has demonstrated remarkable skill in a wide array of language understanding tasks, including question-answering, language generation, and text summarization [20]. However, the question arises: can ChatGPT be effectively harnessed to tackle the difficulties of semantic graph-based induction? On the other hand, Hybrid Parser-based methods integrate multiple NLP components, combining rule-based and machine-learning techniques, to extract and represent semantic relationships from text. The marriage of these disparate approaches promises enhanced robustness and adaptability. This study sets out to investigate which of these approaches outshines in the domain of semantic graph construction, and whether a hybrid approach provides a balanced solution.

This work delves into advancing semantic graph construction techniques, crucial for unraveling intricate relationships and meanings within textual data. The primary contributions encompass an innovative extension of semantic graph structures, including elements like Subject, Predicate, Direct Object, Indirect Object, Conjunction, and various adverb types. The study addresses the limitations of current NLP parsing tools, particularly in capturing detailed event descriptions involving time and place. Furthermore, it introduces a novel ChatGPT-based parsing approach, evaluating the model's proficiency in breaking down sentences into functional components and predicting adverb types. The research also conducts a comparative analysis between ChatGPT-based methods and Hybrid Parser-based techniques, offering valuable insights into their strengths, weaknesses, and real-world applications in semantic graph-based induction.

The paper is structured as follows: In the second section, a semantic graph construction model is presented, and a detailed procedure for building the presented model is provided. We discuss the latest NLP background technology and results. Additionally, we explore different knowledge base resources and their applications. The third section describes the proposed Hybrid Parser-based method, explaining all process steps. In the fourth section, we describe the ChatGPT-based method, encompassing the environment, dataset size, benchmark, and evaluation methods. Next, we present the experimental results, analyze the evaluation findings from multiple perspectives, and demonstrate the potential applications of our approach. In the sixth section, we conduct an efficiency analysis and engage in a discussion. Finally, in the concluding section, we summarize our findings and offer suggestions for future research directions.

II. LITERATURE REVIEW

A semantic graph is a graph model where nodes represent concepts and edges (or arcs) represent relationships between those concepts [21]. This model type is often used in artificial intelligence applications for representing knowledge.

DEFINITION 2.1

A graph $G = (V, E)$ is defined by a set of nodes V and a set of edges E between these nodes. Let $E \subseteq V \times V$ represent directed edges or arcs [22]. Each directed edge $(u, v) \in E$ signifies a connection from a start (tail) vertex u to an end (head) vertex v, where u and v are elements of the node set V. The graph's structure is characterized by these directed connections, providing a representation of relationships between nodes. Each node is associated with a label $Label(v)$.

Building semantic graphs is essential for many practical uses and ongoing research [8], [23], [24]. As we have more and more data available, creating these meaningful graphs becomes increasingly important for learning from different sources. Scientists keep looking for new ways to make this field better, and they use it in things like understanding language, organizing knowledge, and using artificial intelligence. They make structured graphs and networks to show how words, ideas, and things are connected. These graphs help in finding information, answering questions, and suggesting things you might like. So, making these graphs is a big part of helping computers and people work together better. When texts are represented graphically, it allows the preservation of additional information like the text's inner structures, semantic relationships, and term order. However, events like these are not effectively captured using current NLP parsing and semantic graph construction.

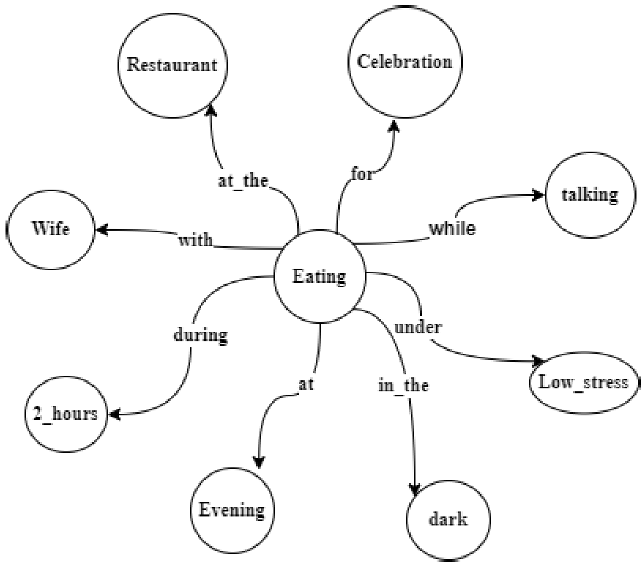

FIGURE 2. A visualization of the basic event knowledge graph for eating [9].

As an illustration, Figure 2 provides a visual insight into a fundamental event knowledge graph centered around the concept of "eating" [9].

Understanding natural language is a big challenge, and that's where semantic graphs come into play. Enhancing our grasp of natural language relies heavily on the development of semantic graphs, a field that's been increasingly in the spotlight. Researchers are actively exploring the creation of these graphs and how they can represent knowledge, diving into structured data, relationships, and more detailed elements, which align with prior work on Semantic Role Labeling (SRL) and adverb sense disambiguation. These efforts aim to provide a more comprehensive understanding of semantic parsing, event descriptions, and the complexities involved, as outlined in related works [25].

In this regard, RDF and ontologies are the foundation for constructing structured, machine-readable semantic graphs, playing a pivotal role in knowledge representation and the advancement of the semantic web. RDF, with its subject-predicate-object triples and URIs, ensures global consistency and interoperability. Ontologies, including OWL and RDFS, enrich RDF's capabilities by defining the vocabulary and structure for resources and relationships within specific domains, making it easier to understand and work with the information [40]. Together, RDF and ontologies are super important for making and using semantic graphs across different fields.

At the same time, the Semantic Web initiative is pushing for structured data to be shared and linked on the web. They're using things like Linked Data, RDF, and SPARQL queries to create big semantic graphs that cover a lot of the web [41]. But there are challenges too. We need better ways to handle big sets of data, put together text and visual data, and make sure the knowledge graphs we create are complete and correct.

Researchers are used new techniques, like word embeddings and entity embeddings, to help to understand the fine details of how words and things are related [7]. As we have more and more data, making meaningful semantic graphs becomes super important for getting useful information from different places.

In general, the fields of RDF, ontologies, and the ideas behind the Semantic Web initiative where semantic graph play an important role that understand and manage information. The semantic graphs serve as a crucial foundation for knowledge representation and data integration, facilitating the consistence management of structured data on the web. However, this field is evolving, with ongoing efforts focused on improving graph construction techniques, addressing data handling challenges, and harnessing the power of embedding techniques to capture richer semantic relationships. As the landscape of available data continues to expand, the construction of semantic graphs becomes essential for unlocking valuable insights and enabling data-driven applications across various domains

III. OPENAI API-BASED SENTENCE PARSING

A significant aspect of language models is the Large Language Model (LLM), recognized for its capacity to achieve a wide-ranging understanding of language and proficiently generate text. LLMs acquire this capability through an extensive training process where they learn from vast amounts of data, effectively processing billions of parameters. This training demands substantial computational resources [31]. These language models primarily employ artificial neural networks, predominantly relying on transformer architectures, and undergo (pre-)training utilizing self-supervised and semi-supervised learning approaches [32].

Prominent examples of LLMs include OpenAI's GPT models like GPT-3.5 and GPT-4 (utilized in ChatGPT), Google's PaLM (employed in Bard), Meta's LLaMa, as well as BLOOM, Ernie 3.0 Titan, and Anthropic's Claude 2. In this study, due to the model's capabilities, researchers utilized the ChatGPT 3.5 OpenAI API for the sentence parsing.

Chat GPT (Generative Pre-trained Transformer) models are designed to understand and generate human-like text by processing vast amounts of data during training. They operate by predicting the next word in a sequence of words and have been instrumental in various NLP tasks. Understanding these fundamental concepts is essential for harnessing the power of GPT-based models in language-related applications. The accuracy of the ChatGPT 3.5 model heavily relies on the quality and representativeness of the labeled dataset used for fine-tuning [39]. The pre-trained ChatGPT model is fine-tuned on a labeled dataset of adverbs to improve its categorization accuracy.

A. THE ARCHITECTURE OF CHATGPT

ChatGPT is based on the transformer architecture, that allows for parallel processing, which makes it well-suited for processing sequences of data such as text. ChatGPT uses the PyTorch library, an open-source machine learning library, for implementation. ChatGPT is made up of a series of layers, each of which performs a specific task.

B. PROMPT ENGINEERING TECHNIQUES

Prompt engineering is a crucial technique employed to guide the behavior of large-scale language models like ChatGPT. By strategically constructing input prompts, researchers and developers aim to obtain more accurate and relevant responses from these models [36]. Several prompts engineering strategies, including prompt rewriting, contextual incorporation, explicit instructions, and templates, have been proposed to address control and responsiveness challenges, aligning the model's outputs with user targets and expectations. The careful design of prompts plays a pivotal role in influencing the quality and relevance of ChatGPT's responses, making it a valuable skill for those working with AI systems. For instance, in a real-world context, prompt engineering bears the potential to enhance the efficiency, accuracy, and effectiveness of healthcare delivery by guiding AI models to provide valuable insights and solutions. However, it's crucial to acknowledge the limitations and risks associated with AI, such as the model's inability to access real-time data or offer personalized medical advice. This necessitates verification by qualified professionals and raises concerns about privacy and data security. Despite these challenges, the significance of prompt engineering has seen exponential growth since the inception of ChatGPT, with ongoing research endeavors aimed at refining and expanding this critical skill, particularly within the medical field. In this specific study, researchers have developed and employed high-quality training sets as templates for prompts to augment the accuracy of responses.

C. METHODOLOGY

The methodology for this study involves the following steps.

1) PRE-TRAINING CHATGPT 3.5

The initial step involves utilizing the pre-trained ChatGPT 3.5 model, which has been fine-tuned by OpenAI on a vast corpus of text data. This model serves as the foundation for the subsequent tasks.

2) CONSTRUCTION OF A LABELED DATASET

A high-quality labeled dataset is carefully collected to fine-tune ChatGPT for sentence parsing by including the adverb type prediction. This dataset includes Subject, Predicate, Direct Object, Indirect Object, Conjunction, and adverb types such as Time, Place, Manner, Degree, and Frequency. The dataset is essential for training ChatGPT to categorize adverbs accurately and for sentence parsing.

3) FINE-TUNING CHATGPT

Fine-tuning is a phase where the pre-trained model is further trained on the specific task it will be used for. The objective of this phase is to adapt the model to the specific task and fine-tune the parameters so that the model can produce outputs that are in line with the expected results. The pre-trained ChatGPT 3.5 model is fine-tuned using the labeled dataset of functional sentence structure. One of the most important things in the fine-tuning phase is the selection of the appropriate prompts. The prompt is the text given to the model to start generating the output. Providing the correct prompt is essential because it sets the context for the model and guides it to generate the expected output. It is also important to use the appropriate parameters during fine-tuning, such as the temperature, which affects the unpredictability of the output generated by the model. As shown sample prompt template the researcher developed and used representative prompt templates from the collected dataset. In the context of ChatGPT 3.5 and its use via the OpenAI API for sentence parsing tasks, prompt engineering is a critical factor in achieving accurate and efficient results. This process involves crafting clear, concise prompts that clearly define the task parameters, such as identifying parts of speech or parsing sentence structures, and specifying the desired output format. Since ChatGPT 3.5 responds based on input structure, prompt engineering also includes efficient token use, vital due to API token limitations, and incorporating relevant contextual information to enhance understanding of nuanced or domain-specific language. Iterative refinement, where prompts are continuously adjusted based on the model's outputs, plays a key role in fine-tuning the model's sentence parsing abilities. Additionally, well-designed prompts can aid in resolving ambiguities and handling errors inherent in NLP. Customization for specific domains, such as including domain-specific language in prompts, further enhances the model's performance in targeted areas, making prompt engineering a blend of art and science essential for optimizing ChatGPT 3.5's sentence parsing capabilities.

Sample prompt template example:

```
Prompt_template="""
Sentence 1: The coffee shop is always busy in the morning.
Parsing Answer 1: {'Predicate': 'is', 'Subject': 'The coffee shop', 'Direct Object':[], 'Indirect Object': [], 'Time': 'in the morning', 'Place': [], 'Manner': 'always busy', 'Frequency': [], 'Degree': []}

Sentence 2: The train arrived at the station on time.
Parsing Answer 2: {'Predicate': 'arrived', 'Subject': 'The train', 'Direct Object':[], 'Indirect Object':[], 'Time': 'on time', 'Place': 'at the station', 'Manner': [], 'Frequency': [], 'Degree':[]}

Sentence 3: Ethiopia defeated Italy at the Battle of Adwa.
Parsing Answer 3: {'Predicate': 'defeated', 'Subject': 'Ethiopia', 'Direct Object': 'Italy', 'Indirect Object': [], 'Time': [], 'Place': 'at the Battle of Adwa', 'Manner': [], 'Frequency': [], 'Degree': []} """

custom_prompt=""" Generate the predicate, Subject, Direct Object, Indirect Object, Time, Place, Manner, Frequency, Degree, Conjunction parts of carefully and attentively at the library everyday. """

prompt=prompt_template + custom_prompt
```

The provided example demonstrates an effective approach to using prompt engineering for sentence parsing in Python with ChatGPT. The *prompt_template* string contains a structured format for parsing sentences, where each sentence is followed by a "*Parsing Answer*" that breaks down the sentence into various grammatical components like Predicate, Subject, Direct Object, etc. This structured format serves as a guide for the expected output.

Following this, a *custom_prompt* is defined, requesting the model to parse a new sentence ("carefully and attentively at the library every day") using the predefined structure. When these two parts are concatenated into the prompt variable, it provides ChatGPT with a clear example of the task at hand, along with a new sentence to parse. This method of prompt engineering leverages the model's ability to understand and mimic the format presented in the examples, aiming to achieve a similar breakdown for the new sentence. It's a practical demonstration of how well-structured prompts can significantly improve the accuracy and relevance of the model's output in sentence-parsing tasks.

IV. HYBRID PARSER-BASED SENTENCE PARSING

The creation of a Hybrid Parser-based sentence parsing framework is a noteworthy breakthrough in the field of NLP. This innovative approach combines rule-based and machine-learning methods to extract meaning from text [42], addressing the limitations of current NLP parsing techniques. By incorporating both rule-based and machine-learning components, this framework becomes capable of handling a wider range of linguistic structures and domains, ensuring robust performance. Its primary objective is to enhance the accuracy of semantic parsing by capturing context-specific elements in language, ultimately improving the comprehension of the underlying meaning in the text. The framework strikes a careful balance between accuracy and efficiency, allowing for the precise construction of a semantic graph from textual content. The architecture of this framework encompasses text preprocessing, rule-based and machine learning-based sentence parsing, adverb-type prediction, and semantic graph construction.

One distinguishing feature of this framework is its dedicated component for predicting adverb types within the text. This feature plays a pivotal role in accurately extracting the essence of a sentence. The integration of outputs from both rule-based and machine learning-based parsing yields a comprehensive semantic graph representing the structured knowledge present in the text. This Hybrid parser-based approach harnesses the strengths of rule-based systems, which excel at handling linguistic patterns and prior knowledge, and machine learning models, which adapt to context and data-driven insights. As a result, the framework enhances natural language understanding and information extraction, offering a promising solution to the challenges presented by traditional parsing methods.

A. METHODOLOGY

The researchers utilized a free cloud-based platform called Google Collaboratory for running and writing Python code. For text analysis and parsing, we used essential parsing tools such as spaCy and NLTK. To improve the analysis and understanding of language, we integrated external resources, including dictionaries like Webster and ontologies such as WordNet.

Furthermore, to train the adverb prediction model, the dataset that contained definitions and synsets derived from a list of adverbs and prepositions is carefully collected, playing a fundamental role in model training.

To enhance the precision of adverb prediction, the researchers incorporated the machine learning technique known as Latent Dirichlet Allocation (LDA), with specific application of the MLP (MultiLayer Perceptron) model. The researchers utilize the power of LDA to construct topic-based feature vectors for words, with a particular focus on adverbs. LDA is commonly used in NLP to discover hidden topics within a corpus of text. The process of generating these feature vectors comprised several key steps: first, LDA modeling was applied, wherein words were associated with specific topics to discover the underlying semantic patterns. Then, the *LDA_vector* method is introduced and designed to take a word as input and determine its LDA representation, representing the word as a vector of topic probabilities based on its contextual associations.

Additionally, the *Webster_LDA_vector* method is defined to extend this capability to adverbs not found in Wordnet but present in word embeddings, thereby broadening the scope of the LDA approach. Ultimately, the LDA-derived vectors obtained from these methods were integrated into the feature vectors for adverbs, providing a structured means to measure their similarity or categorization in the context of the discovered semantic topics. This feature-based analysis allowed for comprehensive comparisons with other word similarity measures, including spaCy and Wordnet-based metrics, enhancing our understanding of adverb similarities and categories.

In addition to the methodological approach, the researchers utilized the power of word embeddings. Word embeddings are a way to represent words as dense vectors in a continuous vector space, allowing us to capture relationships between words and how they fit into sentences. Within the scope of this study, the utilization of word embeddings offers several advantages. First, they help us measure how similar words are to each other, which is particularly useful for understanding adverbs in the context of other words. Second, when we encounter words that aren't in the dictionary (Wordnet) we're using in the code, word embeddings provide a smart solution by giving us vector representations for a wide range of words. Third, they enable us to understand the meaning of words within their context, making it easier to figure out what adverbs mean based on the words they're associated with. Fourth, when we're creating graphs that show how words relate to each other, word embeddings enhance these vectors with more information. This enrichment helps us better understand the roles of adverbs and other words in sentences. Lastly, the integration of word embeddings results in more accurate and detailed graphs, representing words and their connections in sentences, ultimately enhancing our overall understanding.

Now, with the understanding of how word embeddings enhance our analysis of word relationships, let's delve into the process of determining the functional type of a given sentence sequence. This process involves analyzing the structure and components of sentences to categorize them into different functional units. To do this, we consider a set of accepted functional unit types, which include Predicate, Subject, Direct Object, Indirect Object, Time, Place, Manner, Frequency, Degree, and Conjunction. This parsing process is the initial step in our study.

Having an input word sentence, $s = w_1, w_2, \ldots, w_l$. where symbol w denotes a word inside the sentence. The set of accepted functional unit types is given by

$$\begin{aligned} T = \{&\text{Predicate, Subject, Direct Object,Indirect Object,}\\ &\text{Time,Place, Manner,Frequency,}\\ &\text{Degree, Conjunction}\} \end{aligned} \tag{1}$$

To determine the functional type of a given sentence sequence, the following parsing processes are first:

1) The internal dictionary contains the list of frequent adverb words, like the phrase as soon as, in this case, the dictionary contains also the related functional type.
2) Label of the dependency parsing: l_e This property is generated with the spacy parser as the label of the dependency edge from the generated dependency tree.
3) Wordnet-based Lin similarity (l_l): a score denoting how similar two word senses (s_1, s_2) are, based on the Information Content (IC) of the Least Common Subsumer (s_c) most specific ancestor node) and that of

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 1100)              49500

 dropout (Dropout)           (None, 1100)              0

 dense_1 (Dense)             (None, 440)               484440

 dense_2 (Dense)             (None, 132)               58212

 dense_3 (Dense)             (None, 6)                 798

=================================================================
Total params: 592950 (2.26 MB)
Trainable params: 592950 (2.26 MB)
Non-trainable params: 0 (0.00 Byte)
```

FIGURE 3. MLP architecture.

the two input synsets:

$$l_l(s_1, s_2) = \frac{2 \cdot IC(s_c)}{IC(s_1) + IC(s_2)}$$

4) Wordnet-based path similarity (l_p): the path between the two synsets in the concept tree of the wordnet
5) Wordnet LDA similarity (l_d): we take the definition sections from the Wordnet database and calculate the topic similarities using the LDA method.
6) Webster LDA similarity (l_w): the definitions in Webster dictionary are used to calculate the topic similarities using the LDA method.
7) Spacy similarity (l_s): the similarity is based on the grammatical properties generated in the spacy NLP library

The proposed framework also includes a dictionary which contains some selected words with the related unit types labels:

$$D = \{(w, T(w)\}$$

We divide this dictionary into two parts:

$$D = D_B \bigcup D_L$$

where D_B is the set of baseline words, we use to determine the similarity positions of new query words. For a given query word w_q, the following local feature vectors are calculated:

$$\{l_e(w_q, w), l_l(w_q, w), l_p(w_q, w), l_d(w_q, w), l_w(w_q, w), l_s(w_q, w)|w \epsilon D_B\}$$

Using these similarity measures, the generated similarity vectors are merged into a global feature vector

$$l(w_q)$$

These global feature vectors are used to predict the corresponding unit type label of w_q. For the prediction, an MLP neural network module (NN) is involved, where

$$NN(l(w_q))$$

outputs the predicted unit label.

For the training of the MLP unit, the D_L dataset is used as training and test dataset.

The MLP neural network unit under consideration comprises five layers, with one dedicated to model regularization (as depicted in Figure 3). The trained MLP unit demonstrated a commendable average accuracy of 92% on the tested datasets.

Figure 4 displays the validation accuracy curve during the training process of the proposed framework. The curve illustrates how the accuracy of the model evolves as it undergoes training iterations. It provides valuable insights into the model's performance and its ability to generalize to unseen data, showcasing the progress made during the training phase.

As shown in Figure 5 the training process, the neural network underwent 200 epochs on the provided dataset. The model demonstrated a dynamic learning curve as indicated by the changes in loss and accuracy metrics over each epoch. The training loss, starting at 0.1673, gradually decreased, converging towards 0.1592 by the 125th epoch. Simultaneously,

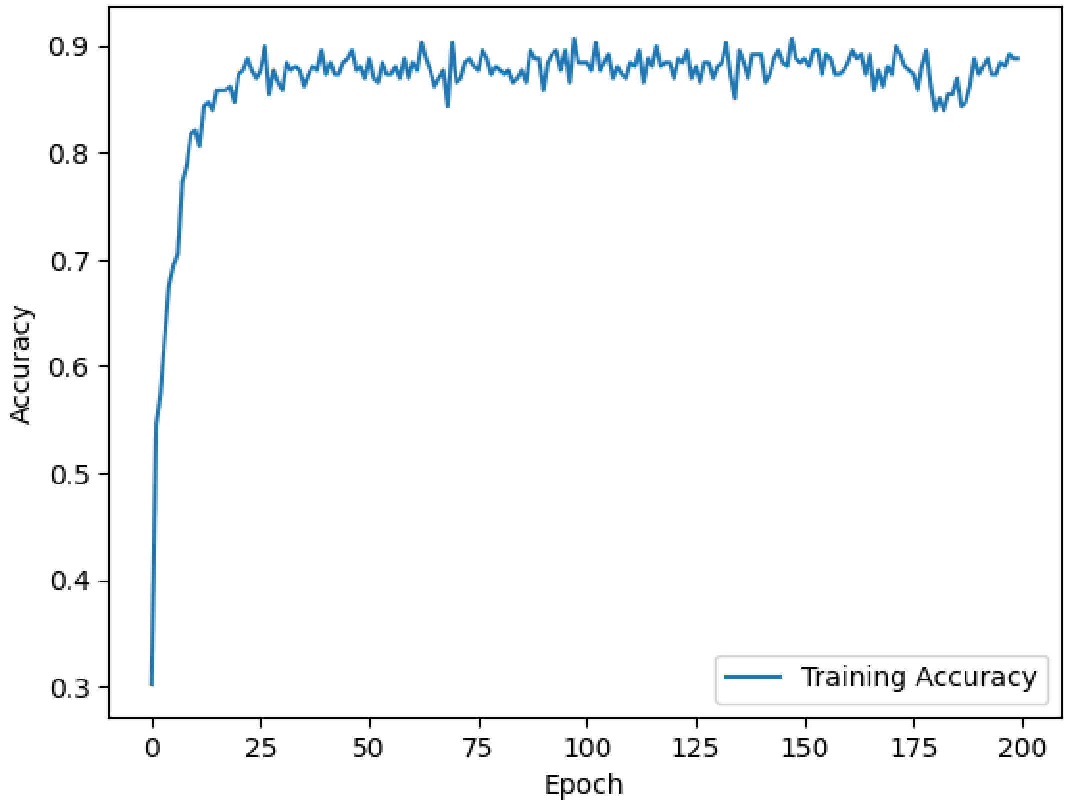

FIGURE 4. Training accuracy curve in the training process.

the training accuracy improved from 87.69% to 89.18%. The validation loss and accuracy also exhibited variations but generally followed a similar trend. It's important to note the fluctuations and convergence patterns to assess the model's performance and potential for further fine-tuning or optimization.

Figure 6 presents an RDF (Resource Description Framework) graph in Turtle format, illustrating the semantic relationships of a simple narrative. Central to this graph is 'ns1:John', around whom various properties are structured. These properties include 'ns1:hasDirectObject', pointing to 'ns1:book', indicating an action involving a book, and 'ns1:hasIndirectObject', linked to 'ns1:Mary', suggesting an interaction with Mary. The action is further described by 'ns1:hasPredicate' as 'ns1:gave', indicating the nature of the interaction as giving. Additional details include 'ns1:hasManner' (generously), 'ns1:hasPlace' (park), 'ns1:hasTime' (yesterday), and 'ns1:hasFrequency' (once), providing context for when, where, how often, and in what manner the action took place. This RDF graph effectively demonstrates the use of semantic web technology to represent complex relationships and attributes in a structured, machine-readable
format.

The graph 11 visualizes the RDF data structure derived from the Turtle syntax, effectively mapping the semantic relationships described in the dataset. At the center of the graph is 'ns1:John', depicted as a node connected to various other nodes, each representing a distinct property or attribute in the narrative. These connections include links to 'ns1:book' through 'ns1:hasDirectObject', indicating the object of John's action, and to 'ns1:Mary' via 'ns1:hasIndirectObject', illustrating the recipient or beneficiary of the action. The action itself is defined by the 'ns1:hasPredicate' node, connected to 'ns1:gave', signifying the nature of John's interaction.

Furthermore, the graph details contextual elements of the event: the 'ns1:hasManner' node linked to 'ns1:generously' portrays how the action was performed, while 'ns1:hasPlace' and 'ns1:hasTime' nodes connect to 'ns1:park' and 'ns1:yesterday', respectively, specifying the location and timing of the event. Finally, the 'ns1:hasFrequency' node, connected to 'ns1:once', provides information on the occurrence frequency of the event.

Each node and link in the graph is clearly labeled, offering an intuitive and easily interpretable visualization of the RDF data. The layout is designed for easy tracing of relationships, making the complex data structure comprehensible at a glance. This graph serves as a practical example of how RDF data can be transformed into a visual representation, facilitating better understanding and analysis of semantic relationships."

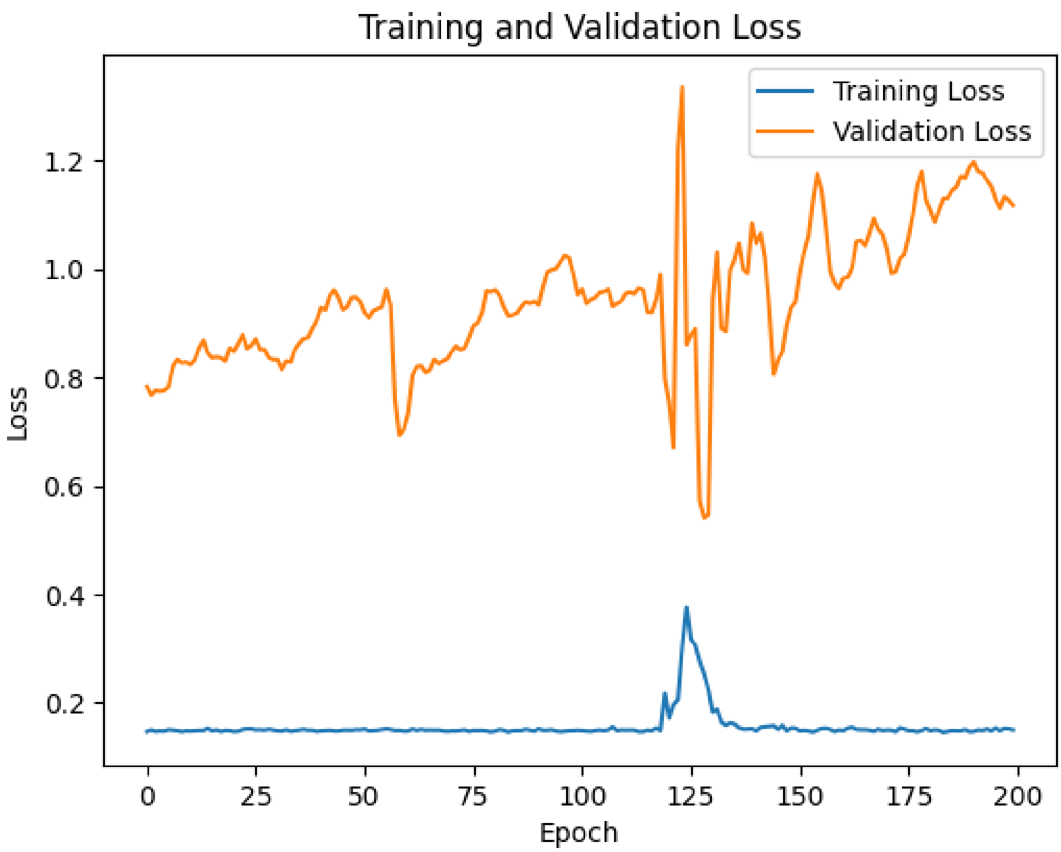

FIGURE 5. Training and validation loss curve in the training process.

```
@prefix ns1: <http://example.org/> .

ns1:John ns1:hasDirectObject ns1:book ;
    ns1:hasFrequency ns1:once ;
    ns1:hasIndirectObject ns1:Mary ;
    ns1:hasManner ns1:generously ;
    ns1:hasPlace ns1:park ;
    ns1:hasPredicate ns1:gave ;
    ns1:hasTime ns1:yesterday .
```

FIGURE 6. RDF data example.

V. EXPERIMENTAL RESULTS

A. DATASET PREPARATION TECHNIQUES

The model semantic graph categorizes words and phrases within a given sentence into various functional structures, encompassing subject, predicate, direct object, indirect object, time adverbs, place adverbs, frequency adverbs, and manner adverbs. Due to the cost-intensive process of manual dataset collection and evaluation, we have only used 160 sentences for the training dataset and 40 sentences for testing. These datasets are collected from various sources, like academic history, biology, and world facts.

One significant limitation within this area of sentence parsing research is the absence of an automated performance evaluation system, which remains unimplemented. To assess the accuracy of the parsing, the researchers used the expertise of linguistic professionals. The survey encompassed five distinct rating categories: "Poor," "Below Average," "Average," "Above Average," and "Excellent.". The researchers used similar test datasets for both approaches and made a comparative result analysis.

B. RESULTS AND ANALYSIS

In the evaluation process, we used the ChatGPT efficiency for prompts of different lengths and complexity. The models evaluated in this study include the OpenAI API and ChatGPT 3.5 Web Interface, as well as a Hybrid parser-based method.

Figure 7 provides an overview of the comprehensive evaluation results of 15 linguistic experts for both methods. The evaluation scores range from a minimum of 1.5 to a maximum of 4, showcasing the experts' assessments of the performance of these methods.

Two methods of parsing OpenAI API-based and Hybrid parser-based are compared across a series of 15 different cases, presumably evaluated by different evaluators. The

TABLE 1. Efficiency of ChatGPT in dependency of prompt size.

Model	Prompt Size	Average Rating
OpenAI API	5	Poor
	15	Below Average
	25	Average
	35	Above Average
	40	Excellent
ChatGPT 3.5 Web Interface	–	Average
Hybrid Parser-based Sentence Parsing	–	Excellent

TABLE 2. ChatGPT-based and hybrid parser-based sentence parsing example: "The coffee shop is always busy in the morning.

Model	Size	Predicate	Subject	DO	Time	Place	Manner	Frequency	Degree	Conjunction
Expert		be	The coffee shop		in the morning		always busy			
OpenAI API	5	be	coffee shop		always		busy			
OpenAI API	15	be	The coffee shop		in the morning		always busy			
OpenAI API	25	be	The coffee shop		always		busy	in the morning		
OpenAI API	35	be	The coffee shop		always		busy			in the morning
OpenAI API	40	be	The coffee shop		always		busy	in the morning		
ChatGPT 3.5		be	The coffee shop		in the morning		always busy			
Hybrid Parser		be	The coffee shop		in the morning		always busy			

scores, ranging on a scale 0-5, are presented for each method across these cases. The OpenAI API-based method shows consistent performance, with scores mostly hovering around the 3.5 mark, indicating a stable and reliable parsing ability across different evaluators. The scores range from a low of 3.0 to a high of 3.8, suggesting some variability but generally clustering around the mid-3 range.

On the other hand, the Hybrid parser-based method exhibits a slightly more varied performance, with scores ranging from 3.2 to 4.0. This method seems to outperform the ChatGPT-based approach in several cases, peaking at 4.0 in multiple instances, which could indicate a higher efficiency or accuracy in certain contexts. However, its performance also dips to 3.2 at times, showing a bit more fluctuation compared to the ChatGPT-based method.

In the bar chart, each pair of bars represents one of the 15 evaluators' scores for both methods. The bars are arranged side by side for a clear, comparative view, with ChatGPT-based results on the left (in each pair) and Hybrid parser-based on the right. This layout facilitates a straightforward comparison between the two methods across different evaluative scenarios.

The X-axis of the chart labels each evaluator (Evaluator 1 to Evaluator 15), while the Y-axis represents the score. The legend, placed at the top, clearly distinguishes between the two methods. The overall layout of the chart, including rotated tick labels and a tight layout, is designed for enhanced readability and efficient data comparison.

Efficiency, as reflected in the average quality rating of responses generated by these models, is a key measure. We explored prompt set sizes ranging from 5 to 40. The OpenAI API model garnered quality ratings, spanning from "poor" with a prompt set size of 5, to "below average" with 15, "average" with 25, "above average" with 35, and ultimately "excellent" with a prompt set size of 40. Surprisingly, both the ChatGPT 3.5 Web Interface and the Hybrid Parser-based Sentence Parsing model consistently maintained an "excellent" response quality rating, irrespective of the prompt set size. This indicates their enduring efficiency across a spectrum of prompt set sizes. This table provides valuable insights into how different prompt set sizes impact ChatGPT model efficiency, revealing noteworthy

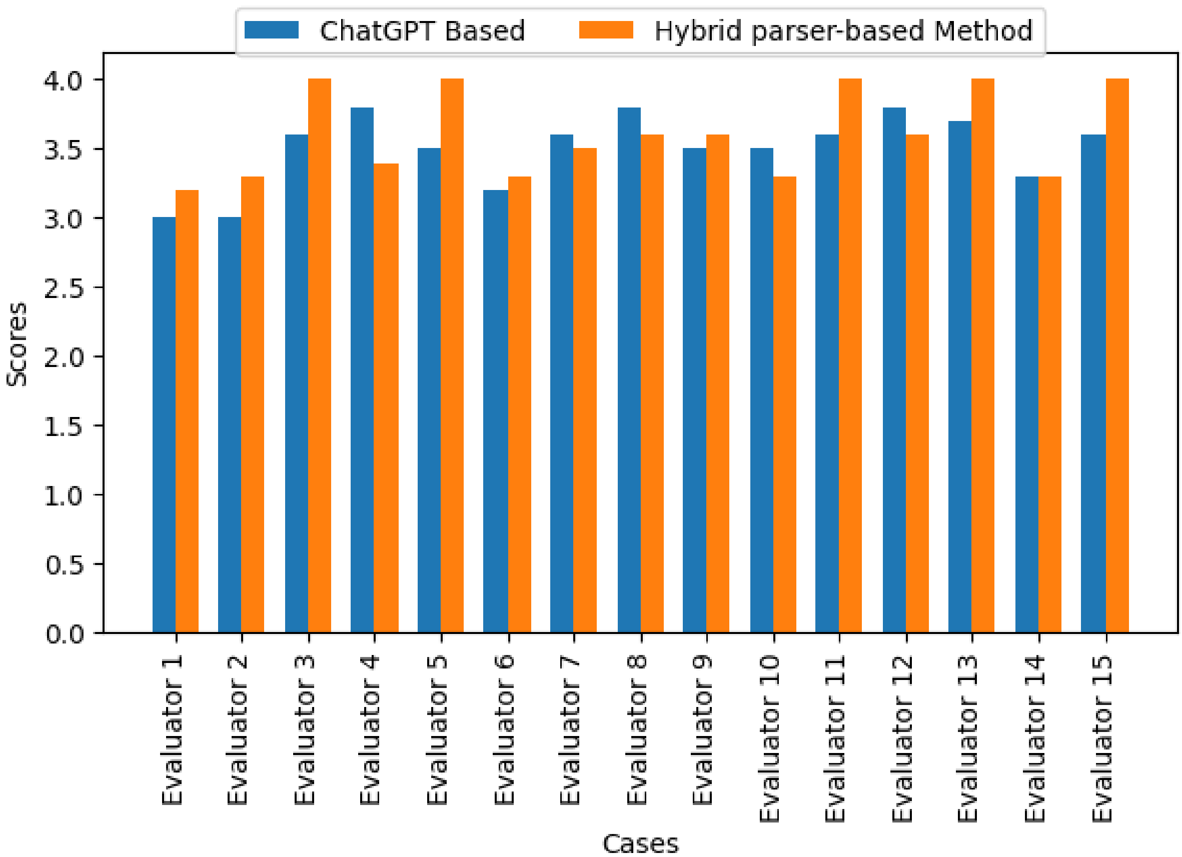

FIGURE 7. The overall evaluation result of linguistic experts.

disparities in performance between the OpenAI API and other models.

Table 1 illustrates the influence of prompt set size on ChatGPT's sentence parsing performance, quantified by accuracy. Accuracy is determined by the ratio of correctly assigned sentences to the total assigned sentences. The OpenAI API employs five distinct prompts, each with varying numbers of sentence parsing templates: prompt one with 5 templates, prompt two with 15 templates, prompt three with 25 templates, prompt four with 35 templates, and prompt five with 40 templates. As seen in Table 1, the accuracy of OpenAI models sees improvement as the number of templates within the prompts increases. In a separate experiment conducted with the ChatGPT 3.5 Web Interface, an accuracy score of 0.74 was achieved.

The graph 8 presents a comparison of accuracy scores for different models and prompts. It features seven data points, representing a mix of the OpenAI API with various prompts ("Prompt 1" through "Prompt 5"), the ChatGPT 3.5 Web Interface, and a Hybrid parser-based model.

Accuracy scores, ranging from approximately 0.64 to 0.87, are plotted on the Y-axis, while the X-axis is labeled with a combination of the model names and prompt numbers. This unique labeling system provides a clear distinction between each model-prompt combination. The OpenAI API's performance across five different prompts shows a general trend of increasing accuracy, starting from 0.64 and reaching up to 0.85.

In contrast, single data points represent the ChatGPT 3.5 Web Interface and the Hybrid parser-based model, with accuracy scores of 0.74 and 0.87, respectively. These points stand out as they are not part of the continuous line representing the OpenAI API's performance.

Table 2 provides a comparative analysis of sentence parsing results for the given input sentence "The coffee shop is always busy in the morning" as an example. The table includes various models used for sentence parsing, such as a Linguistic Expert, OpenAI API with different template prompt sizes, ChatGPT 3.5 Web Interface, and the proposed Hybrid parser-based model. The models' parsing results are presented for different template prompt sizes, with the extracted functional units, including Predicate, Subject, Direct Object, Indirect Object, Time, Place, Manner, Frequency, Degree, and Conjunction. This comparison highlights the effectiveness of the Hybrid Parser-based approach in accurately identifying and extracting functional units from the input sentence.

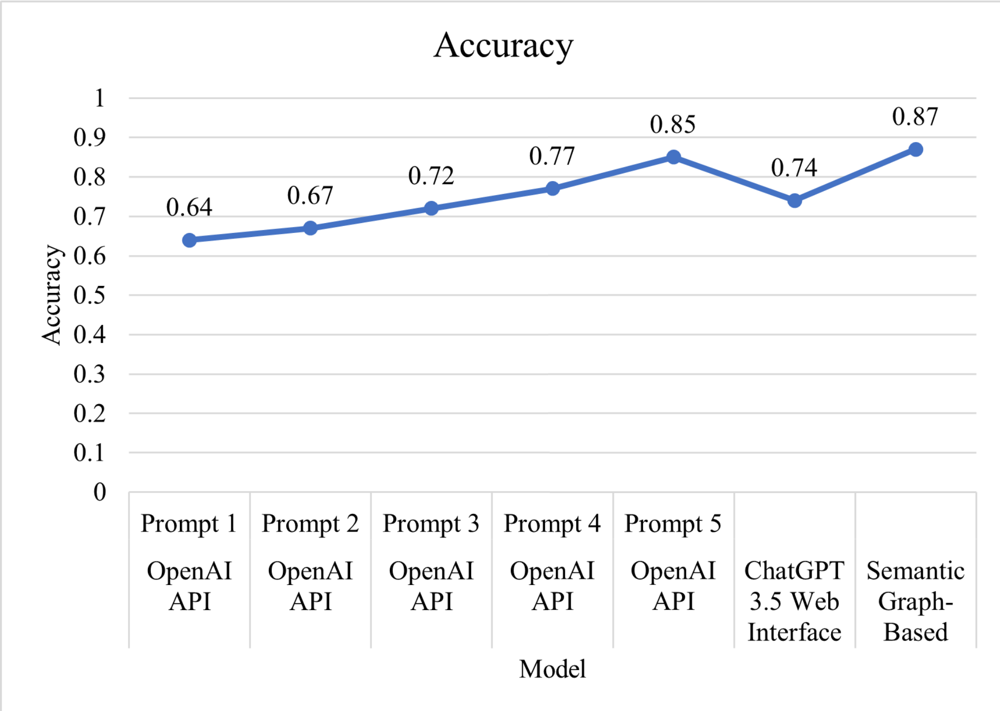

FIGURE 8. OpenAI API and hybrid parser-based sentence parsing accuracy.

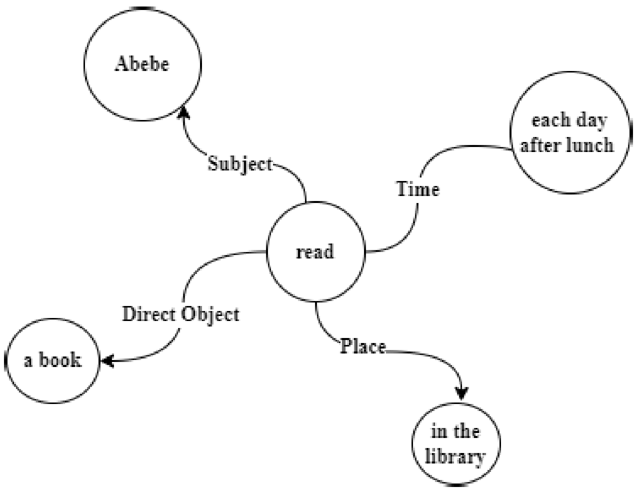

FIGURE 9. Semantic graph generated by the proposed model for the sentence " Abebe reads a book deeply in the library each day after lunch."

VI. REAL APPLICATION OF SENTENCE PARSING

In the area of sentence-parsing applications, two important examples are AQG and the creation of Ontology Semantic Graphs. These applications represent practical uses of sentence parsing, frequently employing sophisticated NLP techniques. Let's delve into how these applications are interconnected with the process of sentence parsing:

A. AUTOMATIC QUESTION GENERATION

The AQG process begins with parsing instructional text, where relevant information is extracted and relationships between different components are understood. This parsed data is then utilized to craft meaningful questions aligned with specific learning objectives. The parsed structure guides question generation, ensuring contextual relevance and contributing to a cohesive learning experience. Thus, sentence parsing serves as a vital beginning to AQG in ITS, showcasing its real-world impact in customizing assessments, delivering prompt feedback, and cultivating a learning environment that is both personalized and adaptive.

Figure 10 illustrates a set of sample questions that have been automatically generated from the sentence "John ate an apple yesterday." The questions showcased in the figure are the result of an automated question-generation process. This process involves analyzing the given sentence and formulating relevant questions to assess comprehension or generate educational content. I have used Gradio for building GUI web applications. In the context of AQG, sentence parsing proves critical for comprehending grammatical structures and extracting meaningful components, such as subjects, predicates, and objects.

B. GENERATION OF ONTOLOGY SEMANTIC GRAPHS

Regarding the implementation of an ontology and NLP engine, Python is one of the most common languages used. In knowledge management, ontology offers a common vocabulary that can be used to model various domains, including the types of objects, related concepts, and their properties and relationships. The shared database model may involve external training sets that can be used

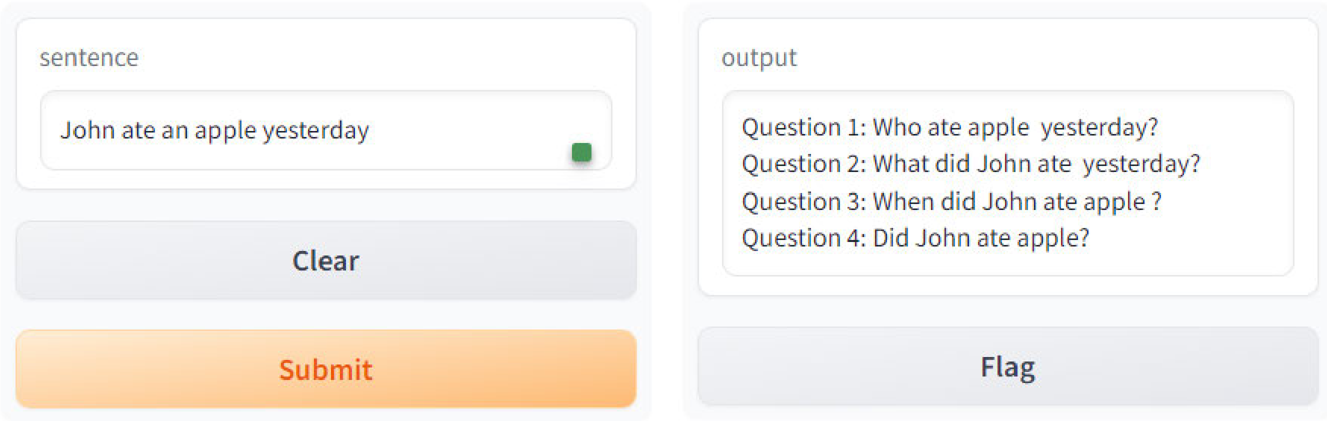

FIGURE 10. Sample automatic generated questions for the sentence Ïohn ate an apple yesterday.

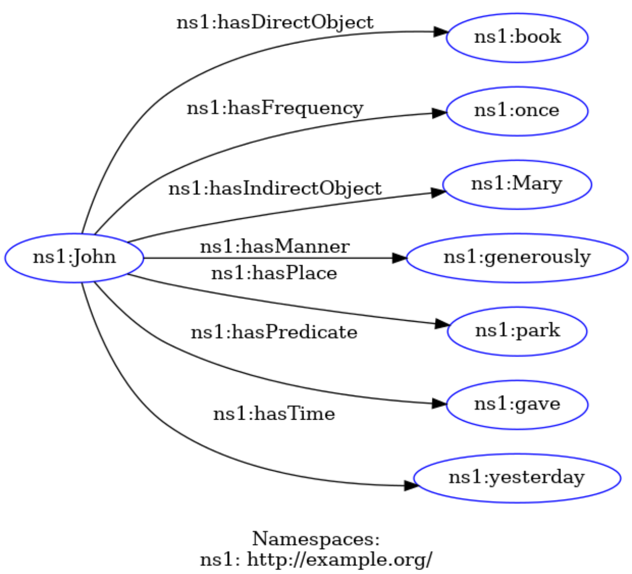

FIGURE 11. Sample RDF graph for the sentence "John gave a book to Mary at the park yesterday."

as input data in different data mining techniques, like NN.

In Figure 11, we can see a sample RDF graph corresponding to the sentence "John gave a book to Mary at the park yesterday." The RDF graph visually represents the structured information derived from the sentence using the RDF format. Each node and link in the graph signifies a distinct element or relationship present in the sentence. This graphical representation offers a clear illustration of how the sentence's content is encoded into RDF, facilitating a more organized and standardized representation of information.

Connection with Sentence Parsing and ontology development often start with parsing and extracting information from natural language text. Sentence parsing helps identify key concepts, relationships, and attributes, which are then structured into an ontology. Ontology semantic graphs find applications in various domains, including healthcare, finance, and the semantic web. They provide a structured representation of knowledge, enabling more effective data integration, search, and reasoning.

VII. CONCLUSION AND FUTURE WORK

In conclusion, the process of semantic graph construction stands as a cornerstone in the field of knowledge representation and artificial intelligence, giving structured meaning upon the vast landscape of textual data. It draws its strength from an array of foundational technologies, encompassing NLP, dependency parsing, word embeddings, LDA, and the integration of ontologies and knowledge graphs. These technological underpinnings empower the creation of semantic graphs, spanning from entity recognition to intricate topic modeling. The infusion of ChatGPT's NLP capabilities further enriches this process, rendering it a dynamic and adaptable tool for semantic graph construction.

Our deliberate experimentation and meticulous evaluation have illuminated the comparative performance, applicability, and constraints of ChatGPT-based and Hybrid Parser based sentence parsing methods within the context of semantic graph construction. These findings not only contribute to the expanding reservoir of knowledge within the field of NLP but also offer invaluable insights to researchers, developers, and practitioners venturing into real-world applications. These applications include information retrieval, knowledge graph development, and automated question-answering systems, among others.

It's worth noting that the accuracy values indicate a slightly better performance of the hybrid parser-based sentence parsing method compared to the ChatGPT-based model $acc\_GPT = 0.85$, $acc\_hybrid = 0.87$. In this evaluation scenario, our test results provide comprehensive insights into the strengths and limitations of ChatGPT 3.5, particularly in the domain of English sentence parsing and language understanding tasks. This knowledge is instrumental in further enhancing the capabilities of ChatGPT for these specific tasks.

As we investigate into the future, ongoing efforts will focus on refining ChatGPT for improved performance in English sentence parsing, thus bridging the gap between language models and semantic graph construction. The integration of additional linguistic resources, enhanced fine-tuning techniques, and prompt engineering strategies will be explored to further empower ChatGPT in its role as a dynamic tool for language understanding and knowledge representation.

ACKNOWLEDGMENT

This research work is carried out as part of the EFOP-3.6.1-16-2016-00011 ''Younger and Renewing University–Innovative Knowledge City–Institutional development of the University of Miskolc aiming at intelligent specialization'' project implemented in the framework of the Széchenyi 2020 program.

REFERENCES

[1] D. Rusu, B. Fortuna, M. Grobelnik, and D. Mladenic, ''Semantic graphs derived from triplets application in document summarization,'' *Informatica*, vol. 33, no. 3, pp. 357–362, 2009.
[2] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, ''OGB-LSC: A large-scale challenge for machine learning on graphs,'' 2021, *arXiv:2103.09430*.
[3] P. Goyal and E. Ferrara, ''Graph embedding techniques, applications, and performance: A survey,'' *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018, doi: 10.1016/j.knosys.2018.03.022.
[4] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, ''A survey on knowledge graphs: Representation, acquisition, and applications,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2021, doi: 10.1109/TNNLS.2021.3070843.
[5] B. Abu-Salih, ''Domain-specific knowledge graphs: A survey,'' *J. Netw. Comput. Appl.*, vol. 185, Jul. 2021, Art. no. 103076, doi: 10.1016/j.jnca.2021.103076.
[6] H. Singh, P. Jain, Mausam, and S. Chakrabarti, ''Multilingual knowledge graph completion with joint relation and entity alignment,'' 2021, *arXiv:2104.08804*.
[7] X. Zou, ''A survey on application of knowledge graph,'' *J. Phys., Conf. Ser.*, vol. 1487, no. 1, Mar. 2020, Art. no. 012016, doi: 10.1088/1742-6596/1487/1/012016.
[8] L. Shi, S. Li, X. Yang, J. Qi, G. Pan, and B. Zhou, ''Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services,'' *BioMed Res. Int.*, vol. 2017, Feb. 2017, Art. no. 2858423, doi: 10.1155/2017/2858423.
[9] H. Oh and R. Jain, ''Detecting events of daily living using multimodal data,'' 2019, *arXiv:1905.09402*.
[10] M. Rotmensch, Y. Halpern, A. Tlimat, S. Horng, and D. Sontag, ''Learning a health knowledge graph from electronic medical records,'' *Sci. Rep.*, vol. 7, no. 1, p. 5994, Jul. 2017, doi: 10.1038/s41598-017-05778-z.
[11] J. Liu, Z. Lu, and W. Du, ''Combining enterprise knowledge graph and news sentiment analysis for stock price prediction,'' in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2019, pp. 1247–1255.
[12] S. A. Elnagdy, M. Qiu, and K. Gai, ''Cyber incident classifications using ontology-based knowledge representation for cybersecurity insurance in financial industry,'' in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2016, pp. 301–306, doi: 10.1109/CSCLOUD.2016.45.
[13] F. Zablith, ''Constructing social media links to formal learning: A knowledge graph approach,'' *Educ. Technol. Res. Develop.*, vol. 70, no. 2, pp. 559–584, Apr. 2022, doi: 10.1007/s11423-022-10091-2.
[14] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, ''A practical approach to constructing a knowledge graph for cybersecurity,'' *Engineering*, vol. 4, no. 1, pp. 53–60, Feb. 2018, doi: 10.1016/j.eng.2018.01.004.
[15] M. J. Cresswell, ''Adverbs of space and time,'' in *Adverbial Modification: Interval Semantics Its Rivals*. Dordrecht, The Netherlands: Springer, 1985, doi: 10.1007/978-94-009-5414-4.
[16] C. Piñón, ''Adverbs of completion in an event semantics,'' in *Perspectives on Aspect*. Dordrecht, The Netherlands: Springer, 2005, doi: 10.1007/1-4020-3232-3_8.
[17] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge, ''Summary of ChatGPT-related research and perspective towards the future of large language models,'' *Meta-Radiology*, vol. 1, no. 2, Sep. 2023, Art. no. 100017, doi: 10.1016/j.metrad.2023.100017.
[18] M. Abdullah, A. Madain, and Y. Jararweh, ''ChatGPT: Fundamentals, applications and social impacts,'' in *Proc. 9th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS)*, Nov. 2022, pp. 1–8, doi: 10.1109/SNAMS58071.2022.10062688.
[19] G. Vassiliou, N. Papadakis, and H. Kondylakis, ''SummaryGPT: Leveraging ChatGPT for summarizing knowledge graphs,'' in *Proc. Eur. Semantic Web Conf.*, 2023, pp. 164–168 doi: 10.1007/978-3-031-43458-7_31.
[20] P. Pichappan, M. Krishnamurthy, and P. Vijayakumar, ''Analysis of ChatGPT as a question-answering tool,'' *J. Digit. Inf. Manage.*, vol. 21, no. 2, pp. 50–61, Jun. 2023.
[21] H. B. D. B. Pereira, M. Grilo, I. D. S. Fadigas, C. T. D. S. Junior, M. D. V. Cunha, R. S. F. D. Barreto, J. C. Andrade, and T. Henrique, ''Systematic review of the 'semantic network' definitions,'' *Expert Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118455, doi: 10.1016/j.eswa.2022.118455.
[22] W. L. Hamilton, *Graph Representation Learning*. San Rafael, CA, USA: Morgan & Claypool, 2020.
[23] Y. Chen, X. Ge, S. Yang, L. Hu, J. Li, and J. Zhang, ''A survey on multimodal knowledge graphs: Construction, completion and applications,'' *Mathematics*, vol. 11, no. 8, p. 1815, Apr. 2023, doi: 10.3390/math11081815.
[24] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, ''Knowledge graphs: Opportunities and challenges,'' *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13071–13102, Nov. 2023, doi: 10.1007/s10462-023-10465-9.
[25] A. Kamath and R. Das, ''A survey on semantic parsing,'' 2018, *arXiv:1812.00978*.
[26] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, ''DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia,'' *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015, doi: 10.3233/sw-140134.
[27] Š. Čebirić, F. Goasdoué, H. Kondylakis, D. Kotzinos, I. Manolescu, G. Troullinou, and M. Zneika, ''Summarizing semantic graphs: A survey,'' *VLDB J.*, vol. 28, no. 3, pp. 295–327, Jun. 2019, doi: 10.1007/s00778-018-0528-3.
[28] A. Harth, ''Ontologies for knowledge representation: Tools and techniques for building ontologies,'' in *Semantic Web Technologies*. Boca Raton, FL, USA: CRC Press, 2022, doi: 10.1201/9781003309420.
[29] P. Pham, L. T. T. Nguyen, W. Pedrycz, and B. Vo, ''Deep learning, graph-based text representation and classification: A survey, perspectives and challenges,'' *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 4893–4927, Jun. 2023, doi: 10.1007/s10462-022-10265-7.
[30] L. Kovacs, ''Mapping between semantic graphs and sentences in grammar induction system,'' *BRAIN, Broad Res. Artif. Intell. Neurosci.*, vol. 1, pp. 100–112, Jul. 2010.
[31] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, ''A comprehensive overview of large language models,'' 2023, *arXiv:2307.06435*.
[32] E. Kotei and R. Thirunavukarasu, ''A systematic review of transformer-based pre-trained language models through self-supervised learning,'' *Information*, vol. 14, no. 3, p. 187, Mar. 2023, doi: 10.3390/info14030187.
[33] S. R. Bowman, ''Eight things to know about large language models,'' 2023, *arXiv:2304.00612*.
[34] T. Brown et al., ''Language models are few-shot learners,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.
[35] C. D. Manning, ''Human language understanding & reasoning,'' *Daedalus*, vol. 151, no. 2, pp. 127–138, 2022, doi: 10.1162/daed_a_01905.
[36] B. Meskó, ''Prompt engineering as an important emerging skill for medical professionals: Tutorial,'' *J. Med. Internet Res.*, vol. 25, Oct. 2023, Art. no. e50638, doi: 10.2196/50638.
[37] W. W. Zachary, ''An information flow model for conflict and fission in small groups,'' *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[38] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "ChatGPT for robotics: Design principles and model abilities," Microsoft, Redmond, WA, USA, Tech. Rep. MSR-TR-2023-8, 2023. [Online]. Available: https://www.microsoft.com/en-us/research/publication/chatgpt-for-robotics-design-principles-and-model-abilities/

[39] P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 121–154, 2023, doi: 10.1016/j.iotcps.2023.04.003.

[40] G. Wohlgenannt, *Learning Ontology Relations by Combining Corpus-Based Techniques and Reasoning on Data From Semantic Web Sources*. Frankfurt am Main, Germany: Peter Lang International Academic, 2018.

[41] A. Harth, K. Hose, and R. Schenkel, *Linked Data Management*. Boca Raton, FL, USA: CRC Press, 2014.

[42] F.-X. Desmarais, M. Gagnon, and A. Zouaq, "Comparing a rule-based and a machine learning approach for semantic analysis," in *Proc. 6th Int. Conf. Adv. Semantic Process.*, Barcelona, Spain, 2012, pp. 103–108.

WALELIGN TEWABE SEWUNETIE received the B.Sc. degree in computer science from Mekelle University, Ethiopia, and the M.Sc. degree in computer science from Arba Minch University, Ethiopia. He is currently pursuing the Ph.D. degree with the University of Miskolc, Hungary. His research interests include soft computing, machine learning, NLP, and intelligent tutoring systems.

LÁSZLÓ KOVÁCS is currently a Full Professor with the Institute of Information Sciences, University of Miskolc, Hungary. He has authored or coauthored more than 194 publications: 109 conference proceedings and 85 journals. He has supervised more than seven Ph.D. students. His research interests include database and knowledge base modeling, concept lattice structures, discrete optimizations, and machine learning algorithms in NLP.